
Boto3 Documentation

Release 1.1.4

Amazon.com, Inc.

September 28, 2015

1	Quickstart	3
1.1	Quickstart	3
1.2	A Sample Tutorial	4
2	User Guide	9
2.1	User Guides	9
3	API Reference	39
3.1	Services	39
3.2	Core	4634
3.3	Customizations	4647
4	Indices and tables	4655
	Python Module Index	4657

Boto is the Amazon Web Services (AWS) SDK for Python, which allows Python developers to write software that makes use of Amazon services like S3 and EC2. Boto provides an easy to use, object-oriented API as well as low-level direct service access.

Quickstart

1.1 Quickstart

Getting started with Boto 3 is easy, but requires a few steps.

1.1.1 Installation

Install the latest Boto 3 release via **pip**:

```
pip install boto3
```

You may also install a specific version:

```
pip install boto3==1.0.0
```

Note: The latest development version can always be found on [GitHub](#).

1.1.2 Configuration

Before you can begin using Boto 3, you should set up authentication credentials. Credentials for your AWS account can be found in the [IAM Console](#). You can create or use an existing user. Go to manage access keys and generate a new set of keys.

If you have the [AWS CLI](#) installed, then you can use it to configure your credentials file:

```
aws configure
```

Alternatively, you can create the credential file yourself. By default, its location is at `~/.aws/credentials`:

```
[default]
aws_access_key_id = YOUR_ACCESS_KEY
aws_secret_access_key = YOUR_SECRET_KEY
```

You may also want to set a default region. This can be done in the configuration file. By default, its location is at `~/.aws/config`:

```
[default]
region=us-east-1
```

Alternatively, you can pass a `region_name` when creating clients and resources.

This sets up credentials for the default profile as well as a default region to use when creating connections. See [Configuration](#) for in-depth configuration sources and options.

1.1.3 Using Boto 3

To use Boto 3, you must first import it and tell it what service you are going to use:

```
import boto3

# Let's use Amazon S3
s3 = boto3.resource('s3')
```

Now that you have an `s3` resource, you can make requests and process responses from the service. The following uses the `buckets` collection to print out all bucket names:

```
# Print out bucket names
for bucket in s3.buckets.all():
    print(bucket.name)
```

It's also easy to upload and download binary data. For example, the following uploads a new file to S3. It assumes that the bucket `my-bucket` already exists:

```
# Upload a new file
data = open('test.jpg', 'rb')
s3.Bucket('my-bucket').put_object(Key='test.jpg', Body=data)
```

[Resources](#) and [Collections](#) will be covered in more detail in the following sections, so don't worry if you do not completely understand the examples.

1.2 A Sample Tutorial

This tutorial will show you how to use Boto3 with an AWS service. In this sample tutorial, you will learn how to use Boto3 with [Amazon Simple Queue Service \(SQS\)](#)

1.2.1 SQS

SQS allows you to queue and then process messages. This tutorial covers how to create a new queue, get and use an existing queue, push new messages onto the queue, and process messages from the queue by using [Resources](#) and [Collections](#).

1.2.2 Creating a Queue

Queues are created with a name. You may also optionally set queue attributes, such as the number of seconds to wait before an item may be processed. The examples below will use the queue name `test`. Before creating a queue, you must first get the SQS service resource:

```
# Get the service resource
sqs = boto3.resource('sqs')

# Create the queue. This returns an SQS.Queue instance
queue = sqs.create_queue(QueueName='test', Attributes={'DelaySeconds': '5'})
```

```
# You can now access identifiers and attributes
print(queue.url)
print(queue.attributes.get('DelaySeconds'))
```

Reference: `SQS.ServiceResource.create_queue()`

Warning: The code above may throw an exception if you already have a queue named `test`.

1.2.3 Using an Existing Queue

It is possible to look up a queue by its name. If the queue does not exist, then an exception will be thrown:

```
# Get the service resource
sqs = boto3.resource('sqs')

# Get the queue. This returns an SQS.Queue instance
queue = sqs.get_queue_by_name(QueueName='test')

# You can now access identifiers and attributes
print(queue.url)
print(queue.attributes.get('DelaySeconds'))
```

It is also possible to list all of your existing queues:

```
# Print out each queue name, which is part of its ARN
for queue in sqs.queues.all():
    print(queue.url)
```

Note: To get the name from a queue, you must use its ARN, which is available in the queue's `attributes` attribute. Using `queue.attributes['QueueArn'].split(':')[4]` will return its name.

Reference: `SQS.ServiceResource.get_queue_by_name()`, `SQS.ServiceResource.queues`

1.2.4 Sending Messages

Sending a message adds it to the end of the queue:

```
# Get the service resource
sqs = boto3.resource('sqs')

# Get the queue
queue = sqs.get_queue_by_name(QueueName='test')

# Create a new message
response = queue.send_message(MessageBody='world')

# The response is NOT a resource, but gives you a message ID and MD5
print(response.get('MessageId'))
print(response.get('MD5OfMessageBody'))
```

You can also create messages with custom attributes:

```
queue.send_message(MessageBody='boto3', MessageAttributes={
    'Author': {
```

```
        'StringValue': 'Daniel',
        'DataType': 'string'
    }
})
```

Messages can also be sent in batches. For example, sending the two messages described above in a single request would look like the following:

```
response = queue.send_messages(Entries=[
    {
        'Id': '1',
        'MessageBody': 'world'
    },
    {
        'Id': '2',
        'MessageBody': 'boto3',
        'MessageAttributes': {
            'Author': {
                'StringValue': 'Daniel',
                'DataType': 'string'
            }
        }
    }
])

# Print out any failures
print(response.get('Failed'))
```

In this case, the response contains lists of Successful and Failed messages, so you can retry failures if needed.

Reference: `SQS.Queue.send_message()`, `SQS.Queue.send_messages()`

1.2.5 Processing Messages

Messages are processed in batches:

```
# Get the service resource
sqs = boto3.resource('sqs')

# Get the queue
queue = sqs.get_queue_by_name(QueueName='test')

# Process messages by printing out body and optional author name
for message in queue.receive_messages(MessageAttributeNames=['Author']):
    # Get the custom author message attribute if it was set
    author_text = ''
    if message.message_attributes is not None:
        author_name = message.message_attributes.get('Author').get('StringValue')
        if author_name:
            author_text = ' ({})'.format(author_name)

    # Print out the body and author (if set)
    print('Hello, {}!{}'.format(message.body, author_text))

    # Let the queue know that the message is processed
    message.delete()
```

Given *only* the messages that were sent in a batch with `SQS.Queue.send_messages()` in the previous section, the above code will print out:

```
Hello, world!  
Hello, boto3! (Daniel)
```

Reference: `SQS.Queue.receive_messages()`, `SQS.Message.delete()`

2.1 User Guides

2.1.1 Migration Guides

What's New

Boto 3 is a ground-up rewrite of Boto. It uses a data-driven approach to generate classes at runtime from JSON description files that are shared between SDKs in various languages. This includes descriptions for a high level, object oriented interface similar to those available in previous versions of Boto.

Because Boto 3 is generated from these shared JSON files, we get fast updates to the latest services and features and a consistent API across services. Community contributions to JSON description files in other SDKs also benefit Boto 3, just as contributions to Boto 3 benefit the other SDKs.

Major Features

Boto 3 consists of the following major features:

- **Resources:** a high level, object oriented interface
- **Collections:** a tool to iterate and manipulate groups of resources
- **Clients:** low level service connections
- **Paginators:** automatic paging of responses
- **Waiters:** a way to block until a certain state has been reached

Along with these major features, Boto 3 also provides *sessions* and per-session *credentials & configuration*, as well as basic components like *authentication*, *parameter & response* handling, an *event system* for customizations and logic to *retry* failed requests.

Botocore Boto 3 is built atop of a library called [Botocore](#), which is shared by the [AWS CLI](#). Botocore provides the low level clients, session, and credential & configuration data. Boto 3 builds on top of Botocore by providing its own session, resources and collections.

Migrating from Boto 2.x

Current Boto users can begin using Boto 3 right away. The two modules can live side-by-side in the same project, which means that a piecemeal approach can be used. New features can be written in Boto 3, or existing code can be migrated over as needed, piece by piece.

High Level Concepts

Boto 2.x modules are typically split into two categories, those which include a high-level object-oriented interface and those which include only a low-level interface which matches the underlying Amazon Web Services API. Some modules are completely high-level (like Amazon S3 or EC2), some include high-level code on top of a low-level connection (like Amazon DynamoDB), and others are 100% low-level (like Amazon Elastic Transcoder).

In Boto 3 this general low-level and high-level concept hasn't changed much, but there are two important points to understand.

Data Driven First, in Boto 3 classes are created at runtime from JSON data files that describe AWS APIs and organizational structures built atop of them. These data files are loaded at runtime and can be modified and updated without the need of installing an entirely new SDK release.

A side effect of having all the services generated from JSON files is that there is now consistency between all AWS service modules. One important change is that *all* API call parameters must now be passed as **keyword arguments**, and these keyword arguments take the form defined by the upstream service. Though there are exceptions, this typically means UpperCamelCasing parameter names. You will see this in the service-specific migration guides linked to below.

Resource Objects Second, while every service now uses the runtime-generated low-level client, some services additionally have high-level generated objects that we refer to as `Resources`. The lower-level is comparable to Boto 2.x layer 1 connection objects in that they provide a one to one mapping of API operations and return low-level responses. The higher level is comparable to the high-level customizations from Boto 2.x: an S3 `Key`, an EC2 `Instance`, and a DynamoDB `Table` are all considered resources in Boto 3. Just like a Boto 2.x `S3Connection`'s `list_buckets` will return `Bucket` objects, the Boto 3 resource interface provides actions and collections that return resources. Some services may also have hand-written customizations built on top of the runtime-generated high-level resources (such as utilities for working with S3 multipart uploads).

```
import boto, boto3

# Low-level connections
conn = boto.connect_elastictranscoder()
client = boto3.client('elastictranscoder')

# High-level connections & resource objects
from boto.s3.bucket import Bucket
s3_conn = boto.connect_s3()
boto2_bucket = Bucket('mybucket')

s3 = boto3.resource('s3')
boto3_bucket = s3.Bucket('mybucket')
```

Installation & Configuration

The [Quickstart](#) guide provides instructions for installing Boto 3. You can also follow the instructions there to set up new credential files, or you can continue to use your existing Boto 2.x credentials. Please note that Boto 3, the AWS

CLI, and several other SDKs all use the shared credentials file (usually at `~/.aws/credentials`).

Once configured, you may begin using Boto 3:

```
import boto3

for bucket in boto3.resource('s3').buckets.all():
    print(bucket.name)
```

See the *tutorial_list* and [Boto 3 Documentation](#) for more information.

The rest of this document will describe specific common usage scenarios of Boto 2 code and how to accomplish the same tasks with Boto 3.

Services

Amazon S3 Boto 2.x contains a number of customizations to make working with Amazon S3 buckets and keys easy. Boto 3 exposes these same objects through its resources interface in a unified and consistent way.

Creating the Connection Boto 3 has both low-level clients and higher-level resources. For Amazon S3, the higher-level resources are the most similar to Boto 2.x's `s3` module:

```
# Boto 2.x
import boto
s3_connection = boto.connect_s3()

# Boto 3
import boto3
s3 = boto3.resource('s3')
```

Creating a Bucket Creating a bucket in Boto 2 and Boto 3 is very similar, except that in Boto 3 all action parameters must be passed via keyword arguments and a bucket configuration must be specified manually:

```
# Boto 2.x
s3_connection.create_bucket('mybucket')
s3_connection.create_bucket('mybucket', location=Location.USWest)

# Boto 3
s3.create_bucket(Bucket='mybucket')
s3.create_bucket(Bucket='mybucket', CreateBucketConfiguration={
    'LocationConstraint': 'us-west-1'})
```

Storing Data Storing data from a file, stream, or string is easy:

```
# Boto 2.x
from boto.s3.key import Key
key = Key('hello.txt')
key.set_contents_from_file('/tmp/hello.txt')

# Boto 3
s3.Object('mybucket', 'hello.txt').put(Body=open('/tmp/hello.txt', 'rb'))
```

Accessing a Bucket Getting a bucket is easy with Boto 3's resources, however these do not automatically validate whether a bucket exists:

```
# Boto 2.x
bucket = s3_connection.get_bucket('mybucket', validate=False)
exists = s3_connection.lookup('mybucket')

# Boto 3
import boto3
bucket = s3.Bucket('mybucket')
exists = True
try:
    s3.meta.client.head_bucket(Bucket='mybucket')
except boto3.exceptions.ClientError as e:
    # If a client error is thrown, then check that it was a 404 error.
    # If it was a 404 error, then the bucket does not exist.
    error_code = int(e.response['Error']['Code'])
    if error_code == 404:
        exists = False
```

Deleting a Bucket All of the keys in a bucket must be deleted before the bucket itself can be deleted:

```
# Boto 2.x
for key in bucket:
    key.delete()
bucket.delete()

# Boto 3
for key in bucket.objects.all():
    key.delete()
bucket.delete()
```

Iteration of Buckets and Keys Bucket and key objects are no longer iterable, but now provide collection attributes which can be iterated:

```
# Boto 2.x
for bucket in s3_connection:
    for key in bucket:
        print(key.name)

# Boto 3
for bucket in s3.buckets.all():
    for key in bucket.objects.all():
        print(key.key)
```

Access Controls Getting and setting canned access control values in Boto 3 operates on an ACL resource object:

```
# Boto 2.x
bucket.set_acl('public-read')
key.set_acl('public-read')

# Boto 3
bucket.Acl().put(ACL='public-read')
obj.put(ACL='public-read')
```

It's also possible to retrieve the policy grant information:

```
# Boto 2.x
acp = bucket.get_acl()
for grant in acp.acl.grants:
    print(grant.display_name, grant.permission)

# Boto 3
acl = bucket.Acl()
for grant in acl.grants:
    print(grant['DisplayName'], grant['Permission'])
```

Boto 3 lacks the grant shortcut methods present in Boto 2.x, but it is still fairly simple to add grantees:

```
# Boto 2.x
bucket.add_email_grant('READ', 'user@domain.tld')

# Boto 3
bucket.Acl.put(GrantRead='emailAddress=user@domain.tld')
```

Key Metadata It's possible to set arbitrary metadata on keys:

```
# Boto 2.x
key.set_metadata('meta1', 'This is my metadata value')
print(key.get_metadata('meta1'))

# Boto 3
key.put(Metadata={'meta1': 'This is my metadata value'})
print(key.metadata['meta1'])
```

Managing CORS Configuration Allows you to manage the cross-origin resource sharing configuration for S3 buckets:

```
# Boto 2.x
cors = bucket.get_cors()

config = CORSConfiguration()
config.add_rule('GET', '*')
bucket.set_cors(config)

bucket.delete_cors()

# Boto 3
cors = bucket.Cors()

config = {
    'CORSRules': [
        {
            'AllowedMethods': ['GET'],
            'AllowedOrigins': ['*']
        }
    ]
}
cors.put(CORSConfiguration=config)

cors.delete()
```

Amazon EC2 Boto 2.x contains a number of customizations to make working with Amazon EC2 instances, storage and networks easy. Boto 3 exposes these same objects through its resources interface in a unified and consistent way.

Creating the Connection Boto 3 has both low-level clients and higher-level resources. For Amazon EC2, the higher-level resources are the most similar to Boto 2.x's `ec2` and `vpc` modules:

```
# Boto 2.x
import boto
ec2_connection = boto.connect_ec2()
vpc_connection = boto.connect_vpc()

# Boto 3
import boto3
ec2 = boto3.resource('ec2')
```

Launching New Instances Launching new instances requires an image ID and the number of instances to launch. It can also take several optional parameters, such as the instance type and security group:

```
# Boto 2.x
ec2_connection.run_instances('<ami-image-id>')

# Boto 3
ec2.create_instances(ImageId='<ami-image-id>', MinCount=1, MaxCount=5)
```

Stopping & Terminating Instances Stopping and terminating multiple instances given a list of instance IDs uses Boto 3 collection filtering:

```
ids = ['instance-id-1', 'instance-id-2', ...]

# Boto 2.x
ec2_connection.stop_instances(instance_ids=ids)
ec2_connection.terminate_instances(instance_ids=ids)

# Boto 3
ec2.instances.filter(InstanceIds=ids).stop()
ec2.instances.filter(InstanceIds=ids).terminate()
```

Checking What Instances Are Running Boto 3 collections come in handy when listing all your running instances as well. Every collection exposes a `filter` method that allows you to pass additional parameters to the underlying service API operation. The EC2 instances collection takes a parameter called `Filters` which is a list of names and values, for example:

```
# Boto 2.x
reservations = ec2_connection.get_all_reservations(
    filters={'instance-state-name': 'running'})
for reservation in reservations:
    for instance in reservation.instances:
        print(instance.instance_id, instance.instance_type)

# Boto 3
# Use the filter() method of the instances collection to retrieve
# all running EC2 instances.
instances = ec2.instances.filter(
    Filters=[{'Name': 'instance-state-name', 'Values': ['running']}])
```

```
for instance in instances:
    print(instance.id, instance.instance_type)
```

Checking Health Status Of Instances It is possible to get scheduled maintenance information for your running instances. At the time of this writing Boto 3 does not have a status resource, so you must drop down to the low-level client via `ec2.meta.client`:

```
# Boto 2.x
for status in ec2_connection.get_all_instance_statuses():
    print(status)

# Boto 3
for status in ec2.meta.client.describe_instance_status()['InstanceStatuses']:
    print(status)
```

Working with EBS Snapshots Snapshots provide a way to create a copy of an EBS volume, as well as make new volumes from the snapshot which can be attached to an instance:

```
# Boto 2.x
snapshot = ec2_connection.create_snapshot('volume-id', 'Description')
volume = snapshot.create_volume('us-west-2')
ec2_connection.attach_volume(volume.id, 'instance-id', '/dev/sdy')
ec2_connection.delete_snapshot(snapshot.id)

# Boto 3
snapshot = ec2.create_snapshot(VolumeId='volume-id', Description='description')
volume = ec2.create_volume(SnapshotId=snapshot.id, AvailabilityZone='us-west-2a')
ec2.Instance('instance-id').attach_volume(VolumeId=volume.id, Device='/dev/sdy')
snapshot.delete()
```

Creating a VPC, Subnet, and Gateway Creating VPC resources in Boto 3 is very similar to Boto 2.x:

```
# Boto 2.x
vpc = vpc_connection.create_vpc('10.0.0.0/24')
subnet = vpc_connection.create_subnet(vpc.id, '10.0.0.0/25')
gateway = vpc_connection.create_internet_gateway()

# Boto 3
vpc = ec2.create_vpc(CidrBlock='10.0.0.0/24')
subnet = vpc.create_subnet(CidrBlock='10.0.0.0/25')
gateway = ec2.create_internet_gateway()
```

Attaching and Detaching an Elastic IP and Gateway Elastic IPs and gateways provide a way for instances inside of a VPC to communicate with the outside world:

```
# Boto 2.x
ec2_connection.attach_internet_gateway(gateway.id, vpc.id)
ec2_connection.detach_internet_gateway(gateway.id, vpc.id)

from boto.ec2.address import Address
address = Address()
address.allocation_id = 'eipalloc-35cf685d'
address.associate('i-71b2f60b')
address.disassociate()
```

```
# Boto 3
gateway.attach_to_vpc(VpcId=vpc.id)
gateway.detach_from_vpc(VpcId=vpc.id)

address = ec2.VpcAddress('eipalloc-35cf685d')
address.associate('i-71b2f60b')
address.association.delete()
```

2.1.2 General Feature Guides

Resources

Overview

Resources represent an object-oriented interface to Amazon Web Services (AWS). They provide a higher-level abstraction than the raw, low-level calls made by service clients. To use resources, you invoke the `resource()` method of a `Session` and pass in a service name:

```
# Get resources from the default session
sqs = boto3.resource('sqs')
s3 = boto3.resource('s3')
```

Every resource instance has a number of attributes and methods. These can conceptually be split up into identifiers, attributes, actions, references, sub-resources, and collections. Each of these is described in further detail below and in the following section.

Resources themselves can also be conceptually split into service resources (like `sqs`, `s3`, `ec2`, etc) and individual resources (like `sqs.Queue` or `s3.Bucket`). Service resources *do not* have identifiers or attributes. The two share the same components otherwise.

Identifiers & Attributes

An identifier is a unique value that is used to call actions on the resource. Resources **must** have at least one identifier, except for the top-level service resources (e.g. `sqs` or `s3`). An identifier is set at instance creation-time, and failing to provide all necessary identifiers during instantiation will result in an exception. Examples of identifiers:

```
# SQS Queue (url is an identifier)
queue = sqs.Queue(url='http://...')
print(queue.url)

# S3 Object (bucket_name and key are identifiers)
obj = s3.Object(bucket_name='boto3', key='test.py')
print(obj.bucket_name)
print(obj.key)

# Raises exception, missing identifier: key!
obj = s3.Object(bucket_name='boto3')
```

Identifiers may also be passed as positional arguments:

```
# SQS Queue
queue = sqs.Queue('http://...')

# S3 Object
```



```
obj = s3.Object('boto3', 'test.py')

# Raises exception, missing key!
obj = s3.Object('boto3')
```

Identifiers also play a role in resource instance equality. For two instances of a resource to be considered equal, their identifiers must be equal:

```
>>> bucket1 = s3.Bucket('boto3')
>>> bucket2 = s3.Bucket('boto3')
>>> bucket3 = s3.Bucket('some-other-bucket')

>>> bucket1 == bucket2
True
>>> bucket1 == bucket3
False
```

Note: Only identifiers are taken into account for instance equality. Region, account ID and other data members are not considered. When using temporary credentials or multiple regions in your code please keep this in mind.

Resources may also have attributes, which are *lazy-loaded* properties on the instance. They may be set at creation time from the response of an action on another resource, or they may be set when accessed or via an explicit call to the `load` or `reload` action. Examples of attributes:

```
# SQS Message
message.body

# S3 Object
obj.last_modified
obj.md5
```

Warning: Attributes may incur a load action when first accessed. If latency is a concern, then manually calling `load` will allow you to control exactly when the load action (and thus latency) is invoked. The documentation for each resource explicitly lists its attributes.

Additionally, attributes may be reloaded after an action has been performed on the resource. For example, if the `last_modified` attribute of an S3 object is loaded and then a `put` action is called, then the next time you access `last_modified` it will reload the object's metadata.

Actions

An action is a method which makes a call to the service. Actions may return a low-level response, a new resource instance or a list of new resource instances. Actions automatically set the resource identifiers as parameters, but allow you to pass additional parameters via keyword arguments. Examples of actions:

```
# SQS Queue
messages = queue.receive_messages()

# SQS Message
for message in messages:
    message.delete()

# S3 Object
obj = s3.Object(bucket_name='boto3', key='test.py')
response = obj.get()
data = response['Body'].read()
```

Examples of sending additional parameters:

```
# SQS Service
queue = sqs.get_queue_by_name(QueueName='test')

# SQS Queue
queue.send_message(MessageBody='hello')
```

Note: Parameters **must** be passed as keyword arguments. They will not work as positional arguments.

References

A reference is an attribute which may be `None` or a related resource instance. The resource instance does not share identifiers with its reference resource, that is, it is not a strict parent to child relationship. In relational terms, these can be considered many-to-one or one-to-one. Examples of references:

```
# EC2 Instance
instance.subnet
instance.vpc
```

In the above example, an EC2 instance may have exactly one associated subnet, and may have exactly one associated VPC. The subnet does not require the instance ID to exist, hence it is not a parent to child relationship.

Sub-resources

A sub-resource is similar to a reference, but is a related class rather than an instance. Sub-resources, when instantiated, share identifiers with their parent. It is a strict parent-child relationship. In relational terms, these can be considered one-to-many. Examples of sub-resources:

```
# SQS
queue = sqs.Queue(url='...')
message = queue.Message(receipt_handle='...')
print(queue.url == message.queue_url)
print(message.receipt_handle)

# S3
obj = bucket.Object(key='new_file.txt')
print(obj.bucket_name)
print(obj.key)
```

Because an SQS message cannot exist without a queue, and an S3 object cannot exist without a bucket, these are parent to child relationships.

Waiters

A waiter is similar to an action. A waiter will poll the status of a resource and suspend execution until the resource reaches the state that is being polling for or a failure occurs while polling. Waiters automatically set the resource identifiers as parameters, but allow you to pass additional parameters via keyword arguments. Examples of waiters include:

```
# S3: Wait for a bucket to exist.
bucket.wait_until_exists()
```

```
# EC2: Wait for an instance to reach the running state.
instance.wait_until_running()
```

Multithreading

It is recommended to create a resource instance for each thread in a multithreaded application rather than sharing a single instance among the threads. For example:

```
import boto3
import boto3.session
import threading

class MyTask(threading.Thread):
    def run(self):
        session = boto3.session.Session()
        s3 = session.resource('s3')
        # ... do some work with S3 ...
```

In the example above, each thread would have its own Boto 3 session and its own instance of the S3 resource. This is a good idea because resources contain shared data when loaded and calling actions, accessing properties, or manually loading or reloading the resource can modify this data.

Collections

Overview

A collection provides an iterable interface to a group of resources. Collections behave similarly to [Django QuerySets](#) and expose a similar API. A collection seamlessly handles pagination for you, making it possible to easily iterate over all items from all pages of data. Example of a collection:

```
# SQS list all queues
sqs = boto3.resource('sqs')
for queue in sqs.queues.all():
    print(queue.url)
```

When Collections Make Requests Collections can be created and manipulated without any request being made to the underlying service. A collection makes a remote service request under the following conditions:

- **Iteration:**

```
for bucket in s3.buckets.all():
    print(bucket.name)
```

- **Conversion to list():**

```
buckets = list(s3.buckets.all())
```

- **Batch actions (see below):**

```
s3.Bucket('my-bucket').objects.delete()
```

Filtering

Some collections support extra arguments to filter the returned data set, which are passed into the underlying service operation. Use the `filter()` method to filter the results:

```
# S3 list all keys with the prefix '/photos'
s3 = boto3.resource('s3')
for bucket in s3.buckets.all():
    for obj in bucket.objects.filter(Prefix='/photos'):
        print('{0}:{1}'.format(bucket.name, obj.key))
```

Warning: Behind the scenes, the above example will call `ListBuckets`, `ListObjects`, and `HeadObject` many times. If you have a large number of S3 objects then this could incur a significant cost.

Chainability

Collection methods are chainable. They return copies of the collection rather than modifying the collection, including a deep copy of any associated operation parameters. For example, this allows you to build up multiple collections from a base which they all have in common:

```
# EC2 find instances
ec2 = boto3.resource('ec2')
base = ec2.instances.filter(InstanceIds=['id1', 'id2', 'id3'])

filters = [{
    'name': 'tenancy',
    'value': 'dedicated'
}]
filtered1 = base.filter(Filters=filters)

# Note, this does NOT modify the filters in ``filtered1``!
filters.append({'name': 'instance-type', 'value': 't1.micro'})
filtered2 = base.filter(Filters=filters)

print('All instances:')
for instance in base:
    print(instance.id)

print('Dedicated instances:')
for instance in filtered1:
    print(instance.id)

print('Dedicated micro instances:')
for instance in filtered2:
    print(instance.id)
```

Limiting Results

It is possible to limit the number of items returned from a collection by using either the `limit()` method:

```
# S3 iterate over first ten buckets
for bucket in s3.buckets.limit(10):
    print(bucket.name)
```

In both cases, up to 10 items total will be returned. If you do not have 10 buckets, then all of your buckets will be returned.

Controlling Page Size

Collections automatically handle paging through results, but you may want to control the number of items returned from a single service operation call. You can do so using the `page_size()` method:

```
# S3 iterate over all objects 100 at a time
for obj in bucket.objects.page_size(100):
    print(obj.key)
```

By default, S3 will return 1000 objects at a time, so the above code would let you process the items in smaller batches, which could be beneficial for slow or unreliable internet connections.

Batch Actions

Some collections support batch actions, which are actions that operate on an entire page of results at a time. They will automatically handle pagination:

```
# S3 delete everything in 'my-bucket'
s3 = boto3.resource('s3')
s3.buckets('my-bucket').objects.delete()
```

Danger: The above example will **completely erase all data** in the `my-bucket` bucket! Please be careful with batch actions.

Low-level Clients

Clients provide a low-level interface to AWS whose methods map close to 1:1 with service APIs. All service operations are supported by clients. Clients are generated from a JSON service definition file.

Creating Clients

Clients are created in a similar fashion to resources:

```
import boto3

# Create a low-level client with the service name
sqs = boto3.client('sqs')
```

It is also possible to access the low-level client from an existing resource:

```
# Create the resource
sqs_resource = boto3.resource('sqs')

# Get the client from the resource
sqs = sqs_resource.meta.client
```

Service Operations

Service operations map to client methods of the same name and provide access to the same operation parameters via keyword arguments:

```
# Make a call using the low-level client
response = sqs.send_message(QueueUrl='...', MessageBody='...')
```

As can be seen above, the method arguments map directly to the associated [SQS API](#).

Note: The method names have been snake-cased for better looking Python code.

Parameters **must** be sent as keyword arguments. They will not work as positional arguments.

Handling Responses

Responses are returned as python dictionaries. It is up to you to traverse or otherwise process the response for the data you need, keeping in mind that responses may not always include all of the expected data. In the example below, `response.get('QueueUrls', [])` is used to ensure that a list is always returned, even when the response has no key 'QueueUrls':

```
# List all your queues
response = sqs.list_queues()
for url in response.get('QueueUrls', []):
    print(url)
```

The response in the example above looks something like this:

```
{
  "QueueUrls": [
    "http://url1",
    "http://url2",
    "http://url3"
  ]
}
```

Waiters

Waiters use a client's service operations to poll the status of an AWS resource and suspend execution until the AWS resource reaches the state that the waiter is polling for or a failure occurs while polling. Using clients, you can learn the name of each waiter that a client has access to:

```
import boto3

s3 = boto3.client('s3')
sqs = boto3.client('sqs')

# List all of the possible waiters for both clients
print("s3 waiters:")
s3.waiter_names

print("sqs waiters:")
sqs.waiter_names
```

Note if a client does not have any waiters, it will return an empty list when accessing its `waiter_names` attribute:

```
s3 waiters:
[u'bucket_exists', u'bucket_not_exists', u'object_exists', u'object_not_exists']
sqs waiters:
[]
```

Using a client's `get_waiter()` method, you can obtain a specific waiter from its list of possible waiters:

```
# Retrieve waiter instance that will wait till a specified
# S3 bucket exists
s3_bucket_exists_waiter = s3.get_waiter('bucket_exists')
```

Then to actually start waiting, you must call the waiter's `wait()` method with the method's appropriate parameters passed in:

```
# Begin waiting for the S3 bucket, mybucket, to exist
s3_bucket_exists_waiter.wait(Bucket='mybucket')
```

Session

A session manages state about a particular configuration. By default a session is created for you when needed. However it is possible and recommended to maintain your own session(s) in some scenarios. Sessions typically store:

- Credentials
- Region
- Other configurations

Default Session

The `boto3` module acts as a proxy to the default session, which is created automatically when needed. Example default session use:

```
# Using the default session
sqs = boto3.client('sqs')
s3 = boto3.resource('s3')
```

Custom Session

It is also possible to manage your own session and create clients or resources from it:

```
# Creating your own session
session = boto3.session.Session()

sqs = session.client('sqs')
s3 = session.resource('s3')
```

Configuration

Boto can be configured in multiple ways. Regardless of the source or sources that you choose, you **must** have AWS credentials and a region set in order to make requests.

Interactive Configuration

If you have the [AWS CLI](#), then you can use its interactive `configure` command to set up your credentials and default region:

```
aws configure
```

Follow the prompts and it will generate configuration files in the correct locations for you.

Configuration Sources

There are multiple sources from which configuration data can be loaded. The general order in which they are checked is as follows:

1. Method parameters
2. Environment variables
3. Configuration files
4. EC2 Instance metadata

If a configuration value is set in multiple places, then the first will be used according to the order above. For example, if I have set a default region in both my environment variables and configuration file, then the environment variable is used.

Available Options

The available options for various configuration sources are listed below.

Method Parameters When creating a session, client, or resource you can pass in credential and configuration options:

```
from boto3.session import Session

session = Session(aws_access_key_id='<YOUR ACCESS KEY ID>',
                  aws_secret_access_key='<YOUR SECRET KEY>',
                  region_name='<REGION NAME>')

ec2 = session.resource('ec2')
ec2_us_west_2 = session.resource('ec2', region_name='us-west-2')

# List all of my EC2 instances in my default region.
print('Default region:')
for instance in ec2.instances.all():
    print(instance.id)

# List all of my EC2 instances in us-west-2.
print('US West 2 region:')
for instance in ec2_us_west_2.instances.all():
    print(instance.id)
```

For a list of all options, look at the [Session](#) documentation.

Environment Variables

AWS_ACCESS_KEY_ID The access key for your AWS account.

AWS_SECRET_ACCESS_KEY The secret key for your AWS account.

AWS_DEFAULT_REGION The default region to use, e.g. *us-east-1*.

AWS_PROFILE The default credential and configuration profile to use, if any.

Configuration Files There are two configuration files that Boto checks. The first is the shared credential file, which holds only credentials and is shared between various SDKs and tools like Boto and the AWS CLI. By default, this file is located at `~/.aws/credentials`:

```
[default]
# The access key for your AWS account
aws_access_key_id=<YOUR ACCESS KEY ID>

# The secret key for your AWS account
aws_secret_access_key=<YOUR SECRET KEY>
```

Credentials can also be set for individual profiles:

```
[dev-profile]
# The access key for your dev-profile account
aws_access_key_id=<YOUR ACCESS KEY ID>

# The secret key for your dev-profile account
aws_secret_access_key=<YOUR SECRET KEY>
```

The second configuration file stores all settings which are not credentials. Its default location is `~/.aws/config`:

```
[default]
# The default region when making requests
region=<REGION NAME>
```

It also supports profiles, but these are prefixed with the word `profile` because this file supports sections other than profiles:

```
[profile dev-profile]
# The default region when using the dev-profile account
region=<REGION NAME>
```

Extensibility Guide

All of Boto3's resource and client classes are generated at runtime. This means that you cannot directly inherit and then extend the functionality of these classes because they do not exist until the program actually starts running.

However it is still possible to extend the functionality of classes through Boto3's event system.

An Introduction to the Event System

Boto3's event system allows users to register a function to a specific event. Then once the running program reaches a line that emits that specific event, Boto3 will call every function registered to the event in the order in which they were registered. When Boto3 calls each of these registered functions, it will call each of them with a specific set of keyword arguments that are associated with that event. Then once the registered function is called, the function may modify the keyword arguments passed to that function or return a value. Here is an example of how the event system works:

```
import boto3

s3 = boto3.client('s3')

# Access the event system on the S3 client
event_system = s3.meta.events

# Create a function
def add_my_bucket(params, **kwargs):
    # Add the name of the bucket you want to default to.
    if 'Bucket' not in params:
        params['Bucket'] = 'mybucket'

# Register the function to an event
event_system.register('provide-client-params.s3.ListObjects', add_my_bucket)

response = s3.list_objects()
```

In this example, the handler `add_my_bucket` is registered such that the handler will inject the value `'mybucket'` for the `Bucket` parameter whenever the `list_objects` client call is made without the `Bucket` parameter. Note that if the same `list_objects` call is made without the `Bucket` parameter and the registered handler, it will result in a validation error.

Here are the takeaways from this example:

- All clients have their own event system that you can use to fire events and register functions. You can access the event system through the `meta.events` attribute on the client.
- All functions registered to the event system must have `**kwargs` in the function signature. This is because emitting an event can have any number of keyword arguments emitted along side it, and so if your function is called without `**kwargs`, its signature will have to match every keyword argument emitted by the event. This also allows for more keyword arguments to be added to the emitted event in the future without breaking existing handlers.
- To register a function to an event, call the `register` method on the event system with the name of the event you want to register the function to and the function handle. Note that if you register the event after the event is emitted, the function will not be called unless the event is emitted again. In the example, the `add_my_bucket` handler was registered to the `'provide-client-params.s3.ListObjects'` event, which is an event that can be used to inject and modify parameters passed in by the client method. To read more about the event refer to [provide-client-params](#)

A Hierarchical Structure

The event system also provides a hierarchy for registering events such that you can register a function to a set of events depending on the event name heirarchy.

An event name can have its own heirachy by specifying `.` in its name. For example, take the event name `'general.specific.more_specific'`. When this event is emitted, the registered functions will be called in the order from most specific to least specific registration. So in this example, the functions will be called in the following order:

1. Functions registered to `'general.specific.more_specific'`
2. Functions registered to `'general.specific'`
3. Functions registered to `'general'`

Here is a deeper example of how the event system works with respect to its hierarchial structure:

```
import boto3

s3 = boto3.client('s3')

# Access the event system on the S3 client
event_system = s3.meta.events

def add_my_general_bucket(params, **kwargs):
    if 'Bucket' not in params:
        params['Bucket'] = 'mybucket'

def add_my_specific_bucket(params, **kwargs):
    if 'Bucket' not in params:
        params['Bucket'] = 'myspecificbucket'

event_system.register('provide-client-params.s3', add_my_general_bucket)
event_system.register('provide-client-params.s3.ListObjects', add_my_specific_bucket)

list_obj_response = s3.list_objects()
put_obj_response = s3.put_object(Key='mykey', Body=b'my body')
```

In this example, the `list_objects` method call will use the `'myspecificbucket'` for the bucket instead of `'mybucket'` because the `add_my_specific_bucket` method was registered to the `'provide-client-params.s3.ListObjects'` event which is more specific than the `'provide-client-params.s3'` event. Thus, the `add_my_specific_bucket` function is called before the `add_my_general_bucket` function is called when the event is emitted.

However for the `put_object` call, the bucket used is `'mybucket'`. This is because the event emitted for the `put_object` client call is `'provide-client-params.s3.PutObject'` and the `add_my_general_bucket` method is called via its registration to `'provide-client-params.s3'`. The `'provide-client-params.s3.ListObjects'` event is never emitted so the registered `add_my_specific_bucket` function is never called.

Wildcard Matching

Another aspect of Boto3's event system is that it has the capability to do wildcard matching using the `'*'` notation. Here is an example of using wildcards in the event system:

```
import boto3

s3 = boto3.client('s3')

# Access the event system on the S3 client
event_system = s3.meta.events

def add_my_wildcard_bucket(params, **kwargs):
    if 'Bucket' not in params:
        params['Bucket'] = 'mybucket'

event_system.register('provide-client-params.s3.*', add_my_wildcard_bucket)
response = s3.list_objects()
```

The `'*'` allows you to register to a group of events without having to know the actual name of the event. This is useful when you have to apply the same handler in multiple places. Also note that if the wildcard is used, it must be isolated. It does not handle globbing with additional characters. So in the previous example, if the `my_wildcard_function` was registered to `'provide-client-params.s3.*objects'`, the handler would not be called because it will consider `'provide-client-params.s3.*objects'` to be a specific event.

The wildcard also respects the hierarchical structure of the event system. If another handler was registered to the 'provide-client-params.s3' event, the `add_my_wildcard_bucket` would be called first because it is registered to 'provide-client-params.s3.*' which is more specific than the event 'provide-client.s3'.

Isolation of Event Systems

The event system in Boto3 has the notion of isolation: all clients maintain their own set of registered handlers. For example if a handler is registered to one client's event system, it will not be registered to another client's event system:

```
import boto3

client1 = boto3.client('s3')
client2 = boto3.client('s3')

def add_my_bucket(params, **kwargs):
    if 'Bucket' not in params:
        params['Bucket'] = 'mybucket'

def add_my_other_bucket(params, **kwargs):
    if 'Bucket' not in params:
        params['Bucket'] = 'myotherbucket'

client1.meta.events.register(
    'provide-client-params.s3.ListObjects', add_my_bucket)
client2.meta.events.register(
    'provide-client-params.s3.ListObjects', add_my_other_bucket)

client1_response = client1.list_objects()
client2_response = client2.list_objects()
```

Thanks to the isolation of clients' event systems, `client1` will inject 'mybucket' for its `list_objects` method call while `client2` will inject 'myotherbucket' for its `list_objects` method call because `add_my_bucket` was registered to `client1` while `add_my_other_bucket` was registered to `client2`.

Boto3 Specific Events

Boto3 emits a set of events that users can register to customize clients or resources and modify the behavior of method calls.

Here is the list of events that users of boto3 can register handlers to:

- 'creating-client-class
- 'creating-resource-class
- 'provide-client-params'

creating-client-class

Full Event Name 'creating-client-class.service-name'

Note: `service-name` refers to the value used to instantiate a client i.e.
`boto3.client('service-name')`

Description This event is emitted upon creation of the client class for a service. The client class for a service is not created until the first instantiation of the client class. Use this event for adding methods to the client class or adding classes for the client class to inherit from.

Keyword Arguments Emitted

type `class_attributes` dict

param `class_attributes` A dictionary where the keys are the names of the attributes of the class and the values are the actual attributes of the class.

type `base_classes` list

param `base_classes` A list of classes that the client class will inherit from where the order of inheritance is the same as the order of the list.

Expected Return Value Do not return anything.

Example Here is an example of how to add a method to the client class:

```
from boto3.session import Session

def custom_method(self):
    print('This is my custom method')

def add_custom_method(class_attributes, **kwargs):
    class_attributes['my_method'] = custom_method

session = Session()
session.events.register('creating-client-class.s3', add_custom_method)

client = session.client('s3')
client.my_method()
```

This should output:

```
This is my custom method
```

Here is an example of how to add a new class for the client class to inherit from:

```
from boto3.session import Session

class MyClass(object):
    def __init__(self, *args, **kwargs):
        super(MyClass, self).__init__(*args, **kwargs)
        print('Client instantiated!')

def add_custom_class(base_classes, **kwargs):
    base_classes.insert(0, MyClass)

session = Session()
session.events.register('creating-client-class.s3', add_custom_class)

client = session.client('s3')
```

This should output:

```
Client instantiated!
```

creating-resource-class

Full Event Name 'creating-resource-class.service-name.resource-name'

Note: `service-name` refers to the value used to instantiate a service resource i.e. `boto3.resource('service-name')` and `resource-name` refers to the name of the resource class.

Description This event is emitted upon creation of the resource class. The resource class is not created until the first instantiation of the resource class. Use this event for adding methods to the resource class or adding classes for the resource class to inherit from.

Keyword Arguments Emitted

type `class_attributes` dict

param `class_attributes` A dictionary where the keys are the names of the attributes of the class and the values are the actual attributes of the class.

type `base_classes` list

param `base_classes` A list of classes that the resource class will inherit from where the order of inheritance is the same as the order of the list.

Expected Return Value Do not return anything.

Example Here is an example of how to add a method to a resource class:

```
from boto3.session import Session

def custom_method(self):
    print('This is my custom method')

def add_custom_method(class_attributes, **kwargs):
    class_attributes['my_method'] = custom_method

session = Session()
session.events.register('creating-resource-class.s3.ServiceResource',
                        add_custom_method)

resource = session.resource('s3')
resource.my_method()
```

This should output:

```
This is my custom method
```

Here is an example of how to add a new class for a resource class to inherit from:

```
from boto3.session import Session

class MyClass(object):
    def __init__(self, *args, **kwargs):
        super(MyClass, self).__init__(*args, **kwargs)
        print('Resource instantiated!')

def add_custom_class(base_classes, **kwargs):
    base_classes.insert(0, MyClass)

session = Session()
session.events.register('creating-resource-class.s3.ServiceResource',
                        add_custom_class)

resource = session.resource('s3')
```

This should output:

```
Resource instantiated!
```

provide-client-params

Full Event Name 'provide-client.service-name.operation-name'

Note: `service-name` refers to the value used to instantiate a client i.e. `boto3.client('service-name')`. `operation-name` refers to the underlying API operation of the corresponding client method. To access the operation API name, retrieve the value from the `client.meta.method_to_api_mapping` dictionary using the name of the desired client method as the key.

Description This event is emitted before validation of the parameters passed to client method. Use this event to inject or modify parameters prior to the parameters being validated and built into a request that is sent over the wire.

Keyword Arguments Emitted

type params dict

param params A dictionary where the keys are the names of the parameters passed through the client method and the values are the values of those parameters.

type model `botocore.model.OperationModel`

param model A model representing the underlying API operation of the client method.

Expected Return Value Do not return anything or return a new dictionary of parameters to use when making the request.

Example Here is an example of how to inject a parameter using the event:

```
import boto3

s3 = boto3.client('s3')

# Access the event system on the S3 client
event_system = s3.meta.events

# Create a function
def add_my_bucket(params, **kwargs):
    # Add the name of the bucket you want to default to.
    if 'Bucket' not in params:
        params['Bucket'] = 'mybucket'

# Register the function to an event
event_system.register('provide-client-params.s3.ListObjects', add_my_bucket)

response = s3.list_objects()
```

2.1.3 Service Feature Guides

DynamoDB

By following this guide, you will learn how to use the `DynamoDB.ServiceResource` and `DynamoDB.Table` resources in order to create tables, write items to tables, modify existing items, retrieve items, and query/filter the items in the table.

Creating a New Table

In order to create a new table, use the `DynamoDB.ServiceResource.create_table()` method:

```
import boto3

# Get the service resource.
dynamodb = boto3.resource('dynamodb')

# Create the DynamoDB table.
table = dynamodb.create_table(
    TableName='users',
    KeySchema=[
        {
            'AttributeName': 'username',
            'KeyType': 'HASH'
        },
        {
            'AttributeName': 'last_name',
            'KeyType': 'RANGE'
        }
    ],
    AttributeDefinitions=[
        {
            'AttributeName': 'username',
            'AttributeType': 'S'
        },
        {
            'AttributeName': 'last_name',
            'AttributeType': 'S'
        }
    ],
    ProvisionedThroughput={
        'ReadCapacityUnits': 5,
        'WriteCapacityUnits': 5
    }
)

# Wait until the table exists.
table.meta.client.get_waiter('table_exists').wait(TableName='users')

# Print out some data about the table.
print(table.item_count)
```

Expected Output:

```
0
```

This creates a table named `users` that respectively has the hash and range primary keys `username` and `last_name`. This method will return a `DynamoDB.Table` resource to call additional methods on the created table.

Using an Existing Table

It is also possible to create a `DynamoDB.Table` resource from an existing table:


```
import boto3

# Get the service resource.
dynamodb = boto3.resource('dynamodb')

# Instantiate a table resource object without actually
# creating a DynamoDB table. Note that the attributes of this table
# are lazy-loaded: a request is not made nor are the attribute
# values populated until the attributes
# on the table resource are accessed or its load() method is called.
table = dynamodb.Table('users')

# Print out some data about the table.
# This will cause a request to be made to DynamoDB and its attribute
# values will be set based on the response.
print(table.creation_date_time)
```

Expected Output (Please note that probably the actual times will not match up):

```
2015-06-26 12:42:45.149000-07:00
```

Creating a New Item

Once you have a `DynamoDB.Table` resource you can add new items to the table using `DynamoDB.Table.put_item()`:

```
table.put_item(
    Item={
        'username': 'janedoe',
        'first_name': 'Jane',
        'last_name': 'Doe',
        'age': 25,
        'account_type': 'standard_user',
    }
)
```

For all of the valid types that can be used for an item, refer to *Valid DynamoDB Types*.

Getting an Item

You can then retrieve the object using `DynamoDB.Table.get_item()`:

```
response = table.get_item(
    Key={
        'username': 'janedoe',
        'last_name': 'Doe'
    }
)
item = response['Item']
print(item)
```

Expected Output:

```
{u'username': u'janedoe',
 u'first_name': u'Jane',
 u'last_name': u'Doe',
```

```
u'account_type': u'standard_user',
u'age': Decimal('25')}
```

Updating Item

Using the retrieved item, you can update attributes of the item in the table:

```
item['age'] = 26
table.put_item(Item=item)
```

Then if you retrieve the item again, it will be updated appropriately:

```
response = table.get_item(
    Key={
        'username': 'janedoe',
        'last_name': 'Doe'
    }
)
item = response['Item']
print(item)
```

Expected Output:

```
{u'username': u'janedoe',
 u'first_name': u'Jane',
 u'last_name': u'Doe',
 u'account_type': u'standard_user',
 u'age': Decimal('26')}
```

Deleting Item

You can also delete the item using `DynamoDB.Table.delete_item()`:

```
table.delete_item(
    Key={
        'username': 'janedoe',
        'last_name': 'Doe'
    }
)
```

Batch Writing

If you are loading a lot of data at a time, you can make use of `DyanmoDB.Table.batch_writer()` so you can both speed up the process and reduce the number of write requests made to the service.

This method returns a handle to a batch writer object that will automatically handle buffering and sending items in batches. In addition, the batch writer will also automatically handle any unprocessed items and resend them as needed. All you need to do is call `put_item` for any items you want to add, and `delete_item` for any items you want to delete:

```
with table.batch_writer() as batch:
    batch.put_item(
        Item={
            'account_type': 'standard_user',
            'username': 'johndoe',
```

```

        'first_name': 'John',
        'last_name': 'Doe',
        'age': 25,
        'address': {
            'road': '1 Jefferson Street',
            'city': 'Los Angeles',
            'state': 'CA',
            'zipcode': 90001
        }
    )
    batch.put_item(
        Item={
            'account_type': 'super_user',
            'username': 'janedoering',
            'first_name': 'Jane',
            'last_name': 'Doering',
            'age': 40,
            'address': {
                'road': '2 Washington Avenue',
                'city': 'Seattle',
                'state': 'WA',
                'zipcode': 98109
            }
        }
    )
    batch.put_item(
        Item={
            'account_type': 'standard_user',
            'username': 'bobsmith',
            'first_name': 'Bob',
            'last_name': 'Smith',
            'age': 18,
            'address': {
                'road': '3 Madison Lane',
                'city': 'Louisville',
                'state': 'KY',
                'zipcode': 40213
            }
        }
    )
    batch.put_item(
        Item={
            'account_type': 'super_user',
            'username': 'alicedoe',
            'first_name': 'Alice',
            'last_name': 'Doe',
            'age': 27,
            'address': {
                'road': '1 Jefferson Street',
                'city': 'Los Angeles',
                'state': 'CA',
                'zipcode': 90001
            }
        }
    )

```

The batch writer is even able to handle a very large amount of writes to the table.

```
with table.batch_writer() as batch:
    for i in range(50):
        batch.put_item(
            Item={
                'account_type': 'anonymous',
                'username': 'user' + str(i),
                'first_name': 'unknown',
                'last_name': 'unknown'
            }
        )
```

Querying and Scanning

With the table full of items, you can then query or scan the items in the table using the `DynamoDB.Table.query()` or `DynamoDB.Table.scan()` methods respectively. To add conditions to scanning and querying the table, you will need to import the `boto3.dynamodb.conditions.Key` and `boto3.dynamodb.conditions.Attr` classes. The `boto3.dynamodb.conditions.Key` should be used when the condition is related to the key of the item. The `boto3.dynamodb.conditions.Attr` should be used when the condition is related to an attribute of the item:

```
from boto3.dynamodb.conditions import Key, Attr
```

This queries for all of the users whose username key equals johndoe:

```
response = table.query(
    KeyConditionExpression=Key('username').eq('johndoe')
)
items = response['Items']
print(items)
```

Expected Output:

```
[{'username': u'johndoe',
  'first_name': u'John',
  'last_name': u'Doe',
  'account_type': u'standard_user',
  'age': Decimal('25'),
  'address': {'city': u'Los Angeles',
              'state': u'CA',
              'zipcode': Decimal('90001'),
              'road': u'1 Jefferson Street'}}]
```

Similarly you can scan the table based on attributes of the items. For example, this scans for all the users whose age is less than 27:

```
response = table.scan(
    FilterExpression=Attr('age').lt(27)
)
items = response['Items']
print(items)
```

Expected Output:

```
[{'username': u'johndoe',
  'first_name': u'John',
  'last_name': u'Doe',
  'account_type': u'standard_user',
```

```

u'age': Decimal('25'),
u'address': {u'city': u'Los Angeles',
             u'state': u'CA',
             u'zipcode': Decimal('90001'),
             u'road': u'1 Jefferson Street'}},
{u'username': u'bobsmith',
 u'first_name': u'Bob',
 u'last_name': u'Smith',
 u'account_type': u'standard_user',
 u'age': Decimal('18'),
 u'address': {u'city': u'Louisville',
              u'state': u'KY',
              u'zipcode': Decimal('40213'),
              u'road': u'3 Madison Lane'}}}]

```

You are also able to chain conditions together using the logical operators: & (and), | (or), and ~ (not). For example, this scans for all users whose `first_name` starts with J and whose `account_type` is `super_user`:

```

response = table.scan(
    FilterExpression=Attr('first_name').begins_with('J') & Attr('account_type').eq('super_user')
)
items = response['Items']
print(items)

```

Expected Output:

```

[{'username': u'janedoering',
  'first_name': u'Jane',
  'last_name': u'Doering',
  'account_type': u'super_user',
  'age': Decimal('40'),
  'address': {u'city': u'Seattle',
              u'state': u'WA',
              u'zipcode': Decimal('98109'),
              u'road': u'2 Washington Avenue'}}}]

```

You can even scan based on conditions of a nested attribute. For example this scans for all users whose state in their address is CA:

```

response = table.scan(
    FilterExpression=Attr('address.state').eq('CA')
)
items = response['Items']
print(items)

```

Expected Output:

```

[{'username': u'johndoe',
  'first_name': u'John',
  'last_name': u'Doe',
  'account_type': u'standard_user',
  'age': Decimal('25'),
  'address': {u'city': u'Los Angeles',
              u'state': u'CA',
              u'zipcode': Decimal('90001'),
              u'road': u'1 Jefferson Street'}},
 {'username': u'alicedoe',
  'first_name': u'Alice',
  'last_name': u'Doe',

```

```
u'account_type': u'super_user',
u'age': Decimal('27'),
u'address': {u'city': u'Los Angeles',
             u'state': u'CA',
             u'zipcode': Decimal('90001'),
             u'road': u'1 Jefferson Street'}}
```

For more information on the various conditions you can use for queries and scans, refer to *DynamoDB Conditions*.

Deleting a Table

Finally, if you want to delete your table call `DynamoDB.Table.delete()`:

```
table.delete()
```

API Reference

3.1 Services

3.1.1 Available Services

AutoScaling

Table of Contents

- AutoScaling
 - Client
 - Paginators

Client

class AutoScaling.**Client**

A low-level client representing Auto Scaling:

```
import boto3
```

```
client = boto3.client('autoscaling')
```

These are the available methods:

- `attach_instances()`
- `attach_load_balancers()`
- `can_paginate()`
- `complete_lifecycle_action()`
- `create_auto_scaling_group()`
- `create_launch_configuration()`
- `create_or_update_tags()`
- `delete_auto_scaling_group()`
- `delete_launch_configuration()`

- `delete_lifecycle_hook()`
- `delete_notification_configuration()`
- `delete_policy()`
- `delete_scheduled_action()`
- `delete_tags()`
- `describe_account_limits()`
- `describe_adjustment_types()`
- `describe_auto_scaling_groups()`
- `describe_auto_scaling_instances()`
- `describe_auto_scaling_notification_types()`
- `describe_launch_configurations()`
- `describe_lifecycle_hook_types()`
- `describe_lifecycle_hooks()`
- `describe_load_balancers()`
- `describe_metric_collection_types()`
- `describe_notification_configurations()`
- `describe_policies()`
- `describe_scaling_activities()`
- `describe_scaling_process_types()`
- `describe_scheduled_actions()`
- `describe_tags()`
- `describe_termination_policy_types()`
- `detach_instances()`
- `detach_load_balancers()`
- `disable_metrics_collection()`
- `enable_metrics_collection()`
- `enter_standby()`
- `execute_policy()`
- `exit_standby()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_waiter()`
- `put_lifecycle_hook()`
- `put_notification_configuration()`
- `put_scaling_policy()`
- `put_scheduled_update_group_action()`

- `record_lifecycle_action_heartbeat()`
- `resume_processes()`
- `set_desired_capacity()`
- `set_instance_health()`
- `suspend_processes()`
- `terminate_instance_in_auto_scaling_group()`
- `update_auto_scaling_group()`

attach_instances (**kwargs)

Attaches one or more EC2 instances to the specified Auto Scaling group.

For more information, see [Attach EC2 Instances to Your Auto Scaling Group](#) in the *Auto Scaling Developer Guide*.

Request Syntax

```
response = client.attach_instances(
    InstanceIds=[
        'string',
    ],
    AutoScalingGroupName='string'
)
```

Parameters

- **InstanceIds** (*list*) – One or more EC2 instance IDs.
– (*string*) –
- **AutoScalingGroupName** (*string*) – **[REQUIRED]**
The name of the group.

Returns None

attach_load_balancers (**kwargs)

Attaches one or more load balancers to the specified Auto Scaling group.

To describe the load balancers for an Auto Scaling group, use `DescribeLoadBalancers`. To detach the load balancer from the Auto Scaling group, use `DetachLoadBalancers`.

For more information, see [Attach a Load Balancer to Your Auto Scaling Group](#) in the *Auto Scaling Developer Guide*.

Request Syntax

```
response = client.attach_load_balancers(
    AutoScalingGroupName='string',
    LoadBalancerNames=[
        'string',
    ]
)
```

Parameters

- **AutoScalingGroupName** (*string*) – The name of the group.
- **LoadBalancerNames** (*list*) – One or more load balancer names.

– (*string*) –

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

complete_lifecycle_action (***kwargs*)

Completes the lifecycle action for the associated token initiated under the given lifecycle hook with the specified result.

This operation is a part of the basic sequence for adding a lifecycle hook to an Auto Scaling group:

- Create a notification target. A target can be either an Amazon SQS queue or an Amazon SNS topic.
- Create an IAM role. This role allows Auto Scaling to publish lifecycle notifications to the designated SQS queue or SNS topic.
- Create the lifecycle hook. You can create a hook that acts when instances launch or when instances terminate.
- If necessary, record the lifecycle action heartbeat to keep the instance in a pending state.
- **Complete the lifecycle action** .

For more information, see [Auto Scaling Pending State](#) and [Auto Scaling Terminating State](#) in the *Auto Scaling Developer Guide* .

Request Syntax

```
response = client.complete_lifecycle_action(  
    LifecycleHookName='string',  
    AutoScalingGroupName='string',  
    LifecycleActionToken='string',  
    LifecycleActionResult='string'  
)
```

Parameters

- **LifecycleHookName** (*string*) – [REQUIRED]
The name of the lifecycle hook.
- **AutoScalingGroupName** (*string*) – [REQUIRED]
The name of the group for the lifecycle hook.

- **LifecycleActionToken** (*string*) – [REQUIRED]

A universally unique identifier (UUID) that identifies a specific lifecycle action associated with an instance. Auto Scaling sends this token to the notification target you specified when you created the lifecycle hook.

- **LifecycleActionResult** (*string*) – [REQUIRED]

The action for the group to take. This parameter can be either `CONTINUE` or `ABANDON`.

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –

create_auto_scaling_group (***kwargs*)

Creates an Auto Scaling group with the specified name and attributes.

If you exceed your maximum limit of Auto Scaling groups, which by default is 20 per region, the call fails. For information about viewing and updating this limit, see [DescribeAccountLimits](#).

For more information, see [Auto Scaling Groups](#) in the *Auto Scaling Developer Guide*.

Request Syntax

```
response = client.create_auto_scaling_group(
    AutoScalingGroupName='string',
    LaunchConfigurationName='string',
    InstanceId='string',
    MinSize=123,
    MaxSize=123,
    DesiredCapacity=123,
    DefaultCooldown=123,
    AvailabilityZones=[
        'string',
    ],
    LoadBalancerNames=[
        'string',
    ],
    HealthCheckType='string',
    HealthCheckGracePeriod=123,
    PlacementGroup='string',
    VPCZoneIdentifier='string',
    TerminationPolicies=[
        'string',
    ],
    Tags=[
        {
            'ResourceId': 'string',
            'ResourceType': 'string',
            'Key': 'string',
            'Value': 'string',
            'PropagateAtLaunch': True|False
        },
    ],
)
```

```
]
)
```

Parameters

- **AutoScalingGroupName** (*string*) – [REQUIRED]

The name of the group. This name must be unique within the scope of your AWS account.

- **LaunchConfigurationName** (*string*) – The name of the launch configuration. Alternatively, use the `InstanceId` parameter to specify an EC2 instance instead of a launch configuration.
- **InstanceId** (*string*) – The ID of the EC2 instance used to create a launch configuration for the group. Alternatively, use the `LaunchConfigurationName` parameter to specify a launch configuration instead of an EC2 instance.

When you specify an ID of an instance, Auto Scaling creates a new launch configuration and associates it with the group. This launch configuration derives its attributes from the specified instance, with the exception of the block device mapping.

For more information, see [Create an Auto Scaling Group from an EC2 Instance](#) in the *Auto Scaling Developer Guide* .

- **MinSize** (*integer*) – [REQUIRED]

The minimum size of the group.

- **MaxSize** (*integer*) – [REQUIRED]

The maximum size of the group.

- **DesiredCapacity** (*integer*) – The number of EC2 instances that should be running in the group. This number must be greater than or equal to the minimum size of the group and less than or equal to the maximum size of the group.
- **DefaultCooldown** (*integer*) – The amount of time, in seconds, after a scaling activity completes before another scaling activity can start.

If this parameter is not specified, the default value is 300. For more information, see [Understanding Auto Scaling Cooldowns](#) in the *Auto Scaling Developer Guide* .

- **AvailabilityZones** (*list*) – One or more Availability Zones for the group. This parameter is optional if you specify subnets using the `VPCZoneIdentifier` parameter.

– (*string*) –

- **LoadBalancerNames** (*list*) – One or more load balancers.

For more information, see [Load Balance Your Auto Scaling Group](#) in the *Auto Scaling Developer Guide* .

– (*string*) –

- **HealthCheckType** (*string*) – The service to use for the health checks. The valid values are EC2 and ELB .

By default, health checks use Amazon EC2 instance status checks to determine the health of an instance. For more information, see [Health Checks](#) .

- **HealthCheckGracePeriod** (*integer*) – The amount of time, in seconds, after an EC2 instance comes into service that Auto Scaling starts checking its health. During this time, any health check failures for the instance are ignored.

This parameter is required if you are adding an ELB health check. Frequently, new instances need to warm up, briefly, before they can pass a health check. To provide ample warm-up time, set the health check grace period of the group to match the expected startup period of your application.

For more information, see [Add an Elastic Load Balancing Health Check to Your Auto Scaling Group](#) in the *Auto Scaling Developer Guide*.

- **PlacementGroup** (*string*) – The name of the placement group into which you’ll launch your instances, if any. For more information, see [Placement Groups](#) in the *Amazon Elastic Compute Cloud User Guide*.
- **VPCZoneIdentifier** (*string*) – A comma-separated list of subnet identifiers for your virtual private cloud (VPC).

If you specify subnets and Availability Zones with this call, ensure that the subnets’ Availability Zones match the Availability Zones specified.

For more information, see [Auto Scaling and Amazon Virtual Private Cloud](#) in the *Auto Scaling Developer Guide*.

- **TerminationPolicies** (*list*) – One or more termination policies used to select the instance to terminate. These policies are executed in the order that they are listed.

For more information, see [Choosing a Termination Policy for Your Auto Scaling Group](#) in the *Auto Scaling Developer Guide*.

– (*string*) –

- **Tags** (*list*) – The tag to be created or updated. Each tag should be defined by its resource type, resource ID, key, value, and a propagate flag. Valid values: key=**value**, value=**value**, propagate=**true** or *false*. Value and propagate are optional parameters.

For more information, see [Tagging Auto Scaling Groups and Instances](#) in the *Auto Scaling Developer Guide*.

– (*dict*) –

Describes a tag for an Auto Scaling group.

* **ResourceId** (*string*) –

The name of the group.

* **ResourceType** (*string*) –

The type of resource. The only supported value is `auto-scaling-group`.

* **Key** (*string*) – [REQUIRED]

The tag key.

* **Value** (*string*) –

The tag value.

* **PropagateAtLaunch** (*boolean*) –

Determines whether the tag is added to new instances as they are launched in the group.

Returns None

create_launch_configuration (***kwargs*)

Creates a launch configuration.

If you exceed your maximum limit of launch configurations, which by default is 100 per region, the call fails. For information about viewing and updating this limit, see [DescribeAccountLimits](#) .

For more information, see [Launch Configurations](#) in the *Auto Scaling Developer Guide* .

Request Syntax

```
response = client.create_launch_configuration(
    LaunchConfigurationName='string',
    ImageId='string',
    KeyName='string',
    SecurityGroups=[
        'string',
    ],
    ClassicLinkVPCId='string',
    ClassicLinkVPCSecurityGroups=[
        'string',
    ],
    UserData='string',
    InstanceId='string',
    InstanceType='string',
    KernelId='string',
    RamdiskId='string',
    BlockDeviceMappings=[
        {
            'VirtualName': 'string',
            'DeviceName': 'string',
            'Ebs': {
                'SnapshotId': 'string',
                'VolumeSize': 123,
                'VolumeType': 'string',
                'DeleteOnTermination': True|False,
                'Iops': 123
            },
            'NoDevice': True|False
        },
    ],
    InstanceMonitoring={
        'Enabled': True|False
    },
    SpotPrice='string',
    IamInstanceProfile='string',
    EbsOptimized=True|False,
    AssociatePublicIpAddress=True|False,
    PlacementTenancy='string'
)
```

Parameters

- **LaunchConfigurationName** (*string*) – [REQUIRED]

The name of the launch configuration. This name must be unique within the scope of your AWS account.

- **ImageId** (*string*) – The ID of the Amazon Machine Image (AMI) to use to launch your EC2 instances. For more information, see [Finding an AMI](#) in the *Amazon Elastic Compute Cloud User Guide* .

- **KeyName** (*string*) – The name of the key pair. For more information, see [Amazon EC2 Key Pairs](#) in the *Amazon Elastic Compute Cloud User Guide* .

- **SecurityGroups** (*list*) – One or more security groups with which to associate the instances.

If your instances are launched in EC2-Classic, you can either specify security group names or the security group IDs. For more information about security groups for EC2-Classic, see [Amazon EC2 Security Groups](#) in the *Amazon Elastic Compute Cloud User Guide*.

If your instances are launched into a VPC, specify security group IDs. For more information, see [Security Groups for Your VPC](#) in the *Amazon Virtual Private Cloud User Guide*.

– (*string*) –

- **ClassicLinkVPCId** (*string*) – The ID of a ClassicLink-enabled VPC to link your EC2-Classic instances to. This parameter is supported only if you are launching EC2-Classic instances. For more information, see [ClassicLink](#) in the *Amazon Elastic Compute Cloud User Guide*.
- **ClassicLinkVPCSecurityGroups** (*list*) – The IDs of one or more security groups for the VPC specified in `ClassicLinkVPCId`. This parameter is required if `ClassicLinkVPCId` is specified, and is not supported otherwise. For more information, see [ClassicLink](#) in the *Amazon Elastic Compute Cloud User Guide*.

– (*string*) –

- **UserData** (*string*) – The user data to make available to the launched EC2 instances. For more information, see [Instance Metadata and User Data](#) in the *Amazon Elastic Compute Cloud User Guide*.

At this time, launch configurations don't support compressed (zipped) user data files.

This value will be base64 encoded automatically. Do not base64 encode this value prior to performing the operation.

- **InstanceId** (*string*) – The ID of the EC2 instance to use to create the launch configuration.

The new launch configuration derives attributes from the instance, with the exception of the block device mapping.

To create a launch configuration with a block device mapping or override any other instance attributes, specify them as part of the same request.

For more information, see [Create a Launch Configuration Using an EC2 Instance](#) in the *Auto Scaling Developer Guide*.

- **InstanceType** (*string*) – The instance type of the EC2 instance. For information about available instance types, see [Available Instance Types](#) in the *Amazon Elastic Cloud Compute User Guide*.
- **KernelId** (*string*) – The ID of the kernel associated with the AMI.
- **RamdiskId** (*string*) – The ID of the RAM disk associated with the AMI.
- **BlockDeviceMappings** (*list*) – One or more mappings that specify how block devices are exposed to the instance. For more information, see [Block Device Mapping](#) in the *Amazon Elastic Compute Cloud User Guide*.

– (*dict*) –

Describes a block device mapping.

- * **VirtualName** (*string*) –

The name of the virtual device, `ephemeral0` to `ephemeral3`.

- * **DeviceName** (*string*) – [REQUIRED]

The device name exposed to the EC2 instance (for example, `/dev/sdh` or `xvdh`).

- * **Ebs** (*dict*) –

The information about the Amazon EBS volume.

- **SnapshotId** (*string*) –

The ID of the snapshot.

- **VolumeSize** (*integer*) –

The volume size, in gigabytes.

Valid values: If the volume type is `io1`, the minimum size of the volume is 10 GiB. If you specify `SnapshotId` and `VolumeSize`, `VolumeSize` must be equal to or larger than the size of the snapshot.

Default: If you create a volume from a snapshot and you don't specify a volume size, the default is the size of the snapshot.

Required: Required when the volume type is `io1`.

- **VolumeType** (*string*) –

The volume type.

Valid values: `standard` | `io1` | `gp2`

Default: `standard`

- **DeleteOnTermination** (*boolean*) –

Indicates whether to delete the volume on instance termination.

Default: `true`

- **Iops** (*integer*) –

For Provisioned IOPS (SSD) volumes only. The number of I/O operations per second (IOPS) to provision for the volume.

Valid values: Range is 100 to 4000.

Default: `None`

- * **NoDevice** (*boolean*) –

Suppresses a device mapping.

If this parameter is true for the root device, the instance might fail the EC2 health check. Auto Scaling launches a replacement instance if the instance fails the health check.

- **InstanceMonitoring** (*dict*) – Enables detailed monitoring if it is disabled. Detailed monitoring is enabled by default.

When detailed monitoring is enabled, Amazon CloudWatch generates metrics every minute and your account is charged a fee. When you disable detailed monitoring, by specifying `False`, CloudWatch generates metrics every 5 minutes. For more

information, see [Monitor Your Auto Scaling Instances](#) in the *Auto Scaling Developer Guide*.

– **Enabled** (*boolean*) –

If `True`, instance monitoring is enabled.

- **SpotPrice** (*string*) – The maximum hourly price to be paid for any Spot Instance launched to fulfill the request. Spot Instances are launched when the price you specify exceeds the current Spot market price. For more information, see [Launch Spot Instances in Your Auto Scaling Group](#) in the *Auto Scaling Developer Guide*.
- **IamInstanceProfile** (*string*) – The name or the Amazon Resource Name (ARN) of the instance profile associated with the IAM role for the instance.

EC2 instances launched with an IAM role will automatically have AWS security credentials available. You can use IAM roles with Auto Scaling to automatically enable applications running on your EC2 instances to securely access other AWS resources. For more information, see [Launch Auto Scaling Instances with an IAM Role](#) in the *Auto Scaling Developer Guide*.

- **EbsOptimized** (*boolean*) – Indicates whether the instance is optimized for Amazon EBS I/O. By default, the instance is not optimized for EBS I/O. The optimization provides dedicated throughput to Amazon EBS and an optimized configuration stack to provide optimal I/O performance. This optimization is not available with all instance types. Additional usage charges apply. For more information, see [Amazon EBS-Optimized Instances](#) in the *Amazon Elastic Compute Cloud User Guide*.
- **AssociatePublicIpAddress** (*boolean*) – Used for groups that launch instances into a virtual private cloud (VPC). Specifies whether to assign a public IP address to each instance. For more information, see [Auto Scaling and Amazon Virtual Private Cloud](#) in the *Auto Scaling Developer Guide*.

If you specify a value for this parameter, be sure to specify at least one subnet using the `VPCZoneIdentifier` parameter when you create your group.

Default: If the instance is launched into a default subnet, the default is `true`. If the instance is launched into a nondefault subnet, the default is `false`. For more information, see [Supported Platforms](#) in the *Amazon Elastic Compute Cloud User Guide*.

- **PlacementTenancy** (*string*) – The tenancy of the instance. An instance with a tenancy of `dedicated` runs on single-tenant hardware and can only be launched into a VPC.

You must set the value of this parameter to `dedicated` if want to launch Dedicated Instances into a shared tenancy VPC (VPC with instance placement tenancy attribute set to `default`).

If you specify a value for this parameter, be sure to specify at least one subnet using the `VPCZoneIdentifier` parameter when you create your group.

For more information, see [Auto Scaling and Amazon Virtual Private Cloud](#) in the *Auto Scaling Developer Guide*.

Valid values: `default` | `dedicated`

Returns None

create_or_update_tags (***kwargs*)

Creates or updates tags for the specified Auto Scaling group.

A tag is defined by its resource ID, resource type, key, value, and propagate flag. The value and the propagate flag are optional parameters. The only supported resource type is `auto-scaling-group`, and the resource ID must be the name of the group. The `PropagateAtLaunch` flag determines whether the tag is added to instances launched in the group. Valid values are `true` or `false`.

When you specify a tag with a key that already exists, the operation overwrites the previous tag definition, and you do not get an error message.

For more information, see [Tagging Auto Scaling Groups and Instances](#) in the *Auto Scaling Developer Guide*.

Request Syntax

```
response = client.create_or_update_tags(
    Tags=[
        {
            'ResourceId': 'string',
            'ResourceType': 'string',
            'Key': 'string',
            'Value': 'string',
            'PropagateAtLaunch': True|False
        },
    ]
)
```

Parameters `Tags (list)` – [REQUIRED]

One or more tags.

- *(dict)* –

Describes a tag for an Auto Scaling group.

- **ResourceId** (*string*) –

The name of the group.

- **ResourceType** (*string*) –

The type of resource. The only supported value is `auto-scaling-group`.

- **Key** (*string*) – [REQUIRED]

The tag key.

- **Value** (*string*) –

The tag value.

- **PropagateAtLaunch** (*boolean*) –

Determines whether the tag is added to new instances as they are launched in the group.

Returns `None`

`delete_auto_scaling_group` (***kwargs*)

Deletes the specified Auto Scaling group.

The group must have no instances and no scaling activities in progress.

To remove all instances before calling `DeleteAutoScalingGroup`, call `UpdateAutoScalingGroup` to set the minimum and maximum size of the Auto Scaling group to zero.

Request Syntax

```
response = client.delete_auto_scaling_group(
    AutoScalingGroupName='string',
    ForceDelete=True|False
)
```

Parameters

- **AutoScalingGroupName** (*string*) – [REQUIRED]

The name of the group to delete.

- **ForceDelete** (*boolean*) – Specifies that the group will be deleted along with all instances associated with the group, without waiting for all instances to be terminated. This parameter also deletes any lifecycle actions associated with the group.

Returns None

delete_launch_configuration (***kwargs*)

Deletes the specified launch configuration.

The launch configuration must not be attached to an Auto Scaling group. When this call completes, the launch configuration is no longer available for use.

Request Syntax

```
response = client.delete_launch_configuration(
    LaunchConfigurationName='string'
)
```

Parameters **LaunchConfigurationName** (*string*) – [REQUIRED]

The name of the launch configuration.

Returns None

delete_lifecycle_hook (***kwargs*)

Deletes the specified lifecycle hook.

If there are any outstanding lifecycle actions, they are completed first (ABANDON for launching instances, CONTINUE for terminating instances).

Request Syntax

```
response = client.delete_lifecycle_hook(
    LifecycleHookName='string',
    AutoScalingGroupName='string'
)
```

Parameters

- **LifecycleHookName** (*string*) – [REQUIRED]

The name of the lifecycle hook.

- **AutoScalingGroupName** (*string*) – [REQUIRED]

The name of the Auto Scaling group for the lifecycle hook.

Return type dict

Returns**Response Syntax**

```
{}
```

Response Structure

- (*dict*) –

delete_notification_configuration (***kwargs*)

Deletes the specified notification.

Request Syntax

```
response = client.delete_notification_configuration(  
    AutoScalingGroupName='string',  
    TopicARN='string'  
)
```

Parameters

- **AutoScalingGroupName** (*string*) – [REQUIRED]
The name of the Auto Scaling group.
- **TopicARN** (*string*) – [REQUIRED]
The Amazon Resource Name (ARN) of the Amazon Simple Notification Service (SNS) topic.

Returns None

delete_policy (***kwargs*)

Deletes the specified Auto Scaling policy.

Request Syntax

```
response = client.delete_policy(  
    AutoScalingGroupName='string',  
    PolicyName='string'  
)
```

Parameters

- **AutoScalingGroupName** (*string*) – The name of the Auto Scaling group.
- **PolicyName** (*string*) – [REQUIRED]
The name or Amazon Resource Name (ARN) of the policy.

Returns None

delete_scheduled_action (***kwargs*)

Deletes the specified scheduled action.

Request Syntax

```
response = client.delete_scheduled_action(  
    AutoScalingGroupName='string',  
    ScheduledActionName='string'  
)
```

Parameters

- **AutoScalingGroupName** (*string*) – The name of the Auto Scaling group.
- **ScheduledActionName** (*string*) – **[REQUIRED]**

The name of the action to delete.

Returns None

delete_tags (***kwargs*)

Deletes the specified tags.

Request Syntax

```
response = client.delete_tags(
    Tags=[
        {
            'ResourceId': 'string',
            'ResourceType': 'string',
            'Key': 'string',
            'Value': 'string',
            'PropagateAtLaunch': True|False
        },
    ]
)
```

Parameters **Tags** (*list*) – **[REQUIRED]**

Each tag should be defined by its resource type, resource ID, key, value, and a propagate flag. Valid values are: Resource type = *auto-scaling-group* , Resource ID = *AutoScalingGroupName* , key=**value** , value=**value** , propagate=**true** or *false* .

- (*dict*) –

Describes a tag for an Auto Scaling group.

- **ResourceId** (*string*) –

The name of the group.

- **ResourceType** (*string*) –

The type of resource. The only supported value is *auto-scaling-group* .

- **Key** (*string*) – **[REQUIRED]**

The tag key.

- **Value** (*string*) –

The tag value.

- **PropagateAtLaunch** (*boolean*) –

Determines whether the tag is added to new instances as they are launched in the group.

Returns None

describe_account_limits ()

Describes the current Auto Scaling resource limits for your AWS account.

For information about requesting an increase in these limits, see [AWS Service Limits](#) in the *Amazon Web Services General Reference* .

Request Syntax

```
response = client.describe_account_limits()
```

Return type dict

Returns

Response Syntax

```
{
    'MaxNumberOfAutoScalingGroups': 123,
    'MaxNumberOfLaunchConfigurations': 123
}
```

Response Structure

- (dict) –
 - **MaxNumberOfAutoScalingGroups** (*integer*) –
The maximum number of groups allowed for your AWS account. The default limit is 20 per region.
 - **MaxNumberOfLaunchConfigurations** (*integer*) –
The maximum number of launch configurations allowed for your AWS account. The default limit is 100 per region.

describe_adjustment_types()

Describes the policy adjustment types for use with PutScalingPolicy .

Request Syntax

```
response = client.describe_adjustment_types()
```

Return type dict

Returns

Response Syntax

```
{
    'AdjustmentTypes': [
        {
            'AdjustmentType': 'string'
        },
    ]
}
```

Response Structure

- (dict) –
 - **AdjustmentTypes** (*list*) –
The policy adjustment types.
 - * (dict) –
Describes a policy adjustment type.

For more information, see [Dynamic Scaling](#) in the *Auto Scaling Developer Guide*.

• **AdjustmentType** (*string*) –

The policy adjustment type. The valid values are `ChangeInCapacity`, `ExactCapacity`, and `PercentChangeInCapacity`.

describe_auto_scaling_groups (***kwargs*)

Describes one or more Auto Scaling groups. If a list of names is not provided, the call describes all Auto Scaling groups.

Request Syntax

```
response = client.describe_auto_scaling_groups(
    AutoScalingGroupNames=[
        'string',
    ],
    NextToken='string',
    MaxRecords=123
)
```

Parameters

- **AutoScalingGroupNames** (*list*) – The group names.
– (*string*) –
- **NextToken** (*string*) – The token for the next set of items to return. (You received this token from a previous call.)
- **MaxRecords** (*integer*) – The maximum number of items to return with this call.

Return type dict

Returns

Response Syntax

```
{
    'AutoScalingGroups': [
        {
            'AutoScalingGroupName': 'string',
            'AutoScalingGroupARN': 'string',
            'LaunchConfigurationName': 'string',
            'MinSize': 123,
            'MaxSize': 123,
            'DesiredCapacity': 123,
            'DefaultCooldown': 123,
            'AvailabilityZones': [
                'string',
            ],
            'LoadBalancerNames': [
                'string',
            ],
            'HealthCheckType': 'string',
            'HealthCheckGracePeriod': 123,
            'Instances': [
                {
                    'InstanceId': 'string',
                    'AvailabilityZone': 'string',
                    'LifecycleState': 'Pending'|'Pending:Wait'|'Pending:Proceed'|'Q
```

```
        'HealthStatus': 'string',
        'LaunchConfigurationName': 'string'
    },
],
'CreateTime': datetime(2015, 1, 1),
'SuspendedProcesses': [
    {
        'ProcessName': 'string',
        'SuspensionReason': 'string'
    },
],
'PlacementGroup': 'string',
'VPCZoneIdentifier': 'string',
'EnabledMetrics': [
    {
        'Metric': 'string',
        'Granularity': 'string'
    },
],
'Status': 'string',
'Tags': [
    {
        'ResourceId': 'string',
        'ResourceType': 'string',
        'Key': 'string',
        'Value': 'string',
        'PropagateAtLaunch': True|False
    },
],
'TerminationPolicies': [
    'string',
]
},
],
'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

- **AutoScalingGroups** (*list*) –

The groups.

- * (*dict*) –

Describes an Auto Scaling group.

- **AutoScalingGroupName** (*string*) –

The name of the group.

- **AutoScalingGroupARN** (*string*) –

The Amazon Resource Name (ARN) of the group.

- **LaunchConfigurationName** (*string*) –

The name of the associated launch configuration.

- **MinSize** (*integer*) –

The minimum size of the group.

- **MaxSize** (*integer*) –

The maximum size of the group.

- **DesiredCapacity** (*integer*) –

The desired size of the group.

- **DefaultCooldown** (*integer*) –

The number of seconds after a scaling activity completes before any further scaling activities can start.

- **AvailabilityZones** (*list*) –

One or more Availability Zones for the group.

- (*string*) –

- **LoadBalancerNames** (*list*) –

One or more load balancers associated with the group.

- (*string*) –

- **HealthCheckType** (*string*) –

The service of interest for the health status check, which can be either EC2 for Amazon EC2 or ELB for Elastic Load Balancing.

- **HealthCheckGracePeriod** (*integer*) –

The amount of time that Auto Scaling waits before checking an instance's health status. The grace period begins when an instance comes into service.

- **Instances** (*list*) –

The EC2 instances associated with the group.

- (*dict*) –

Describes an EC2 instance.

- **InstanceId** (*string*) –

The ID of the instance.

- **AvailabilityZone** (*string*) –

The Availability Zone in which the instance is running.

- **LifecycleState** (*string*) –

A description of the current lifecycle state. Note that the `Quarantined` state is not used.

- **HealthStatus** (*string*) –

The health status of the instance.

- **LaunchConfigurationName** (*string*) –

The launch configuration associated with the instance.

- **CreatedTime** (*datetime*) –

The date and time the group was created.

- **SuspendedProcesses** (*list*) –
The suspended processes associated with the group.
- (*dict*) –
Describes an Auto Scaling process that has been suspended. For more information, see `ProcessType` .
- **ProcessName** (*string*) –
The name of the suspended process.
- **SuspensionReason** (*string*) –
The reason that the process was suspended.
- **PlacementGroup** (*string*) –
The name of the placement group into which you'll launch your instances, if any. For more information, see [Placement Groups](#) .
- **VPCZoneIdentifier** (*string*) –
One or more subnet IDs, if applicable, separated by commas.

If you specify `VPCZoneIdentifier` and `AvailabilityZones` , ensure that the `Availability Zones` of the subnets match the values for `AvailabilityZones` .
- **EnabledMetrics** (*list*) –
The metrics enabled for the group.
- (*dict*) –
Describes an enabled metric.
- **Metric** (*string*) –
The name of the metric.
- `GroupMinSize`
- `GroupMaxSize`
- `GroupDesiredCapacity`
- `GroupInServiceInstances`
- `GroupPendingInstances`
- `GroupStandbyInstances`
- `GroupTerminatingInstances`
- `GroupTotalInstances`
- **Granularity** (*string*) –
The granularity of the metric. The only valid value is `1Minute` .
- **Status** (*string*) –
The current state of the group when `DeleteAutoScalingGroup` is in progress.
- **Tags** (*list*) –
The tags for the group.

- *(dict)* –
Describes a tag for an Auto Scaling group.
- **ResourceId** (*string*) –
The name of the group.
- **ResourceType** (*string*) –
The type of resource. The only supported value is `auto-scaling-group`.
- **Key** (*string*) –
The tag key.
- **Value** (*string*) –
The tag value.
- **PropagateAtLaunch** (*boolean*) –
Determines whether the tag is added to new instances as they are launched in the group.
- **TerminationPolicies** (*list*) –
The termination policies for the group.
- (*string*) –
- **NextToken** (*string*) –
The token to use when requesting the next set of items. If there are no additional items to return, the string is empty.

describe_auto_scaling_instances (***kwargs*)

Describes one or more Auto Scaling instances. If a list is not provided, the call describes all instances.

Request Syntax

```
response = client.describe_auto_scaling_instances(
    InstanceIds=[
        'string',
    ],
    MaxRecords=123,
    NextToken='string'
)
```

Parameters

- **InstanceIds** (*list*) – One or more Auto Scaling instances to describe, up to 50 instances. If you omit this parameter, all Auto Scaling instances are described. If you specify an ID that does not exist, it is ignored with no error.
– (*string*) –
- **MaxRecords** (*integer*) – The maximum number of items to return with this call.
- **NextToken** (*string*) – The token for the next set of items to return. (You received this token from a previous call.)

Return type dict

Returns

Response Syntax

```
{
  'AutoScalingInstances': [
    {
      'InstanceId': 'string',
      'AutoScalingGroupName': 'string',
      'AvailabilityZone': 'string',
      'LifecycleState': 'string',
      'HealthStatus': 'string',
      'LaunchConfigurationName': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- *(dict)* –
 - **AutoScalingInstances** (*list*) –

The instances.

 - * *(dict)* –

Describes an EC2 instance associated with an Auto Scaling group.

 - **InstanceId** (*string*) –

The ID of the instance.
 - **AutoScalingGroupName** (*string*) –

The name of the Auto Scaling group associated with the instance.
 - **AvailabilityZone** (*string*) –

The Availability Zone for the instance.
 - **LifecycleState** (*string*) –

The lifecycle state for the instance. For more information, see [Auto Scaling Instance States](#) in the *Auto Scaling Developer Guide* .
 - **HealthStatus** (*string*) –

The health status of this instance. “Healthy” means that the instance is healthy and should remain in service. “Unhealthy” means that the instance is unhealthy and Auto Scaling should terminate and replace it.
 - **LaunchConfigurationName** (*string*) –

The launch configuration associated with the instance.
 - **NextToken** (*string*) –

The token to use when requesting the next set of items. If there are no additional items to return, the string is empty.

describe_auto_scaling_notification_types()

Describes the notification types that are supported by Auto Scaling.

Request Syntax

```
response = client.describe_auto_scaling_notification_types()
```

Return type dict

Returns

Response Syntax

```
{
    'AutoScalingNotificationTypes': [
        'string',
    ]
}
```

Response Structure

- (*dict*) –
 - **AutoScalingNotificationTypes** (*list*) –
 - One or more of the following notification types:
 - * `autoscaling:EC2_INSTANCE_LAUNCH`
 - * `autoscaling:EC2_INSTANCE_LAUNCH_ERROR`
 - * `autoscaling:EC2_INSTANCE_TERMINATE`
 - * `autoscaling:EC2_INSTANCE_TERMINATE_ERROR`
 - * `autoscaling:TEST_NOTIFICATION`
 - * (*string*) –

describe_launch_configurations (***kwargs*)

Describes one or more launch configurations. If you omit the list of names, then the call describes all launch configurations.

Request Syntax

```
response = client.describe_launch_configurations(
    LaunchConfigurationNames=[
        'string',
    ],
    NextToken='string',
    MaxRecords=123
)
```

Parameters

- **LaunchConfigurationNames** (*list*) – The launch configuration names.
 - (*string*) –
- **NextToken** (*string*) – The token for the next set of items to return. (You received this token from a previous call.)
- **MaxRecords** (*integer*) – The maximum number of items to return with this call. The default is 100.

Return type dict

Returns

Response Syntax

```
{
    'LaunchConfigurations': [
        {
            'LaunchConfigurationName': 'string',
            'LaunchConfigurationARN': 'string',
            'ImageId': 'string',
            'KeyName': 'string',
            'SecurityGroups': [
                'string',
            ]
        }
    ]
}
```

```
],
  'ClassicLinkVPCId': 'string',
  'ClassicLinkVPCSecurityGroups': [
    'string',
  ],
  'UserData': 'string',
  'InstanceType': 'string',
  'KernelId': 'string',
  'RamdiskId': 'string',
  'BlockDeviceMappings': [
    {
      'VirtualName': 'string',
      'DeviceName': 'string',
      'Ebs': {
        'SnapshotId': 'string',
        'VolumeSize': 123,
        'VolumeType': 'string',
        'DeleteOnTermination': True|False,
        'Iops': 123
      },
      'NoDevice': True|False
    },
  ],
  'InstanceMonitoring': {
    'Enabled': True|False
  },
  'SpotPrice': 'string',
  'IamInstanceProfile': 'string',
  'CreatedTime': datetime(2015, 1, 1),
  'EbsOptimized': True|False,
  'AssociatePublicIpAddress': True|False,
  'PlacementTenancy': 'string'
},
],
'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

- **LaunchConfigurations** (*list*) –

The launch configurations.

- * (*dict*) –

Describes a launch configuration.

- **LaunchConfigurationName** (*string*) –

The name of the launch configuration.

- **LaunchConfigurationARN** (*string*) –

The Amazon Resource Name (ARN) of the launch configuration.

- **ImageId** (*string*) –

The ID of the Amazon Machine Image (AMI).

- **KeyName** (*string*) –

The name of the key pair.

- **SecurityGroups** (*list*) –

The security groups to associate with the instances.

- *(string)* –
- **ClassicLinkVPCId** *(string)* –

The ID of a ClassicLink-enabled VPC to link your EC2-Classic instances to. This parameter can only be used if you are launching EC2-Classic instances. For more information, see [ClassicLink](#) in the *Amazon Elastic Compute Cloud User Guide*.

- **ClassicLinkVPCSecurityGroups** *(list)* –

The IDs of one or more security groups for the VPC specified in `ClassicLinkVPCId`. This parameter is required if `ClassicLinkVPCId` is specified, and cannot be used otherwise. For more information, see [ClassicLink](#) in the *Amazon Elastic Compute Cloud User Guide*.

- *(string)* –
- **UserData** *(string)* –

The user data available to the instances.

- **InstanceType** *(string)* –

The instance type for the instances.

- **KernelId** *(string)* –

The ID of the kernel associated with the AMI.

- **RamdiskId** *(string)* –

The ID of the RAM disk associated with the AMI.

- **BlockDeviceMappings** *(list)* –

A block device mapping, which specifies the block devices for the instance.

- *(dict)* –

Describes a block device mapping.

- **VirtualName** *(string)* –

The name of the virtual device, `ephemeral0` to `ephemeral3`.

- **DeviceName** *(string)* –

The device name exposed to the EC2 instance (for example, `/dev/sdh` or `xvdh`).

- **Ebs** *(dict)* –

The information about the Amazon EBS volume.

- **SnapshotId** *(string)* –

The ID of the snapshot.

- **VolumeSize** *(integer)* –

The volume size, in gigabytes.

Valid values: If the volume type is `io1`, the minimum size of the volume is 10 GiB. If you specify `SnapshotId` and `VolumeSize`, `VolumeSize` must be equal to or larger than the size of the snapshot.

Default: If you create a volume from a snapshot and you don't specify a volume size, the default is the size of the snapshot.

Required: Required when the volume type is `io1`.

- **VolumeType** (*string*) –

The volume type.

Valid values: `standard` | `io1` | `gp2`

Default: `standard`

- **DeleteOnTermination** (*boolean*) –

Indicates whether to delete the volume on instance termination.

Default: `true`

- **Iops** (*integer*) –

For Provisioned IOPS (SSD) volumes only. The number of I/O operations per second (IOPS) to provision for the volume.

Valid values: Range is 100 to 4000.

Default: `None`

- **NoDevice** (*boolean*) –

Suppresses a device mapping.

If this parameter is true for the root device, the instance might fail the EC2 health check. Auto Scaling launches a replacement instance if the instance fails the health check.

- **InstanceMonitoring** (*dict*) –

Controls whether instances in this group are launched with detailed monitoring.

- **Enabled** (*boolean*) –

If `True`, instance monitoring is enabled.

- **SpotPrice** (*string*) –

The price to bid when launching Spot Instances.

- **IamInstanceProfile** (*string*) –

The name or Amazon Resource Name (ARN) of the instance profile associated with the IAM role for the instance.

- **CreatedTime** (*datetime*) –

The creation date and time for the launch configuration.

- **EbsOptimized** (*boolean*) –

Controls whether the instance is optimized for EBS I/O (`true`) or not (`false`).

- **AssociatePublicIpAddress** (*boolean*) –

Specifies whether the instances are associated with a public IP address (`true`) or not (`false`).

- **PlacementTenancy** (*string*) –

The tenancy of the instance, either `default` or `dedicated`. An instance with `dedicated` tenancy runs in an isolated, single-tenant hardware and can only be launched into a VPC.

- **NextToken** (*string*) –

The token to use when requesting the next set of items. If there are no additional items to return, the string is empty.

describe_lifecycle_hook_types()

Describes the available types of lifecycle hooks.

Request Syntax

```
response = client.describe_lifecycle_hook_types()
```

Return type dict

Returns**Response Syntax**

```
{
    'LifecycleHookTypes': [
        'string',
    ]
}
```

Response Structure

- (dict) –
 - **LifecycleHookTypes** (list) –

One or more of the following notification types:

- * autoscaling:EC2_INSTANCE_LAUNCHING
- * autoscaling:EC2_INSTANCE_TERMINATING
- * (string) –

describe_lifecycle_hooks(kwargs)**

Describes the lifecycle hooks for the specified Auto Scaling group.

Request Syntax

```
response = client.describe_lifecycle_hooks(
    AutoScalingGroupName='string',
    LifecycleHookNames=[
        'string',
    ]
)
```

Parameters

- **AutoScalingGroupName** (string) – [REQUIRED]
The name of the group.
- **LifecycleHookNames** (list) – The names of one or more lifecycle hooks.
– (string) –

Return type dict

Returns**Response Syntax**

```
{
    'LifecycleHooks': [
        {
            'LifecycleHookName': 'string',
            'AutoScalingGroupName': 'string',
            'LifecycleTransition': 'string',
            'NotificationTargetARN': 'string',
            'RoleARN': 'string',
        }
    ]
}
```

```
        'NotificationMetadata': 'string',
        'HeartbeatTimeout': 123,
        'GlobalTimeout': 123,
        'DefaultResult': 'string'
    },
]
}
```

Response Structure

- (*dict*) –
 - **LifecycleHooks** (*list*) –

The lifecycle hooks for the specified group.

* (*dict*) –

Describes a lifecycle hook, which tells Auto Scaling that you want to perform an action when an instance launches or terminates. When you have a lifecycle hook in place, the Auto Scaling group will either:

- Pause the instance after it launches, but before it is put into service
- Pause the instance as it terminates, but before it is fully terminated

For more information, see [Auto Scaling Pending State](#) and [Auto Scaling Terminating State](#) in the *Auto Scaling Developer Guide*.

- **LifecycleHookName** (*string*) –

The name of the lifecycle hook.

- **AutoScalingGroupName** (*string*) –

The name of the Auto Scaling group for the lifecycle hook.

- **LifecycleTransition** (*string*) –

The state of the EC2 instance to which you want to attach the lifecycle hook. For a list of lifecycle hook types, see [DescribeLifecycleHookTypes](#).

- **NotificationTargetARN** (*string*) –

The ARN of the notification target that Auto Scaling uses to notify you when an instance is in the transition state for the lifecycle hook. This ARN target can be either an SQS queue or an SNS topic. The notification message sent to the target includes the following:

- Lifecycle action token
- User account ID
- Name of the Auto Scaling group
- Lifecycle hook name
- EC2 instance ID
- Lifecycle transition
- Notification metadata
- **RoleARN** (*string*) –

The ARN of the IAM role that allows the Auto Scaling group to publish to the specified notification target.

- **NotificationMetadata** (*string*) –

Additional information that you want to include any time Auto Scaling sends a message to the notification target.

- **HeartbeatTimeout** (*integer*) –

The amount of time that can elapse before the lifecycle hook times out. When the lifecycle hook times out, Auto Scaling performs the action defined in the `DefaultResult` parameter. You can prevent the lifecycle hook from timing out by calling `RecordLifecycleActionHeartbeat`.

- **GlobalTimeout** (*integer*) –

The maximum length of time an instance can remain in a `Pending:Wait` or `Terminating:Wait` state. Currently, the maximum is set to 48 hours.

- **DefaultResult** (*string*) –

Defines the action the Auto Scaling group should take when the lifecycle hook timeout elapses or if an unexpected failure occurs. The valid values are `CONTINUE` and `ABANDON`. The default value is `CONTINUE`.

describe_load_balancers (***kwargs*)

Describes the load balancers for the specified Auto Scaling group.

Request Syntax

```
response = client.describe_load_balancers(
    AutoScalingGroupName='string',
    NextToken='string',
    MaxRecords=123
)
```

Parameters

- **AutoScalingGroupName** (*string*) – [REQUIRED]

The name of the group.

- **NextToken** (*string*) – The token for the next set of items to return. (You received this token from a previous call.)

- **MaxRecords** (*integer*) – The maximum number of items to return with this call.

Return type dict

Returns

Response Syntax

```
{
  'LoadBalancers': [
    {
      'LoadBalancerName': 'string',
      'State': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

- **LoadBalancers** (*list*) –

The load balancers.

- * (*dict*) –

Describes the state of a load balancer.

- **LoadBalancerName** (*string*) –
The name of the load balancer.
- **State** (*string*) –
The state of the load balancer.
- Adding - The instances in the group are being registered with the load balancer.
- Added - All instances in the group are registered with the load balancer.
- InService - At least one instance in the group passed an ELB health check.
- Removing - The instances are being deregistered from the load balancer. If connection draining is enabled, Elastic Load Balancing waits for in-flight requests to complete before deregistering the instances.
- **NextToken** (*string*) –
The token to use when requesting the next set of items. If there are no additional items to return, the string is empty.

describe_metric_collection_types()

Describes the available CloudWatch metrics for Auto Scaling.

Note that the `GroupStandbyInstances` metric is not returned by default. You must explicitly request this metric when calling `EnableMetricsCollection`.

Request Syntax

```
response = client.describe_metric_collection_types()
```

Return type dict

Returns

Response Syntax

```
{
    'Metrics': [
        {
            'Metric': 'string'
        },
    ],
    'Granularities': [
        {
            'Granularity': 'string'
        },
    ],
}
```

Response Structure

- (*dict*) –
 - **Metrics** (*list*) –
One or more metrics.
 - * (*dict*) –
Describes a metric.
 - **Metric** (*string*) –
The metric.

- GroupMinSize
- GroupMaxSize
- GroupDesiredCapacity
- GroupInServiceInstances
- GroupPendingInstances
- GroupStandbyInstances
- GroupTerminatingInstances
- GroupTotalInstances

– **Granularities** (*list*) –

The granularities for the metrics.

* (*dict*) –

Describes a granularity of a metric.

- **Granularity** (*string*) –

The granularity. The only valid value is 1Minute .

describe_notification_configurations (***kwargs*)

Describes the notification actions associated with the specified Auto Scaling group.

Request Syntax

```
response = client.describe_notification_configurations(
    AutoScalingGroupNames=[
        'string',
    ],
    NextToken='string',
    MaxRecords=123
)
```

Parameters

- **AutoScalingGroupNames** (*list*) – The name of the group.
 - (*string*) –
- **NextToken** (*string*) – The token for the next set of items to return. (You received this token from a previous call.)
- **MaxRecords** (*integer*) – The maximum number of items to return with this call.

Return type dict

Returns

Response Syntax

```
{
    'NotificationConfigurations': [
        {
            'AutoScalingGroupName': 'string',
            'TopicARN': 'string',
            'NotificationType': 'string'
        },
    ],
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **NotificationConfigurations** (*list*) –

The notification configurations.

* (*dict*) –

Describes a notification.

- **AutoScalingGroupName** (*string*) –

The name of the group.

- **TopicARN** (*string*) –

The Amazon Resource Name (ARN) of the Amazon Simple Notification Service (SNS) topic.

- **NotificationType** (*string*) –

The types of events for an action to start.

- `autoscaling:EC2_INSTANCE_LAUNCH`
- `autoscaling:EC2_INSTANCE_LAUNCH_ERROR`
- `autoscaling:EC2_INSTANCE_TERMINATE`
- `autoscaling:EC2_INSTANCE_TERMINATE_ERROR`
- `autoscaling:TEST_NOTIFICATION`

- **NextToken** (*string*) –

The token to use when requesting the next set of items. If there are no additional items to return, the string is empty.

describe_policies (***kwargs*)

Describes the policies for the specified Auto Scaling group.

Request Syntax

```
response = client.describe_policies(
    AutoScalingGroupName='string',
    PolicyNames=[
        'string',
    ],
    PolicyTypes=[
        'string',
    ],
    NextToken='string',
    MaxRecords=123
)
```

Parameters

- **AutoScalingGroupName** (*string*) – The name of the group.
- **PolicyNames** (*list*) – One or more policy names or policy ARNs to be described. If you omit this list, all policy names are described. If an group name is provided, the results are limited to that group. This list is limited to 50 items. If you specify an unknown policy name, it is ignored with no error.
 - (*string*) –
- **PolicyTypes** (*list*) – One or more policy types. Valid values are `SimpleScaling` and `StepScaling`.
 - (*string*) –
- **NextToken** (*string*) – The token for the next set of items to return. (You received this token from a previous call.)
- **MaxRecords** (*integer*) – The maximum number of items to be returned with each call.

Return type dict

Returns

Response Syntax

```

{
  'ScalingPolicies': [
    {
      'AutoScalingGroupName': 'string',
      'PolicyName': 'string',
      'PolicyARN': 'string',
      'PolicyType': 'string',
      'AdjustmentType': 'string',
      'MinAdjustmentStep': 123,
      'MinAdjustmentMagnitude': 123,
      'ScalingAdjustment': 123,
      'Cooldown': 123,
      'StepAdjustments': [
        {
          'MetricIntervalLowerBound': 123.0,
          'MetricIntervalUpperBound': 123.0,
          'ScalingAdjustment': 123
        },
      ],
      'MetricAggregationType': 'string',
      'EstimatedInstanceWarmup': 123,
      'Alarms': [
        {
          'AlarmName': 'string',
          'AlarmARN': 'string'
        },
      ],
    },
  ],
  'NextToken': 'string'
}

```

Response Structure

- *(dict)* –
 - **ScalingPolicies** (*list*) –

The scaling policies.

 - * *(dict)* –

Describes a scaling policy.

 - **AutoScalingGroupName** (*string*) –

The name of the Auto Scaling group associated with this scaling policy.
 - **PolicyName** (*string*) –

The name of the scaling policy.
 - **PolicyARN** (*string*) –

The Amazon Resource Name (ARN) of the policy.
 - **PolicyType** (*string*) –

The policy type. Valid values are SimpleScaling and StepScaling.
 - **AdjustmentType** (*string*) –

The adjustment type, which specifies how ScalingAdjustment is interpreted. Valid values are ChangeInCapacity , ExactCapacity , and

`PercentChangeInCapacity`.

- **MinAdjustmentStep** (*integer*) –

Available for backward compatibility. Use `MinAdjustmentMagnitude` instead.

- **MinAdjustmentMagnitude** (*integer*) –

The minimum number of instances to scale. If the value of `AdjustmentType` is `PercentChangeInCapacity`, the scaling policy changes the `DesiredCapacity` of the Auto Scaling group by at least this many instances. Otherwise, the error is `ValidationError`.

- **ScalingAdjustment** (*integer*) –

The amount by which to scale, based on the specified adjustment type. A positive value adds to the current capacity while a negative number removes from the current capacity.

- **Cooldown** (*integer*) –

The amount of time, in seconds, after a scaling activity completes before any further trigger-related scaling activities can start.

- **StepAdjustments** (*list*) –

A set of adjustments that enable you to scale based on the size of the alarm breach.

- (*dict*) –

Describes an adjustment based on the difference between the value of the aggregated CloudWatch metric and the breach threshold that you’ve defined for the alarm.

For the following examples, suppose that you have an alarm with a breach threshold of 50:

- If you want the adjustment to be triggered when the metric is greater than or equal to 50 and less than 60, specify a lower bound of 0 and an upper bound of 10.
- If you want the adjustment to be triggered when the metric is greater than 40 and less than or equal to 50, specify a lower bound of -10 and an upper bound of 0.

There are a few rules for the step adjustments for your step policy:

- The ranges of your step adjustments can’t overlap or have a gap.
- At most one step adjustment can have a null lower bound. If one step adjustment has a negative lower bound, then there must be a step adjustment with a null lower bound.
- At most one step adjustment can have a null upper bound. If one step adjustment has a positive upper bound, then there must be a step adjustment with a null upper bound.
- The upper and lower bound can’t be null in the same step adjustment.

- **MetricIntervalLowerBound** (*float*) –

The lower bound for the difference between the alarm threshold and the CloudWatch metric. If the metric value is above the breach threshold, the lower bound is inclusive (the metric must be greater than or equal to the threshold plus the lower

bound). Otherwise, it is exclusive (the metric must be greater than the threshold plus the lower bound). A null value indicates negative infinity.

- **MetricIntervalUpperBound** (*float*) –

The upper bound for the difference between the alarm threshold and the CloudWatch metric. If the metric value is above the breach threshold, the upper bound is exclusive (the metric must be less than the threshold plus the upper bound). Otherwise, it is inclusive (the metric must be less than or equal to the threshold plus the upper bound). A null value indicates positive infinity.

The upper bound must be greater than the lower bound.

- **ScalingAdjustment** (*integer*) –

The amount by which to scale, based on the specified adjustment type. A positive value adds to the current capacity while a negative number removes from the current capacity.

- **MetricAggregationType** (*string*) –

The aggregation type for the CloudWatch metrics. Valid values are `Minimum`, `Maximum`, and `Average`.

- **EstimatedInstanceWarmup** (*integer*) –

The estimated time, in seconds, until a newly launched instance can contribute to the CloudWatch metrics.

- **Alarms** (*list*) –

The CloudWatch alarms related to the policy.

- (*dict*) –

Describes an alarm.

- **AlarmName** (*string*) –

The name of the alarm.

- **AlarmARN** (*string*) –

The Amazon Resource Name (ARN) of the alarm.

- **NextToken** (*string*) –

The token to use when requesting the next set of items. If there are no additional items to return, the string is empty.

describe_scaling_activities (**kwargs)

Describes one or more scaling activities for the specified Auto Scaling group. If you omit the `ActivityIds`, the call returns all activities from the past six weeks. Activities are sorted by the start time. Activities still in progress appear first on the list.

Request Syntax

```
response = client.describe_scaling_activities(
    ActivityIds=[
        'string',
    ],
    AutoScalingGroupName='string',
    MaxRecords=123,
    NextToken='string'
)
```

Parameters

- **ActivityIds** (*list*) – The activity IDs of the desired scaling activities. If this list is omitted, all activities are described. If the `AutoScalingGroupName` parameter is provided, the results are limited to that group. The list of requested activities cannot contain more than 50 items. If unknown activities are requested, they are ignored with no error.
 - (*string*) –
- **AutoScalingGroupName** (*string*) – The name of the group.
- **MaxRecords** (*integer*) – The maximum number of items to return with this call.
- **NextToken** (*string*) – The token for the next set of items to return. (You received this token from a previous call.)

Return type dict

Returns

Response Syntax

```
{
  'Activities': [
    {
      'ActivityId': 'string',
      'AutoScalingGroupName': 'string',
      'Description': 'string',
      'Cause': 'string',
      'StartTime': datetime(2015, 1, 1),
      'EndTime': datetime(2015, 1, 1),
      'StatusCode': 'WaitingForSpotInstanceRequestId'|'WaitingForSpotInstance...',
      'StatusMessage': 'string',
      'Progress': 123,
      'Details': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **Activities** (*list*) –
The scaling activities.
 - * (*dict*) –

Describes scaling activity, which is a long-running process that represents a change to your Auto Scaling group, such as changing its size or replacing an instance.

- **ActivityId** (*string*) –
The ID of the activity.
- **AutoScalingGroupName** (*string*) –
The name of the Auto Scaling group.
- **Description** (*string*) –
A friendly, more verbose description of the activity.
- **Cause** (*string*) –
The reason the activity began.
- **StartTime** (*datetime*) –
The start time of the activity.
- **EndTime** (*datetime*) –

- The end time of the activity.
 - **StatusCode** (*string*) –
 - The current status of the activity.
 - **StatusMessage** (*string*) –
 - A friendly, more verbose description of the activity status.
 - **Progress** (*integer*) –
 - A value between 0 and 100 that indicates the progress of the activity.
 - **Details** (*string*) –
 - The details about the activity.
- **NextToken** (*string*) –
- The token to use when requesting the next set of items. If there are no additional items to return, the string is empty.

describe_scaling_process_types()

Describes the scaling process types for use with ResumeProcesses and SuspendProcesses .

Request Syntax

```
response = client.describe_scaling_process_types()
```

Return type dict

Returns

Response Syntax

```
{
  'Processes': [
    {
      'ProcessName': 'string'
    },
  ]
}
```

Response Structure

- (*dict*) –
- **Processes** (*list*) –

The names of the process types.

- * (*dict*) –

Describes a process type.

For more information, see [Auto Scaling Processes](#) in the *Auto Scaling Developer Guide* .

- **ProcessName** (*string*) –

The name of the process.

- Launch
- Terminate
- AddToLoadBalancer
- AlarmNotification
- AZRebalance
- HealthCheck
- ReplaceUnhealthy
- ScheduledActions

describe_scheduled_actions (***kwargs*)

Describes the actions scheduled for your Auto Scaling group that haven't run. To describe the actions that have already run, use `DescribeScalingActivities`.

Request Syntax

```
response = client.describe_scheduled_actions(
    AutoScalingGroupName='string',
    ScheduledActionNames=[
        'string',
    ],
    StartTime=datetime(2015, 1, 1),
    EndTime=datetime(2015, 1, 1),
    NextToken='string',
    MaxRecords=123
)
```

Parameters

- **AutoScalingGroupName** (*string*) – The name of the group.
- **ScheduledActionNames** (*list*) – Describes one or more scheduled actions. If you omit this list, the call describes all scheduled actions. If you specify an unknown scheduled action it is ignored with no error.

You can describe up to a maximum of 50 instances with a single call. If there are more items to return, the call returns a token. To get the next set of items, repeat the call with the returned token in the `NextToken` parameter.

- (*string*) –
- **StartTime** (*datetime*) – The earliest scheduled start time to return. If scheduled action names are provided, this parameter is ignored.
- **EndTime** (*datetime*) – The latest scheduled start time to return. If scheduled action names are provided, this parameter is ignored.
- **NextToken** (*string*) – The token for the next set of items to return. (You received this token from a previous call.)
- **MaxRecords** (*integer*) – The maximum number of items to return with this call.

Return type dict

Returns**Response Syntax**

```
{
    'ScheduledUpdateGroupActions': [
        {
            'AutoScalingGroupName': 'string',
            'ScheduledActionName': 'string',
            'ScheduledActionARN': 'string',
            'Time': datetime(2015, 1, 1),
            'StartTime': datetime(2015, 1, 1),
            'EndTime': datetime(2015, 1, 1),
            'Recurrence': 'string',
            'MinSize': 123,
            'MaxSize': 123,
            'DesiredCapacity': 123
        },
    ],
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **ScheduledUpdateGroupActions** (*list*) –

The scheduled actions.

 - * (*dict*) –

Describes a scheduled update to an Auto Scaling group.

 - **AutoScalingGroupName** (*string*) –

The name of the group.
 - **ScheduledActionName** (*string*) –

The name of the scheduled action.
 - **ScheduledActionARN** (*string*) –

The Amazon Resource Name (ARN) of the scheduled action.
 - **Time** (*datetime*) –

This parameter is deprecated; use `StartTime` instead.
 - **StartTime** (*datetime*) –

The date and time that the action is scheduled to begin. This date and time can be up to one month in the future.

When `StartTime` and `EndTime` are specified with `Recurrence`, they form the boundaries of when the recurring action will start and stop.
 - **EndTime** (*datetime*) –

The date and time that the action is scheduled to end. This date and time can be up to one month in the future.
 - **Recurrence** (*string*) –

The recurring schedule for the action.
 - **MinSize** (*integer*) –

The minimum size of the group.
 - **MaxSize** (*integer*) –

The maximum size of the group.
 - **DesiredCapacity** (*integer*) –

The number of instances you prefer to maintain in the group.
 - **NextToken** (*string*) –

The token to use when requesting the next set of items. If there are no additional items to return, the string is empty.

describe_tags (***kwargs*)

Describes the specified tags.

You can use filters to limit the results. For example, you can query for the tags for a specific Auto Scaling group. You can specify multiple values for a filter. A tag must match at least one of the specified values for it to be included in the results.

You can also specify multiple filters. The result includes information for a particular tag only if it matches all the filters. If there's no match, no special message is returned.

Request Syntax

```
response = client.describe_tags(
    Filters=[
        {
```

```
        'Name': 'string',
        'Values': [
            'string',
        ]
    },
],
NextToken='string',
MaxRecords=123
)
```

Parameters

- **Filters** (*list*) – A filter used to scope the tags to return.
 - (*dict*) –

Describes a filter.

- * **Name** (*string*) –

The name of the filter. The valid values are: "auto-scaling-group" , "key" , "value" , and "propagate-at-launch".

- * **Values** (*list*) –

The value of the filter.

- (*string*) –

- **NextToken** (*string*) – The token for the next set of items to return. (You received this token from a previous call.)
- **MaxRecords** (*integer*) – The maximum number of items to return with this call.

Return type dict

Returns

Response Syntax

```
{
  'Tags': [
    {
      'ResourceId': 'string',
      'ResourceType': 'string',
      'Key': 'string',
      'Value': 'string',
      'PropagateAtLaunch': True|False
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **Tags** (*list*) –

The tags.

- * (*dict*) –

Describes a tag for an Auto Scaling group.

- **ResourceId** (*string*) –

The name of the group.

- **ResourceType** (*string*) –

The type of resource. The only supported value is auto-scaling-group.

- **Key** (*string*) –
The tag key.
- **Value** (*string*) –
The tag value.
- **PropagateAtLaunch** (*boolean*) –
Determines whether the tag is added to new instances as they are launched in the group.
- **NextToken** (*string*) –
The token to use when requesting the next set of items. If there are no additional items to return, the string is empty.

describe_termination_policy_types()

Describes the termination policies supported by Auto Scaling.

Request Syntax

```
response = client.describe_termination_policy_types()
```

Return type dict

Returns

Response Syntax

```
{
    'TerminationPolicyTypes': [
        'string',
    ]
}
```

Response Structure

- (*dict*) –
 - **TerminationPolicyTypes** (*list*) –
The termination policies supported by Auto Scaling (OldestInstance , OldestLaunchConfiguration , NewestInstance , ClosestToNextInstanceHour , and Default).
* (*string*) –

detach_instances(kwargs)**

Removes one or more instances from the specified Auto Scaling group. After the instances are detached, you can manage them independently from the rest of the Auto Scaling group.

For more information, see [Detach EC2 Instances from Your Auto Scaling Group](#) in the *Auto Scaling Developer Guide* .

Request Syntax

```
response = client.detach_instances(
    InstanceIds=[
        'string',
    ],
    AutoScalingGroupName='string',
    ShouldDecrementDesiredCapacity=True|False
)
```

Parameters

- **InstanceIds** (*list*) – One or more instance IDs.

- (*string*) –
- **AutoScalingGroupName** (*string*) – [REQUIRED]
The name of the group.
- **ShouldDecrementDesiredCapacity** (*boolean*) – [REQUIRED]
If `True`, the Auto Scaling group decrements the desired capacity value by the number of instances detached.

Return type dict

Returns

Response Syntax

```
{
  'Activities': [
    {
      'ActivityId': 'string',
      'AutoScalingGroupName': 'string',
      'Description': 'string',
      'Cause': 'string',
      'StartTime': datetime(2015, 1, 1),
      'EndTime': datetime(2015, 1, 1),
      'StatusCode': 'WaitingForSpotInstanceRequestId'|'WaitingForSpotInstance...',
      'StatusMessage': 'string',
      'Progress': 123,
      'Details': 'string'
    },
  ]
}
```

Response Structure

- (*dict*) –
 - **Activities** (*list*) –
The activities related to detaching the instances from the Auto Scaling group.
 - * (*dict*) –
Describes scaling activity, which is a long-running process that represents a change to your Auto Scaling group, such as changing its size or replacing an instance.
 - **ActivityId** (*string*) –
The ID of the activity.
 - **AutoScalingGroupName** (*string*) –
The name of the Auto Scaling group.
 - **Description** (*string*) –
A friendly, more verbose description of the activity.
 - **Cause** (*string*) –
The reason the activity began.
 - **StartTime** (*datetime*) –
The start time of the activity.
 - **EndTime** (*datetime*) –
The end time of the activity.
 - **StatusCode** (*string*) –

- The current status of the activity.
- **StatusMessage** (*string*) –
A friendly, more verbose description of the activity status.
- **Progress** (*integer*) –
A value between 0 and 100 that indicates the progress of the activity.
- **Details** (*string*) –
The details about the activity.

detach_load_balancers (***kwargs*)

Removes one or more load balancers from the specified Auto Scaling group.

When you detach a load balancer, it enters the `Removing` state while deregistering the instances in the group. When all instances are deregistered, then you can no longer describe the load balancer using `DescribeLoadBalancers`. Note that the instances remain running.

Request Syntax

```
response = client.detach_load_balancers(
    AutoScalingGroupName='string',
    LoadBalancerNames=[
        'string',
    ]
)
```

Parameters

- **AutoScalingGroupName** (*string*) – The name of the group.
- **LoadBalancerNames** (*list*) – One or more load balancer names.
– (*string*) –

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –

disable_metrics_collection (***kwargs*)

Disables monitoring of the specified metrics for the specified Auto Scaling group.

Request Syntax

```
response = client.disable_metrics_collection(
    AutoScalingGroupName='string',
    Metrics=[
        'string',
    ]
)
```

Parameters

- **AutoScalingGroupName** (*string*) – **[REQUIRED]**
The name or Amazon Resource Name (ARN) of the group.
- **Metrics** (*list*) – One or more metrics. If you omit this parameter, all metrics are disabled.

- GroupMinSize
- GroupMaxSize
- GroupDesiredCapacity
- GroupInServiceInstances
- GroupPendingInstances
- GroupStandbyInstances
- GroupTerminatingInstances
- GroupTotalInstances
- (*string*) -

Returns None

enable_metrics_collection (***kwargs*)

Enables monitoring of the specified metrics for the specified Auto Scaling group.

You can only enable metrics collection if InstanceMonitoring in the launch configuration for the group is set to True .

Request Syntax

```
response = client.enable_metrics_collection(
    AutoScalingGroupName='string',
    Metrics=[
        'string',
    ],
    Granularity='string'
)
```

Parameters

- **AutoScalingGroupName** (*string*) – [REQUIRED]

The name or ARN of the Auto Scaling group.

- **Metrics** (*list*) – One or more metrics. If you omit this parameter, all metrics are enabled.

- GroupMinSize
- GroupMaxSize
- GroupDesiredCapacity
- GroupInServiceInstances
- GroupPendingInstances
- GroupStandbyInstances
- GroupTerminatingInstances
- GroupTotalInstances

Note that the GroupStandbyInstances metric is not enabled by default. You must explicitly request this metric.

- (*string*) -

- **Granularity** (*string*) – [REQUIRED]

The granularity to associate with the metrics to collect. The only valid value is 1Minute .

Returns None

enter_standby (***kwargs*)

Moves the specified instances into Standby mode.

For more information, see [Auto Scaling InService State](#) in the *Auto Scaling Developer Guide* .

Request Syntax

```

response = client.enter_standby(
    InstanceIds=[
        'string',
    ],
    AutoScalingGroupName='string',
    ShouldDecrementDesiredCapacity=True|False
)

```

Parameters

- **InstanceIds** (*list*) – One or more instances to move into Standby mode. You must specify at least one instance ID.
– (*string*) –

- **AutoScalingGroupName** (*string*) – [REQUIRED]

The name of the Auto Scaling group.

- **ShouldDecrementDesiredCapacity** (*boolean*) – [REQUIRED]

Specifies whether the instances moved to Standby mode count as part of the Auto Scaling group's desired capacity. If set, the desired capacity for the Auto Scaling group decrements by the number of instances moved to Standby mode.

Return type dict

Returns

Response Syntax

```

{
    'Activities': [
        {
            'ActivityId': 'string',
            'AutoScalingGroupName': 'string',
            'Description': 'string',
            'Cause': 'string',
            'StartTime': datetime(2015, 1, 1),
            'EndTime': datetime(2015, 1, 1),
            'StatusCode': 'WaitingForSpotInstanceRequestId'|'WaitingForSpotInstance...',
            'StatusMessage': 'string',
            'Progress': 123,
            'Details': 'string'
        },
    ]
}

```

Response Structure

- (*dict*) –
– **Activities** (*list*) –

The activities related to moving instances into Standby mode.

- * (*dict*) –

Describes scaling activity, which is a long-running process that represents a change to your Auto Scaling group, such as changing its size or replacing an instance.

- **ActivityId** (*string*) –

The ID of the activity.

- **AutoScalingGroupName** (*string*) –

The name of the Auto Scaling group.

- **Description** (*string*) –
A friendly, more verbose description of the activity.
- **Cause** (*string*) –
The reason the activity began.
- **StartTime** (*datetime*) –
The start time of the activity.
- **EndTime** (*datetime*) –
The end time of the activity.
- **StatusCode** (*string*) –
The current status of the activity.
- **StatusMessage** (*string*) –
A friendly, more verbose description of the activity status.
- **Progress** (*integer*) –
A value between 0 and 100 that indicates the progress of the activity.
- **Details** (*string*) –
The details about the activity.

execute_policy (***kwargs*)
Executes the specified policy.

Request Syntax

```
response = client.execute_policy(  
    AutoScalingGroupName='string',  
    PolicyName='string',  
    HonorCooldown=True|False,  
    MetricValue=123.0,  
    BreachThreshold=123.0  
)
```

Parameters

- **AutoScalingGroupName** (*string*) – The name or Amazon Resource Name (ARN) of the Auto Scaling group.
- **PolicyName** (*string*) – **[REQUIRED]**

The name or ARN of the policy.

- **HonorCooldown** (*boolean*) – If this parameter is true, Auto Scaling waits for the cooldown period to complete before executing the policy. Otherwise, Auto Scaling executes the policy without waiting for the cooldown period to complete.

This parameter is not supported if the policy type is `StepScaling`.

For more information, see [Understanding Auto Scaling Cooldowns](#) in the *Auto Scaling Developer Guide*.

- **MetricValue** (*float*) – The metric value to compare to `BreachThreshold`. This enables you to execute a policy of type `StepScaling` and determine which step adjustment to use. For example, if the breach threshold is 50 and you want to use a step adjustment with a lower bound of 0 and an upper bound of 10, you can set the metric value to 59.

If you specify a metric value that doesn't correspond to a step adjustment for the policy, the call returns an error.

This parameter is required if the policy type is `StepScaling` and not supported otherwise.

- **BreachThreshold** (*float*) – The breach threshold for the alarm.

This parameter is required if the policy type is `StepScaling` and not supported otherwise.

Returns None

exit_standby (***kwargs*)

Moves the specified instances out of Standby mode.

For more information, see [Auto Scaling InService State](#) in the *Auto Scaling Developer Guide*.

Request Syntax

```
response = client.exit_standby(
    InstanceIds=[
        'string',
    ],
    AutoScalingGroupName='string'
)
```

Parameters

- **InstanceIds** (*list*) – One or more instance IDs. You must specify at least one instance ID.
 - (*string*) –
- **AutoScalingGroupName** (*string*) – [REQUIRED]

The name of the Auto Scaling group.

Return type dict

Returns

Response Syntax

```
{
    'Activities': [
        {
            'ActivityId': 'string',
            'AutoScalingGroupName': 'string',
            'Description': 'string',
            'Cause': 'string',
            'StartTime': datetime(2015, 1, 1),
            'EndTime': datetime(2015, 1, 1),
            'StatusCode': 'WaitingForSpotInstanceRequestId'|'WaitingForSpotInstance',
            'StatusMessage': 'string',
            'Progress': 123,
            'Details': 'string'
        },
    ]
}
```

Response Structure

- (*dict*) –
 - **Activities** (*list*) –

The activities related to moving instances out of Standby mode.

* (*dict*) –

Describes scaling activity, which is a long-running process that represents a change to your Auto Scaling group, such as changing its size or replacing an instance.

- **ActivityId** (*string*) –
The ID of the activity.
- **AutoScalingGroupName** (*string*) –
The name of the Auto Scaling group.
- **Description** (*string*) –
A friendly, more verbose description of the activity.
- **Cause** (*string*) –
The reason the activity began.
- **StartTime** (*datetime*) –
The start time of the activity.
- **EndTime** (*datetime*) –
The end time of the activity.
- **StatusCode** (*string*) –
The current status of the activity.
- **StatusMessage** (*string*) –
A friendly, more verbose description of the activity status.
- **Progress** (*integer*) –
A value between 0 and 100 that indicates the progress of the activity.
- **Details** (*string*) –
The details about the activity.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_waiter (*waiter_name*)

put_lifecycle_hook (**kwargs)

Creates or updates a lifecycle hook for the specified Auto Scaling Group.

A lifecycle hook tells Auto Scaling that you want to perform an action on an instance that is not actively in service; for example, either when the instance launches or before the instance terminates.

This operation is a part of the basic sequence for adding a lifecycle hook to an Auto Scaling group:

- Create a notification target. A target can be either an Amazon SQS queue or an Amazon SNS topic.
- Create an IAM role. This role allows Auto Scaling to publish lifecycle notifications to the designated SQS queue or SNS topic.
- **Create the lifecycle hook. You can create a hook that acts when instances launch or when instances terminate.**
- If necessary, record the lifecycle action heartbeat to keep the instance in a pending state.
- Complete the lifecycle action.

For more information, see [Auto Scaling Pending State](#) and [Auto Scaling Terminating State](#) in the *Auto Scaling Developer Guide*.

If you exceed your maximum limit of lifecycle hooks, which by default is 50 per region, the call fails. For information about updating this limit, see [AWS Service Limits](#) in the *Amazon Web Services General Reference*.

Request Syntax

```
response = client.put_lifecycle_hook(
    LifecycleHookName='string',
    AutoScalingGroupName='string',
    LifecycleTransition='string',
    RoleARN='string',
    NotificationTargetARN='string',
    NotificationMetadata='string',
    HeartbeatTimeout=123,
    DefaultResult='string'
)
```

Parameters

- **LifecycleHookName** (*string*) – [REQUIRED]

The name of the lifecycle hook.

- **AutoScalingGroupName** (*string*) – [REQUIRED]

The name of the Auto Scaling group to which you want to assign the lifecycle hook.

- **LifecycleTransition** (*string*) – The instance state to which you want to attach the lifecycle hook. For a list of lifecycle hook types, see [DescribeLifecycleHookTypes](#).

This parameter is required for new lifecycle hooks, but optional when updating existing hooks.

- **RoleARN** (*string*) – The ARN of the IAM role that allows the Auto Scaling group to publish to the specified notification target.

This parameter is required for new lifecycle hooks, but optional when updating existing hooks.

- **NotificationTargetARN** (*string*) – The ARN of the notification target that Auto Scaling will use to notify you when an instance is in the transition state for the lifecycle hook. This ARN target can be either an SQS queue or an SNS topic.

This parameter is required for new lifecycle hooks, but optional when updating existing hooks.

The notification message sent to the target will include:

- **LifecycleActionToken** . The Lifecycle action token.
- **AccountId** . The user account ID.
- **AutoScalingGroupName** . The name of the Auto Scaling group.
- **LifecycleHookName** . The lifecycle hook name.
- **EC2InstanceId** . The EC2 instance ID.
- **LifecycleTransition** . The lifecycle transition.
- **NotificationMetadata** . The notification metadata.

This operation uses the JSON format when sending notifications to an Amazon SQS queue, and an email key/value pair format when sending notifications to an Amazon SNS topic.

When you call this operation, a test message is sent to the notification target. This test message contains an additional key/value pair: `Event:autoscaling:TEST_NOTIFICATION`.

- **NotificationMetadata** (*string*) – Contains additional information that you want to include any time Auto Scaling sends a message to the notification target.
- **HeartbeatTimeout** (*integer*) – Defines the amount of time, in seconds, that can elapse before the lifecycle hook times out. When the lifecycle hook times out, Auto Scaling performs the action defined in the `DefaultResult` parameter. You can prevent the lifecycle hook from timing out by calling `RecordLifecycleActionHeartbeat` . The default value for this parameter is 3600 seconds (1 hour).
- **DefaultResult** (*string*) – Defines the action the Auto Scaling group should take when the lifecycle hook timeout elapses or if an unexpected failure occurs. The value for this parameter can be either `CONTINUE` or `ABANDON` . The default value for this parameter is `ABANDON` .

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –

put_notification_configuration (***kwargs*)

Configures an Auto Scaling group to send notifications when specified events take place. Subscribers to this topic can have messages for events delivered to an endpoint such as a web server or email address.

For more information see [Getting Notifications When Your Auto Scaling Group Changes](#) in the *Auto Scaling Developer Guide* .

This configuration overwrites an existing configuration.

Request Syntax

```
response = client.put_notification_configuration(
    AutoScalingGroupName='string',
    TopicARN='string',
    NotificationTypes=[
        'string',
    ]
)
```

Parameters

- **AutoScalingGroupName** (*string*) – [REQUIRED]

The name of the Auto Scaling group.

- **TopicARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the Amazon Simple Notification Service (SNS) topic.

- **NotificationTypes** (*list*) – [REQUIRED]

The type of event that will cause the notification to be sent. For details about notification types supported by Auto Scaling, see `DescribeAutoScalingNotificationTypes`.

– (*string*) –

Returns None

put_scaling_policy (***kwargs*)

Creates or updates a policy for an Auto Scaling group. To update an existing policy, use the existing policy name and set the parameters you want to change. Any existing parameter not changed in an update to an existing policy is not changed in this update request.

If you exceed your maximum limit of step adjustments, which by default is 20 per region, the call fails. For information about updating this limit, see [AWS Service Limits](#) in the *Amazon Web Services General Reference*.

Request Syntax

```
response = client.put_scaling_policy(
    AutoScalingGroupName='string',
    PolicyName='string',
    PolicyType='string',
    AdjustmentType='string',
    MinAdjustmentStep=123,
    MinAdjustmentMagnitude=123,
    ScalingAdjustment=123,
    Cooldown=123,
    MetricAggregationType='string',
    StepAdjustments=[
        {
            'MetricIntervalLowerBound': 123.0,
            'MetricIntervalUpperBound': 123.0,
            'ScalingAdjustment': 123
        },
    ],
    EstimatedInstanceWarmup=123
)
```

Parameters

- **AutoScalingGroupName** (*string*) – [REQUIRED]

The name or ARN of the group.

- **PolicyName** (*string*) – [REQUIRED]

The name of the policy.

- **PolicyType** (*string*) – The policy type. Valid values are `SimpleScaling` and `StepScaling`. If the policy type is null, the value is treated as `SimpleScaling`.

- **AdjustmentType** (*string*) – [REQUIRED]

The adjustment type. Valid values are `ChangeInCapacity`, `ExactCapacity`, and `PercentChangeInCapacity`.

For more information, see [Dynamic Scaling](#) in the *Auto Scaling Developer Guide*.

- **MinAdjustmentStep** (*integer*) – Available for backward compatibility. Use `MinAdjustmentMagnitude` instead.
- **MinAdjustmentMagnitude** (*integer*) – The minimum number of instances to scale. If the value of `AdjustmentType` is `PercentChangeInCapacity`, the scaling policy changes the `DesiredCapacity` of the Auto Scaling group by at least this many instances. Otherwise, the error is `ValidationError`.
- **ScalingAdjustment** (*integer*) – The amount by which to scale, based on the specified adjustment type. A positive value adds to the current capacity while a negative number removes from the current capacity.

This parameter is required if the policy type is `SimpleScaling` and not supported otherwise.

- **Cooldown** (*integer*) – The amount of time, in seconds, after a scaling activity completes and before the next scaling activity can start. If this parameter is not specified, the default cooldown period for the group applies.

This parameter is not supported unless the policy type is `SimpleScaling`.

For more information, see [Understanding Auto Scaling Cooldowns](#) in the *Auto Scaling Developer Guide*.

- **MetricAggregationType** (*string*) – The aggregation type for the CloudWatch metrics. Valid values are `Minimum`, `Maximum`, and `Average`. If the aggregation type is null, the value is treated as `Average`.

This parameter is not supported if the policy type is `SimpleScaling`.

- **StepAdjustments** (*list*) – A set of adjustments that enable you to scale based on the size of the alarm breach.

This parameter is required if the policy type is `StepScaling` and not supported otherwise.

– (*dict*) –

Describes an adjustment based on the difference between the value of the aggregated CloudWatch metric and the breach threshold that you’ve defined for the alarm.

For the following examples, suppose that you have an alarm with a breach threshold of 50:

- * If you want the adjustment to be triggered when the metric is greater than or equal to 50 and less than 60, specify a lower bound of 0 and an upper bound of 10.
- * If you want the adjustment to be triggered when the metric is greater than 40 and less than or equal to 50, specify a lower bound of -10 and an upper bound of 0.

There are a few rules for the step adjustments for your step policy:

- * The ranges of your step adjustments can’t overlap or have a gap.
- * At most one step adjustment can have a null lower bound. If one step adjustment has a negative lower bound, then there must be a step adjustment with a null lower bound.
- * At most one step adjustment can have a null upper bound. If one step adjustment has a positive upper bound, then there must be a step adjustment with a null upper bound.
- * The upper and lower bound can’t be null in the same step adjustment.

- * **MetricIntervalLowerBound** (*float*) –

The lower bound for the difference between the alarm threshold and the CloudWatch metric. If the metric value is above the breach threshold, the lower bound is inclusive (the metric must be greater than or equal to the threshold plus the lower bound). Otherwise, it is exclusive (the metric must be greater than the threshold plus the lower bound). A null value indicates negative infinity.

* **MetricIntervalUpperBound** (*float*) –

The upper bound for the difference between the alarm threshold and the CloudWatch metric. If the metric value is above the breach threshold, the upper bound is exclusive (the metric must be less than the threshold plus the upper bound). Otherwise, it is inclusive (the metric must be less than or equal to the threshold plus the upper bound). A null value indicates positive infinity.

The upper bound must be greater than the lower bound.

* **ScalingAdjustment** (*integer*) – **[REQUIRED]**

The amount by which to scale, based on the specified adjustment type. A positive value adds to the current capacity while a negative number removes from the current capacity.

- **EstimatedInstanceWarmup** (*integer*) – The estimated time, in seconds, until a newly launched instance can contribute to the CloudWatch metrics. The default is to use the value specified for the default cooldown period for the group.

This parameter is not supported if the policy type is `SimpleScaling`.

Return type dict

Returns

Response Syntax

```
{
    'PolicyARN': 'string'
}
```

Response Structure

- (*dict*) –
 - **PolicyARN** (*string*) –

The Amazon Resource Name (ARN) of the policy.

put_scheduled_update_group_action (***kwargs*)

Creates or updates a scheduled scaling action for an Auto Scaling group. When updating a scheduled scaling action, if you leave a parameter unspecified, the corresponding value remains unchanged in the affected Auto Scaling group.

For more information, see [Scheduled Scaling](#) in the *Auto Scaling Developer Guide*.

Request Syntax

```
response = client.put_scheduled_update_group_action(
    AutoScalingGroupName='string',
    ScheduledActionName='string',
    Time=datetime(2015, 1, 1),
    StartTime=datetime(2015, 1, 1),
    EndTime=datetime(2015, 1, 1),
    Recurrence='string',
    MinSize=123,
    MaxSize=123,
```

```
DesiredCapacity=123
)
```

Parameters

- **AutoScalingGroupName** (*string*) – [REQUIRED]

The name or Amazon Resource Name (ARN) of the Auto Scaling group.

- **ScheduledActionName** (*string*) – [REQUIRED]

The name of this scaling action.

- **Time** (*datetime*) – This parameter is deprecated; use `StartTime` instead.

The time for this action to start. If both `Time` and `StartTime` are specified, their values must be identical.

- **StartTime** (*datetime*) – The time for this action to start, in “YYYY-MM-DDThh:mm:ssZ” format in UTC/GMT only (for example, 2014-06-01T00:00:00Z).

If you try to schedule your action in the past, Auto Scaling returns an error message.

When `StartTime` and `EndTime` are specified with `Recurrence`, they form the boundaries of when the recurring action starts and stops.

- **EndTime** (*datetime*) – The time for this action to end.
- **Recurrence** (*string*) – The time when recurring future actions will start. Start time is specified by the user following the Unix cron syntax format. For more information, see [Cron](#) in Wikipedia.

When `StartTime` and `EndTime` are specified with `Recurrence`, they form the boundaries of when the recurring action will start and stop.

- **MinSize** (*integer*) – The minimum size for the Auto Scaling group.
- **MaxSize** (*integer*) – The maximum size for the Auto Scaling group.
- **DesiredCapacity** (*integer*) – The number of EC2 instances that should be running in the group.

Returns None

`record_lifecycle_action_heartbeat` (**kwargs)

Records a heartbeat for the lifecycle action associated with a specific token. This extends the timeout by the length of time defined by the `HeartbeatTimeout` parameter of `PutLifecycleHook`.

This operation is a part of the basic sequence for adding a lifecycle hook to an Auto Scaling group:

- Create a notification target. A target can be either an Amazon SQS queue or an Amazon SNS topic.
- Create an IAM role. This role allows Auto Scaling to publish lifecycle notifications to the designated SQS queue or SNS topic.
- Create the lifecycle hook. You can create a hook that acts when instances launch or when instances terminate.
- **If necessary, record the lifecycle action heartbeat to keep the instance in a pending state.**
- Complete the lifecycle action.

For more information, see [Auto Scaling Pending State](#) and [Auto Scaling Terminating State](#) in the *Auto Scaling Developer Guide*.

Request Syntax

```
response = client.record_lifecycle_action_heartbeat(
    LifecycleHookName='string',
    AutoScalingGroupName='string',
    LifecycleActionToken='string'
)
```

Parameters

- **LifecycleHookName** (*string*) – [REQUIRED]

The name of the lifecycle hook.

- **AutoScalingGroupName** (*string*) – [REQUIRED]

The name of the Auto Scaling group for the hook.

- **LifecycleActionToken** (*string*) – [REQUIRED]

A token that uniquely identifies a specific lifecycle action associated with an instance. Auto Scaling sends this token to the notification target you specified when you created the lifecycle hook.

Return type dict

Returns**Response Syntax**

```
{ }
```

Response Structure

- (*dict*) –

resume_processes (***kwargs*)

Resumes the specified suspended Auto Scaling processes for the specified Auto Scaling group. To resume specific processes, use the `ScalingProcesses` parameter. To resume all processes, omit the `ScalingProcesses` parameter. For more information, see [Suspend and Resume Auto Scaling Processes](#) in the *Auto Scaling Developer Guide*.

Request Syntax

```
response = client.resume_processes(
    AutoScalingGroupName='string',
    ScalingProcesses=[
        'string',
    ]
)
```

Parameters

- **AutoScalingGroupName** (*string*) – [REQUIRED]

The name or Amazon Resource Name (ARN) of the Auto Scaling group.

- **ScalingProcesses** (*list*) – One or more of the following processes:

- Launch
- Terminate
- HealthCheck
- ReplaceUnhealthy
- AZRebalance
- AlarmNotification
- ScheduledActions
- AddToLoadBalancer

- (*string*) –

Returns None

set_desired_capacity (***kwargs*)

Sets the size of the specified Auto Scaling group.

For more information about desired capacity, see [What Is Auto Scaling?](#) in the *Auto Scaling Developer Guide*.

Request Syntax

```
response = client.set_desired_capacity(
    AutoScalingGroupName='string',
    DesiredCapacity=123,
    HonorCooldown=True|False
)
```

Parameters

- **AutoScalingGroupName** (*string*) – [REQUIRED]
The name of the Auto Scaling group.
- **DesiredCapacity** (*integer*) – [REQUIRED]
The number of EC2 instances that should be running in the Auto Scaling group.
- **HonorCooldown** (*boolean*) – By default, `SetDesiredCapacity` overrides any cooldown period associated with the Auto Scaling group. Specify `True` to make Auto Scaling to wait for the cool-down period associated with the Auto Scaling group to complete before initiating a scaling activity to set your Auto Scaling group to its new capacity.

Returns None

set_instance_health (***kwargs*)

Sets the health status of the specified instance.

For more information, see [Health Checks](#) in the *Auto Scaling Developer Guide* .

Request Syntax

```
response = client.set_instance_health(
    InstanceId='string',
    HealthStatus='string',
    ShouldRespectGracePeriod=True|False
)
```

Parameters

- **InstanceId** (*string*) – [REQUIRED]
The ID of the EC2 instance.
- **HealthStatus** (*string*) – [REQUIRED]
The health status of the instance. Set to `Healthy` if you want the instance to remain in service. Set to `Unhealthy` if you want the instance to be out of service. Auto Scaling will terminate and replace the unhealthy instance.
- **ShouldRespectGracePeriod** (*boolean*) – If the Auto Scaling group of the specified instance has a `HealthCheckGracePeriod` specified for the group, by default, this call will respect the grace period. Set this to `False` , if you do not want the call to respect the grace period associated with the group.

For more information, see the `HealthCheckGracePeriod` parameter description for `CreateAutoScalingGroup` .

Returns None

suspend_processes (***kwargs*)

Suspends the specified Auto Scaling processes for the specified Auto Scaling group. To suspend specific processes, use the `ScalingProcesses` parameter. To suspend all processes, omit the `ScalingProcesses` parameter.

Note that if you suspend either the `Launch` or `Terminate` process types, it can prevent other process types from functioning properly.

To resume processes that have been suspended, use `ResumeProcesses` .

For more information, see [Suspend and Resume Auto Scaling Processes](#) in the *Auto Scaling Developer Guide* .

Request Syntax

```
response = client.suspend_processes(
    AutoScalingGroupName='string',
    ScalingProcesses=[
        'string',
    ]
)
```

Parameters

- **AutoScalingGroupName** (*string*) – [REQUIRED]

The name or Amazon Resource Name (ARN) of the Auto Scaling group.

- **ScalingProcesses** (*list*) – One or more of the following processes:

- Launch
- Terminate
- HealthCheck
- ReplaceUnhealthy
- AZRebalance
- AlarmNotification
- ScheduledActions
- AddToLoadBalancer

– (*string*) –

Returns None

terminate_instance_in_auto_scaling_group (***kwargs*)

Terminates the specified instance and optionally adjusts the desired group size.

This call simply makes a termination request. The instances is not terminated immediately.

Request Syntax

```
response = client.terminate_instance_in_auto_scaling_group(
    InstanceId='string',
    ShouldDecrementDesiredCapacity=True|False
)
```

Parameters

- **InstanceId** (*string*) – [REQUIRED]

The ID of the EC2 instance.

- **ShouldDecrementDesiredCapacity** (*boolean*) – [REQUIRED]

If `true` , terminating this instance also decrements the size of the Auto Scaling group.

Return type dict

Returns

Response Syntax

```
{
    'Activity': {
        'ActivityId': 'string',
        'AutoScalingGroupName': 'string',
```

```
        'Description': 'string',
        'Cause': 'string',
        'StartTime': datetime(2015, 1, 1),
        'EndTime': datetime(2015, 1, 1),
        'StatusCode': 'WaitingForSpotInstanceRequestId'|'WaitingForSpotInstanceId',
        'StatusMessage': 'string',
        'Progress': 123,
        'Details': 'string'
    }
}
```

Response Structure

- *(dict)* –
 - **Activity** (*dict*) –
 - A scaling activity.
 - * **ActivityId** (*string*) –
 - The ID of the activity.
 - * **AutoScalingGroupName** (*string*) –
 - The name of the Auto Scaling group.
 - * **Description** (*string*) –
 - A friendly, more verbose description of the activity.
 - * **Cause** (*string*) –
 - The reason the activity began.
 - * **StartTime** (*datetime*) –
 - The start time of the activity.
 - * **EndTime** (*datetime*) –
 - The end time of the activity.
 - * **StatusCode** (*string*) –
 - The current status of the activity.
 - * **StatusMessage** (*string*) –
 - A friendly, more verbose description of the activity status.
 - * **Progress** (*integer*) –
 - A value between 0 and 100 that indicates the progress of the activity.
 - * **Details** (*string*) –
 - The details about the activity.

update_auto_scaling_group (**kwargs)

Updates the configuration for the specified Auto Scaling group.

To update an Auto Scaling group with a launch configuration with `InstanceMonitoring` set to `False`, you must first disable the collection of group metrics. Otherwise, you will get an error. If you have previously enabled the collection of group metrics, you can disable it using `DisableMetricsCollection`.

The new settings are registered upon the completion of this call. Any launch configuration settings take effect on any triggers after this call returns. Scaling activities that are currently in progress aren't affected.

Note the following:

- If you specify a new value for `MinSize` without specifying a value for `DesiredCapacity`, and the new `MinSize` is larger than the current size of the group, we implicitly call `SetDesiredCapacity` to set the size of the group to the new value of `MinSize`.

- If you specify a new value for `MaxSize` without specifying a value for `DesiredCapacity`, and the new `MaxSize` is smaller than the current size of the group, we implicitly call `SetDesiredCapacity` to set the size of the group to the new value of `MaxSize`.
- All other optional parameters are left unchanged if not specified.

Request Syntax

```
response = client.update_auto_scaling_group(
    AutoScalingGroupName='string',
    LaunchConfigurationName='string',
    MinSize=123,
    MaxSize=123,
    DesiredCapacity=123,
    DefaultCooldown=123,
    AvailabilityZones=[
        'string',
    ],
    HealthCheckType='string',
    HealthCheckGracePeriod=123,
    PlacementGroup='string',
    VPCZoneIdentifier='string',
    TerminationPolicies=[
        'string',
    ]
)
```

Parameters

- **AutoScalingGroupName** (*string*) – [REQUIRED]

The name of the Auto Scaling group.

- **LaunchConfigurationName** (*string*) – The name of the launch configuration.
- **MinSize** (*integer*) – The minimum size of the Auto Scaling group.
- **MaxSize** (*integer*) – The maximum size of the Auto Scaling group.
- **DesiredCapacity** (*integer*) – The number of EC2 instances that should be running in the Auto Scaling group. This number must be greater than or equal to the minimum size of the group and less than or equal to the maximum size of the group.
- **DefaultCooldown** (*integer*) – The amount of time, in seconds, after a scaling activity completes before another scaling activity can start. For more information, see [Understanding Auto Scaling Cooldowns](#).
- **AvailabilityZones** (*list*) – One or more Availability Zones for the group.
 - (*string*) –
- **HealthCheckType** (*string*) – The type of health check for the instances in the Auto Scaling group. The health check type can either be EC2 for Amazon EC2 or ELB for Elastic Load Balancing.
- **HealthCheckGracePeriod** (*integer*) – The amount of time, in seconds, that Auto Scaling waits before checking the health status of an instance. The grace period begins when the instance passes the system status and instance status checks from Amazon EC2. For more information, see ‘.
- **PlacementGroup** (*string*) – The name of the placement group into which you’ll launch your instances, if any. For more information, see [Placement Groups](#).
- **VPCZoneIdentifier** (*string*) – The ID of the subnet, if you are launching into a VPC. You can specify several subnets in a comma-separated list.

When you specify `VPCZoneIdentifier` with `AvailabilityZones`, ensure that the subnets’ Availability Zones match the values you specify for `AvailabilityZones`.

For more information, see [Auto Scaling and Amazon Virtual Private Cloud](#) in the *Auto Scaling Developer Guide*.

- **TerminationPolicies** (*list*) – A standalone termination policy or a list of termination policies used to select the instance to terminate. The policies are executed in the order that they are listed.

For more information, see [Choosing a Termination Policy for Your Auto Scaling Group](#) in the *Auto Scaling Developer Guide*.

– (*string*) –

Returns None

Paginators

The available paginators are:

- `AutoScaling.Paginator.DescribeAutoScalingGroups`
- `AutoScaling.Paginator.DescribeAutoScalingInstances`
- `AutoScaling.Paginator.DescribeLaunchConfigurations`
- `AutoScaling.Paginator.DescribeNotificationConfigurations`
- `AutoScaling.Paginator.DescribePolicies`
- `AutoScaling.Paginator.DescribeScalingActivities`
- `AutoScaling.Paginator.DescribeScheduledActions`
- `AutoScaling.Paginator.DescribeTags`

class `AutoScaling.Paginator.DescribeAutoScalingGroups`

```
paginator = client.get_paginator('describe_auto_scaling_groups')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `AutoScaling.Client.describe_auto_scaling_groups()`.

Request Syntax

```
response_iterator = paginator.paginate(
    AutoScalingGroupNames=[
        'string',
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **AutoScalingGroupNames** (*list*) – The group names.
– (*string*) –
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

– **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

– **PageSize** (*integer*) –

The size of each page.

– **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'AutoScalingGroups': [
    {
      'AutoScalingGroupName': 'string',
      'AutoScalingGroupARN': 'string',
      'LaunchConfigurationName': 'string',
      'MinSize': 123,
      'MaxSize': 123,
      'DesiredCapacity': 123,
      'DefaultCooldown': 123,
      'AvailabilityZones': [
        'string',
      ],
      'LoadBalancerNames': [
        'string',
      ],
      'HealthCheckType': 'string',
      'HealthCheckGracePeriod': 123,
      'Instances': [
        {
          'InstanceId': 'string',
          'AvailabilityZone': 'string',
          'LifecycleState': 'Pending'|'Pending:Wait'|'Pending:Proceed'|'
          'HealthStatus': 'string',
          'LaunchConfigurationName': 'string'
        },
      ],
      'CreatedTime': datetime(2015, 1, 1),
      'SuspendedProcesses': [
        {
          'ProcessName': 'string',
          'SuspensionReason': 'string'
        },
      ],
      'PlacementGroup': 'string',
      'VPCZoneIdentifier': 'string',
      'EnabledMetrics': [
        {
          'Metric': 'string',
          'Granularity': 'string'
        },
      ],
    },
  ],
}
```

```
        'Status': 'string',
        'Tags': [
            {
                'ResourceId': 'string',
                'ResourceType': 'string',
                'Key': 'string',
                'Value': 'string',
                'PropagateAtLaunch': True|False
            },
        ],
        'TerminationPolicies': [
            'string',
        ]
    },
],
}
```

Response Structure

- *(dict)* –
 - **AutoScalingGroups** (*list*) –

The groups.

- * *(dict)* –

Describes an Auto Scaling group.

- **AutoScalingGroupName** (*string*) –

The name of the group.

- **AutoScalingGroupARN** (*string*) –

The Amazon Resource Name (ARN) of the group.

- **LaunchConfigurationName** (*string*) –

The name of the associated launch configuration.

- **MinSize** (*integer*) –

The minimum size of the group.

- **MaxSize** (*integer*) –

The maximum size of the group.

- **DesiredCapacity** (*integer*) –

The desired size of the group.

- **DefaultCooldown** (*integer*) –

The number of seconds after a scaling activity completes before any further scaling activities can start.

- **AvailabilityZones** (*list*) –

One or more Availability Zones for the group.

- (*string*) –

- **LoadBalancerNames** (*list*) –

One or more load balancers associated with the group.

- (*string*) –

- **HealthCheckType** (*string*) –

The service of interest for the health status check, which can be either EC2 for Amazon EC2 or ELB for Elastic Load Balancing.

- **HealthCheckGracePeriod** (*integer*) –
The amount of time that Auto Scaling waits before checking an instance’s health status. The grace period begins when an instance comes into service.
- **Instances** (*list*) –
The EC2 instances associated with the group.
- (*dict*) –
Describes an EC2 instance.
- **InstanceId** (*string*) –
The ID of the instance.
- **AvailabilityZone** (*string*) –
The Availability Zone in which the instance is running.
- **LifecycleState** (*string*) –
A description of the current lifecycle state. Note that the `Quarantined` state is not used.
- **HealthStatus** (*string*) –
The health status of the instance.
- **LaunchConfigurationName** (*string*) –
The launch configuration associated with the instance.
- **CreatedTime** (*datetime*) –
The date and time the group was created.
- **SuspendedProcesses** (*list*) –
The suspended processes associated with the group.
- (*dict*) –
Describes an Auto Scaling process that has been suspended. For more information, see `ProcessType` .
- **ProcessName** (*string*) –
The name of the suspended process.
- **SuspensionReason** (*string*) –
The reason that the process was suspended.
- **PlacementGroup** (*string*) –
The name of the placement group into which you’ll launch your instances, if any. For more information, see [Placement Groups](#) .
- **VPCZoneIdentifier** (*string*) –
One or more subnet IDs, if applicable, separated by commas.

If you specify `VPCZoneIdentifier` and `AvailabilityZones` , ensure that the `AvailabilityZones` of the subnets match the values for `AvailabilityZones` .
- **EnabledMetrics** (*list*) –
The metrics enabled for the group.
- (*dict*) –
Describes an enabled metric.

- **Metric** (*string*) –
The name of the metric.
- `GroupMinSize`
- `GroupMaxSize`
- `GroupDesiredCapacity`
- `GroupInServiceInstances`
- `GroupPendingInstances`
- `GroupStandbyInstances`
- `GroupTerminatingInstances`
- `GroupTotalInstances`
- **Granularity** (*string*) –
The granularity of the metric. The only valid value is `1Minute`.
- **Status** (*string*) –
The current state of the group when `DeleteAutoScalingGroup` is in progress.
- **Tags** (*list*) –
The tags for the group.
- (*dict*) –
Describes a tag for an Auto Scaling group.
- **ResourceId** (*string*) –
The name of the group.
- **ResourceType** (*string*) –
The type of resource. The only supported value is `auto-scaling-group`.
- **Key** (*string*) –
The tag key.
- **Value** (*string*) –
The tag value.
- **PropagateAtLaunch** (*boolean*) –
Determines whether the tag is added to new instances as they are launched in the group.
- **TerminationPolicies** (*list*) –
The termination policies for the group.
- (*string*) –

`class AutoScaling.Paginator.DescribeAutoScalingInstances`

```
paginator = client.get_paginator('describe_auto_scaling_instances')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `AutoScaling.Client.describe_auto_scaling_instances()`.

Request Syntax

```
response_iterator = paginator.paginate(  
    InstanceIds=[
```

```

        'string',
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)

```

Parameters

- **InstanceIds** (*list*) – One or more Auto Scaling instances to describe, up to 50 instances. If you omit this parameter, all Auto Scaling instances are described. If you specify an ID that does not exist, it is ignored with no error.
 - (*string*) –
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```

{
    'AutoScalingInstances': [
        {
            'InstanceId': 'string',
            'AutoScalingGroupName': 'string',
            'AvailabilityZone': 'string',
            'LifecycleState': 'string',
            'HealthStatus': 'string',
            'LaunchConfigurationName': 'string'
        },
    ],
}

```

Response Structure

- (*dict*) –
 - **AutoScalingInstances** (*list*) –
The instances.
 - * (*dict*) –
Describes an EC2 instance associated with an Auto Scaling group.
 - **InstanceId** (*string*) –
The ID of the instance.

- **AutoScalingGroupName** (*string*) –
The name of the Auto Scaling group associated with the instance.
- **AvailabilityZone** (*string*) –
The Availability Zone for the instance.
- **LifecycleState** (*string*) –
The lifecycle state for the instance. For more information, see [Auto Scaling Instance States](#) in the *Auto Scaling Developer Guide*.
- **HealthStatus** (*string*) –
The health status of this instance. “Healthy” means that the instance is healthy and should remain in service. “Unhealthy” means that the instance is unhealthy and Auto Scaling should terminate and replace it.
- **LaunchConfigurationName** (*string*) –
The launch configuration associated with the instance.

class AutoScaling.Paginator.**DescribeLaunchConfigurations**

```
paginator = client.get_paginator('describe_launch_configurations')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `AutoScaling.Client.describe_launch_configurations()`.

Request Syntax

```
response_iterator = paginator.paginate(
    LaunchConfigurationNames=[
        'string',
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **LaunchConfigurationNames** (*list*) – The launch configuration names.
 - (*string*) –
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –

A token to specify where to start paginating. This is the NextToken from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'LaunchConfigurations': [
    {
      'LaunchConfigurationName': 'string',
      'LaunchConfigurationARN': 'string',
      'ImageId': 'string',
      'KeyName': 'string',
      'SecurityGroups': [
        'string',
      ],
      'ClassicLinkVPCId': 'string',
      'ClassicLinkVPCSecurityGroups': [
        'string',
      ],
      'UserData': 'string',
      'InstanceType': 'string',
      'KernelId': 'string',
      'RamdiskId': 'string',
      'BlockDeviceMappings': [
        {
          'VirtualName': 'string',
          'DeviceName': 'string',
          'Ebs': {
            'SnapshotId': 'string',
            'VolumeSize': 123,
            'VolumeType': 'string',
            'DeleteOnTermination': True|False,
            'Iops': 123
          },
          'NoDevice': True|False
        },
      ],
      'InstanceMonitoring': {
        'Enabled': True|False
      },
      'SpotPrice': 'string',
      'IamInstanceProfile': 'string',
      'CreatedTime': datetime(2015, 1, 1),
      'EbsOptimized': True|False,
      'AssociatePublicIpAddress': True|False,
      'PlacementTenancy': 'string'
    },
  ],
}
```

Response Structure

- (*dict*) –
 - **LaunchConfigurations** (*list*) –
 - The launch configurations.
- * (*dict*) –

Describes a launch configuration.

- **LaunchConfigurationName** (*string*) –
The name of the launch configuration.
- **LaunchConfigurationARN** (*string*) –
The Amazon Resource Name (ARN) of the launch configuration.
- **ImageId** (*string*) –
The ID of the Amazon Machine Image (AMI).
- **KeyName** (*string*) –
The name of the key pair.
- **SecurityGroups** (*list*) –
The security groups to associate with the instances.
- (*string*) –
- **ClassicLinkVPCId** (*string*) –
The ID of a ClassicLink-enabled VPC to link your EC2-Classic instances to. This parameter can only be used if you are launching EC2-Classic instances. For more information, see [ClassicLink](#) in the *Amazon Elastic Compute Cloud User Guide*.
- **ClassicLinkVPCSecurityGroups** (*list*) –
The IDs of one or more security groups for the VPC specified in `ClassicLinkVPCId`. This parameter is required if `ClassicLinkVPCId` is specified, and cannot be used otherwise. For more information, see [ClassicLink](#) in the *Amazon Elastic Compute Cloud User Guide*.
- (*string*) –
- **UserData** (*string*) –
The user data available to the instances.
- **InstanceType** (*string*) –
The instance type for the instances.
- **KernelId** (*string*) –
The ID of the kernel associated with the AMI.
- **RamdiskId** (*string*) –
The ID of the RAM disk associated with the AMI.
- **BlockDeviceMappings** (*list*) –
A block device mapping, which specifies the block devices for the instance.
- (*dict*) –
Describes a block device mapping.
- **VirtualName** (*string*) –
The name of the virtual device, `ephemeral0` to `ephemeral3`.
- **DeviceName** (*string*) –
The device name exposed to the EC2 instance (for example, `/dev/sdh` or `xvdh`).

- **Ebs** (*dict*) –
The information about the Amazon EBS volume.
- **SnapshotId** (*string*) –
The ID of the snapshot.
- **VolumeSize** (*integer*) –
The volume size, in gigabytes.

Valid values: If the volume type is `io1`, the minimum size of the volume is 10 GiB. If you specify `SnapshotId` and `VolumeSize`, `VolumeSize` must be equal to or larger than the size of the snapshot.

Default: If you create a volume from a snapshot and you don't specify a volume size, the default is the size of the snapshot.

Required: Required when the volume type is `io1`.
- **VolumeType** (*string*) –
The volume type.

Valid values: `standard` | `io1` | `gp2`

Default: `standard`
- **DeleteOnTermination** (*boolean*) –
Indicates whether to delete the volume on instance termination.

Default: `true`
- **Iops** (*integer*) –
For Provisioned IOPS (SSD) volumes only. The number of I/O operations per second (IOPS) to provision for the volume.

Valid values: Range is 100 to 4000.

Default: `None`
- **NoDevice** (*boolean*) –
Suppresses a device mapping.

If this parameter is true for the root device, the instance might fail the EC2 health check. Auto Scaling launches a replacement instance if the instance fails the health check.
- **InstanceMonitoring** (*dict*) –
Controls whether instances in this group are launched with detailed monitoring.
- **Enabled** (*boolean*) –
If `True`, instance monitoring is enabled.
- **SpotPrice** (*string*) –
The price to bid when launching Spot Instances.
- **IamInstanceProfile** (*string*) –
The name or Amazon Resource Name (ARN) of the instance profile associated with the IAM role for the instance.
- **CreatedTime** (*datetime*) –
The creation date and time for the launch configuration.

- **EbsOptimized** (*boolean*) –
Controls whether the instance is optimized for EBS I/O (`true`) or not (`false`).
- **AssociatePublicIpAddress** (*boolean*) –
Specifies whether the instances are associated with a public IP address (`true`) or not (`false`).
- **PlacementTenancy** (*string*) –
The tenancy of the instance, either default or dedicated . An instance with dedicated tenancy runs in an isolated, single-tenant hardware and can only be launched into a VPC.

class `AutoScaling.Paginator.DescribeNotificationConfigurations`

```
paginator = client.get_paginator('describe_notification_configurations')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `AutoScaling.Client.describe_notification_configurations()`.

Request Syntax

```
response_iterator = paginator.paginate(
    AutoScalingGroupNames=[
        'string',
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **AutoScalingGroupNames** (*list*) – The name of the group.
 - (*string*) –
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'NotificationConfigurations': [
    {
      'AutoScalingGroupName': 'string',
      'TopicARN': 'string',
      'NotificationType': 'string'
    },
  ],
}
```

Response Structure

- *(dict)* –
 - **NotificationConfigurations** (*list*) –

The notification configurations.

 - * *(dict)* –

Describes a notification.

 - **AutoScalingGroupName** (*string*) –

The name of the group.
 - **TopicARN** (*string*) –

The Amazon Resource Name (ARN) of the Amazon Simple Notification Service (SNS) topic.
 - **NotificationType** (*string*) –

The types of events for an action to start.

- autoscaling:EC2_INSTANCE_LAUNCH
- autoscaling:EC2_INSTANCE_LAUNCH_ERROR
- autoscaling:EC2_INSTANCE_TERMINATE
- autoscaling:EC2_INSTANCE_TERMINATE_ERROR
- autoscaling:TEST_NOTIFICATION

class AutoScaling.Paginator.**DescribePolicies**

```
paginator = client.get_paginator('describe_policies')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `AutoScaling.Client.describe_policies()`.

Request Syntax

```
response_iterator = paginator.paginate(
    AutoScalingGroupName='string',
    PolicyNames=[
        'string',
    ],
    PolicyTypes=[
        'string',
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

```
}  
)
```

Parameters

- **AutoScalingGroupName** (*string*) – The name of the group.
- **PolicyNames** (*list*) – One or more policy names or policy ARNs to be described. If you omit this list, all policy names are described. If an group name is provided, the results are limited to that group. This list is limited to 50 items. If you specify an unknown policy name, it is ignored with no error.
 - (*string*) –
- **PolicyTypes** (*list*) – One or more policy types. Valid values are `SimpleScaling` and `StepScaling`.
 - (*string*) –
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{  
  'ScalingPolicies': [  
    {  
      'AutoScalingGroupName': 'string',  
      'PolicyName': 'string',  
      'PolicyARN': 'string',  
      'PolicyType': 'string',  
      'AdjustmentType': 'string',  
      'MinAdjustmentStep': 123,  
      'MinAdjustmentMagnitude': 123,  
      'ScalingAdjustment': 123,  
      'Cooldown': 123,  
      'StepAdjustments': [  
        {  
          'MetricIntervalLowerBound': 123.0,  
          'MetricIntervalUpperBound': 123.0,  
          'ScalingAdjustment': 123  
        },  
      ],  
      'MetricAggregationType': 'string',  
      'EstimatedInstanceWarmup': 123,  
      'Alarms': [  
        {  
          'AlarmName': 'string',  
          'AlarmARN': 'string'  
        },  
      ],  
    },  
  ],  
}
```

```

    ],
    },
],
}

```

Response Structure

- *(dict)* –
 - **ScalingPolicies** (*list*) –

The scaling policies.

- * *(dict)* –

Describes a scaling policy.

- **AutoScalingGroupName** (*string*) –

The name of the Auto Scaling group associated with this scaling policy.

- **PolicyName** (*string*) –

The name of the scaling policy.

- **PolicyARN** (*string*) –

The Amazon Resource Name (ARN) of the policy.

- **PolicyType** (*string*) –

The policy type. Valid values are `SimpleScaling` and `StepScaling`.

- **AdjustmentType** (*string*) –

The adjustment type, which specifies how `ScalingAdjustment` is interpreted. Valid values are `ChangeInCapacity`, `ExactCapacity`, and `PercentChangeInCapacity`.

- **MinAdjustmentStep** (*integer*) –

Available for backward compatibility. Use `MinAdjustmentMagnitude` instead.

- **MinAdjustmentMagnitude** (*integer*) –

The minimum number of instances to scale. If the value of `AdjustmentType` is `PercentChangeInCapacity`, the scaling policy changes the `DesiredCapacity` of the Auto Scaling group by at least this many instances. Otherwise, the error is `ValidationError`.

- **ScalingAdjustment** (*integer*) –

The amount by which to scale, based on the specified adjustment type. A positive value adds to the current capacity while a negative number removes from the current capacity.

- **Cooldown** (*integer*) –

The amount of time, in seconds, after a scaling activity completes before any further trigger-related scaling activities can start.

- **StepAdjustments** (*list*) –

A set of adjustments that enable you to scale based on the size of the alarm breach.

- *(dict)* –

Describes an adjustment based on the difference between the value of the aggregated CloudWatch metric and the breach threshold that you’ve defined for the alarm.

For the following examples, suppose that you have an alarm with a breach threshold of 50:

- If you want the adjustment to be triggered when the metric is greater than or equal to 50 and less than 60, specify a lower bound of 0 and an upper bound of 10.
- If you want the adjustment to be triggered when the metric is greater than 40 and less than or equal to 50, specify a lower bound of -10 and an upper bound of 0.

There are a few rules for the step adjustments for your step policy:

- The ranges of your step adjustments can’t overlap or have a gap.
- At most one step adjustment can have a null lower bound. If one step adjustment has a negative lower bound, then there must be a step adjustment with a null lower bound.
- At most one step adjustment can have a null upper bound. If one step adjustment has a positive upper bound, then there must be a step adjustment with a null upper bound.
- The upper and lower bound can’t be null in the same step adjustment.

- **MetricIntervalLowerBound** (*float*) –

The lower bound for the difference between the alarm threshold and the CloudWatch metric. If the metric value is above the breach threshold, the lower bound is inclusive (the metric must be greater than or equal to the threshold plus the lower bound). Otherwise, it is exclusive (the metric must be greater than the threshold plus the lower bound). A null value indicates negative infinity.

- **MetricIntervalUpperBound** (*float*) –

The upper bound for the difference between the alarm threshold and the CloudWatch metric. If the metric value is above the breach threshold, the upper bound is exclusive (the metric must be less than the threshold plus the upper bound). Otherwise, it is inclusive (the metric must be less than or equal to the threshold plus the upper bound). A null value indicates positive infinity.

The upper bound must be greater than the lower bound.

- **ScalingAdjustment** (*integer*) –

The amount by which to scale, based on the specified adjustment type. A positive value adds to the current capacity while a negative number removes from the current capacity.

- **MetricAggregationType** (*string*) –

The aggregation type for the CloudWatch metrics. Valid values are `Minimum`, `Maximum`, and `Average`.

- **EstimatedInstanceWarmup** (*integer*) –

The estimated time, in seconds, until a newly launched in-

- stance can contribute to the CloudWatch metrics.
- **Alarms** (*list*) –
The CloudWatch alarms related to the policy.
- (*dict*) –
Describes an alarm.
- **AlarmName** (*string*) –
The name of the alarm.
- **AlarmARN** (*string*) –
The Amazon Resource Name (ARN) of the alarm.

class `AutoScaling.Paginator.DescribeScalingActivities`

```
paginator = client.get_paginator('describe_scaling_activities')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `AutoScaling.Client.describe_scaling_activities()`.

Request Syntax

```
response_iterator = paginator.paginate(
    ActivityIds=[
        'string',
    ],
    AutoScalingGroupName='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **ActivityIds** (*list*) – The activity IDs of the desired scaling activities. If this list is omitted, all activities are described. If the `AutoScalingGroupName` parameter is provided, the results are limited to that group. The list of requested activities cannot contain more than 50 items. If unknown activities are requested, they are ignored with no error.
 - (*string*) –
- **AutoScalingGroupName** (*string*) – The name of the group.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Activities': [
    {
      'ActivityId': 'string',
      'AutoScalingGroupName': 'string',
      'Description': 'string',
      'Cause': 'string',
      'StartTime': datetime(2015, 1, 1),
      'EndTime': datetime(2015, 1, 1),
      'StatusCode': 'WaitingForSpotInstanceRequestId'|'WaitingForSpotInstance...',
      'StatusMessage': 'string',
      'Progress': 123,
      'Details': 'string'
    },
  ],
}
```

Response Structure

- (*dict*) –

- **Activities** (*list*) –

The scaling activities.

- * (*dict*) –

Describes scaling activity, which is a long-running process that represents a change to your Auto Scaling group, such as changing its size or replacing an instance.

- **ActivityId** (*string*) –

The ID of the activity.

- **AutoScalingGroupName** (*string*) –

The name of the Auto Scaling group.

- **Description** (*string*) –

A friendly, more verbose description of the activity.

- **Cause** (*string*) –

The reason the activity began.

- **StartTime** (*datetime*) –

The start time of the activity.

- **EndTime** (*datetime*) –

The end time of the activity.

- **StatusCode** (*string*) –

The current status of the activity.

- **StatusMessage** (*string*) –

A friendly, more verbose description of the activity status.

- **Progress** (*integer*) –

A value between 0 and 100 that indicates the progress of the activity.

· **Details** (*string*) –

The details about the activity.

class AutoScaling.Paginator.**DescribeScheduledActions**

```
paginator = client.get_paginator('describe_scheduled_actions')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `AutoScaling.Client.describe_scheduled_actions()`.

Request Syntax

```
response_iterator = paginator.paginate(
    AutoScalingGroupName='string',
    ScheduledActionNames=[
        'string',
    ],
    StartTime=datetime(2015, 1, 1),
    EndTime=datetime(2015, 1, 1),
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **AutoScalingGroupName** (*string*) – The name of the group.
- **ScheduledActionNames** (*list*) – Describes one or more scheduled actions. If you omit this list, the call describes all scheduled actions. If you specify an unknown scheduled action it is ignored with no error.

You can describe up to a maximum of 50 instances with a single call. If there are more items to return, the call returns a token. To get the next set of items, repeat the call with the returned token in the `NextToken` parameter.

- (*string*) –
- **StartTime** (*datetime*) – The earliest scheduled start time to return. If scheduled action names are provided, this parameter is ignored.
- **EndTime** (*datetime*) – The latest scheduled start time to return. If scheduled action names are provided, this parameter is ignored.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'ScheduledUpdateGroupActions': [
    {
      'AutoScalingGroupName': 'string',
      'ScheduledActionName': 'string',
      'ScheduledActionARN': 'string',
      'Time': datetime(2015, 1, 1),
      'StartTime': datetime(2015, 1, 1),
      'EndTime': datetime(2015, 1, 1),
      'Recurrence': 'string',
      'MinSize': 123,
      'MaxSize': 123,
      'DesiredCapacity': 123
    },
  ],
}
```

Response Structure

- (*dict*) –
 - **ScheduledUpdateGroupActions** (*list*) –

The scheduled actions.

- * (*dict*) –

Describes a scheduled update to an Auto Scaling group.

- **AutoScalingGroupName** (*string*) –
The name of the group.
- **ScheduledActionName** (*string*) –
The name of the scheduled action.
- **ScheduledActionARN** (*string*) –
The Amazon Resource Name (ARN) of the scheduled action.
- **Time** (*datetime*) –
This parameter is deprecated; use `StartTime` instead.
- **StartTime** (*datetime*) –
The date and time that the action is scheduled to begin. This date and time can be up to one month in the future.

When `StartTime` and `EndTime` are specified with `Recurrence`, they form the boundaries of when the recurring action will start and stop.
- **EndTime** (*datetime*) –
The date and time that the action is scheduled to end. This date and time can be up to one month in the future.
- **Recurrence** (*string*) –
The recurring schedule for the action.
- **MinSize** (*integer*) –
The minimum size of the group.

- **MaxSize** (*integer*) –
The maximum size of the group.
- **DesiredCapacity** (*integer*) –
The number of instances you prefer to maintain in the group.

class `AutoScaling.Paginator.DescribeTags`

```
paginator = client.get_paginator('describe_tags')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `AutoScaling.Client.describe_tags()`.

Request Syntax

```
response_iterator = paginator.paginate(
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **Filters** (*list*) – A filter used to scope the tags to return.
 - (*dict*) –
Describes a filter.
 - * **Name** (*string*) –
The name of the filter. The valid values are: "auto-scaling-group" , "key" , "value" , and "propagate-at-launch".
 - * **Values** (*list*) –
The value of the filter.
 - (*string*) –
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Tags': [
    {
      'ResourceId': 'string',
      'ResourceType': 'string',
      'Key': 'string',
      'Value': 'string',
      'PropagateAtLaunch': True|False
    },
  ],
}
```

Response Structure

- (*dict*) –

- **Tags** (*list*) –

The tags.

- * (*dict*) –

Describes a tag for an Auto Scaling group.

- **ResourceId** (*string*) –

The name of the group.

- **ResourceType** (*string*) –

The type of resource. The only supported value is `auto-scaling-group`.

- **Key** (*string*) –

The tag key.

- **Value** (*string*) –

The tag value.

- **PropagateAtLaunch** (*boolean*) –

Determines whether the tag is added to new instances as they are launched in the group.

CloudFormation

Table of Contents

- CloudFormation
 - Client
 - Paginators
 - Service Resource
 - Event
 - Stack
 - StackResource
 - StackResourceSummary

Client**class CloudFormation.Client**

A low-level client representing AWS CloudFormation:

```
import boto3
```

```
client = boto3.client('cloudformation')
```

These are the available methods:

- `can_paginate()`
- `cancel_update_stack()`
- `create_stack()`
- `delete_stack()`
- `describe_stack_events()`
- `describe_stack_resource()`
- `describe_stack_resources()`
- `describe_stacks()`
- `estimate_template_cost()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_stack_policy()`
- `get_template()`
- `get_template_summary()`
- `get_waiter()`
- `list_stack_resources()`
- `list_stacks()`
- `set_stack_policy()`
- `signal_resource()`
- `update_stack()`
- `validate_template()`

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

cancel_update_stack (***kwargs*)

Cancels an update on the specified stack. If the call completes successfully, the stack will roll back the

update and revert to the previous stack configuration.

Note: Only stacks that are in the UPDATE_IN_PROGRESS state can be canceled.

Request Syntax

```
response = client.cancel_update_stack(  
    StackName='string'  
)
```

Parameters `StackName` (*string*) – [REQUIRED]

The name or the unique stack ID that is associated with the stack.

Returns None

`create_stack` (**kwargs)

Creates a stack as specified in the template. After the call completes successfully, the stack creation starts. You can check the status of the stack via the DescribeStacks API.

Request Syntax

```
response = client.create_stack(  
    StackName='string',  
    TemplateBody='string',  
    TemplateURL='string',  
    Parameters=[  
        {  
            'ParameterKey': 'string',  
            'ParameterValue': 'string',  
            'UsePreviousValue': True|False  
        },  
    ],  
    DisableRollback=True|False,  
    TimeoutInMinutes=123,  
    NotificationARNs=[  
        'string',  
    ],  
    Capabilities=[  
        'CAPABILITY_IAM',  
    ],  
    OnFailure='DO_NOTHING'|'ROLLBACK'|'DELETE',  
    StackPolicyBody='string',  
    StackPolicyURL='string',  
    Tags=[  
        {  
            'Key': 'string',  
            'Value': 'string'  
        },  
    ],  
)
```

Parameters

- **`StackName`** (*string*) – [REQUIRED]

The name that is associated with the stack. The name must be unique in the region in which you are creating the stack.

Note: A stack name can contain only alphanumeric characters (case sensitive)

and hyphens. It must start with an alphabetic character and cannot be longer than 255 characters.

- **TemplateBody** (*string*) – Structure containing the template body with a minimum length of 1 byte and a maximum length of 51,200 bytes. For more information, go to [Template Anatomy](#) in the AWS CloudFormation User Guide.

Conditional: You must specify either the `TemplateBody` or the `TemplateURL` parameter, but not both.

- **TemplateURL** (*string*) – Location of file containing the template body. The URL must point to a template (max size: 460,800 bytes) located in an S3 bucket in the same region as the stack. For more information, go to the [Template Anatomy](#) in the AWS CloudFormation User Guide.

Conditional: You must specify either the `TemplateBody` or the `TemplateURL` parameter, but not both.

- **Parameters** (*list*) – A list of `Parameter` structures that specify input parameters for the stack.
 - (*dict*) –

The Parameter data type.

- * **ParameterKey** (*string*) –

The key associated with the parameter. If you don't specify a key and value for a particular parameter, AWS CloudFormation uses the default value that is specified in your template.

- * **ParameterValue** (*string*) –

The value associated with the parameter.

- * **UsePreviousValue** (*boolean*) –

During a stack update, use the existing parameter value that the stack is using for a given parameter key. If you specify `true`, do not specify a parameter value.

- **DisableRollback** (*boolean*) – Set to `true` to disable rollback of the stack if stack creation failed. You can specify either `DisableRollback` or `OnFailure`, but not both.

Default: `false`

- **TimeoutInMinutes** (*integer*) – The amount of time that can pass before the stack status becomes `CREATE_FAILED`; if `DisableRollback` is not set or is set to `false`, the stack will be rolled back.
- **NotificationARNs** (*list*) – The Simple Notification Service (SNS) topic ARNs to publish stack related events. You can find your SNS topic ARNs using the [SNS console](#) or your Command Line Interface (CLI).
 - (*string*) –

- **Capabilities** (*list*) – A list of capabilities that you must specify before AWS CloudFormation can create or update certain stacks. Some stack templates might include resources that can affect permissions in your AWS account. For those stacks, you must explicitly acknowledge their capabilities by specifying this parameter.

Currently, the only valid value is `CAPABILITY_IAM`, which is required for the following resources: `AWS::IAM::AccessKey`, `AWS::IAM::Group`, `AWS::IAM::InstanceProfile`, `AWS::IAM::Policy`, `AWS::IAM::Role`, `AWS::IAM::User`, and `AWS::IAM::UserToGroupAddition`. If your stack template contains these resources, we recommend that you review any permissions associated with them. If you don't specify this parameter, this action returns an

InsufficientCapabilities error.

– (string) –

- **OnFailure** (string) – Determines what action will be taken if stack creation fails. This must be one of: DO_NOTHING, ROLLBACK, or DELETE. You can specify either OnFailure or DisableRollback , but not both.

Default: ROLLBACK

- **StackPolicyBody** (string) – Structure containing the stack policy body. For more information, go to [Prevent Updates to Stack Resources](#) in the AWS CloudFormation User Guide. You can specify either the StackPolicyBody or the StackPolicyURL parameter, but not both.
- **StackPolicyURL** (string) – Location of a file containing the stack policy. The URL must point to a policy (max size: 16KB) located in an S3 bucket in the same region as the stack. You can specify either the StackPolicyBody or the StackPolicyURL parameter, but not both.
- **Tags** (list) – A set of user-defined Tags to associate with this stack, represented by key/value pairs. Tags defined for the stack are propagated to EC2 resources that are created as part of the stack. A maximum number of 10 tags can be specified.
 - (dict) –

The Tag type is used by CreateStack in the Tags parameter. It allows you to specify a key/value pair that can be used to store information related to cost allocation for an AWS CloudFormation stack.

* **Key** (string) –

Required . A string used to identify this tag. You can specify a maximum of 128 characters for a tag key. Tags owned by Amazon Web Services (AWS) have the reserved prefix: aws : .

* **Value** (string) –

Required . A string containing the value for this tag. You can specify a maximum of 256 characters for a tag value.

Return type dict

Returns

Response Syntax

```
{
    'StackId': 'string'
}
```

Response Structure

- (dict) –

The output for a CreateStack action.

– **StackId** (string) –

Unique identifier of the stack.

delete_stack (**kwargs)

Deletes a specified stack. Once the call completes successfully, stack deletion starts. Deleted stacks do not show up in the DescribeStacks API if the deletion has been completed successfully.

Request Syntax

```
response = client.delete_stack(
    StackName='string'
)
```

Parameters `StackName` (*string*) – [REQUIRED]

The name or the unique stack ID that is associated with the stack.

Returns None

describe_stack_events (***kwargs*)

Returns all stack related events for a specified stack. For more information about a stack's event history, go to [Stacks](#) in the AWS CloudFormation User Guide.

Note: You can list events for stacks that have failed to create or have been deleted by specifying the unique stack identifier (stack ID).

Request Syntax

```
response = client.describe_stack_events(
    StackName='string',
    NextToken='string'
)
```

Parameters

- **StackName** (*string*) – The name or the unique stack ID that is associated with the stack, which are not always interchangeable:
 - Running stacks: You can specify either the stack's name or its unique stack ID.
 - Deleted stacks: You must specify the unique stack ID.
 Default: There is no default value.
- **NextToken** (*string*) – String that identifies the start of the next list of events, if there is one.

Default: There is no default value.

Return type dict

Returns

Response Syntax

```
{
  'StackEvents': [
    {
      'StackId': 'string',
      'EventId': 'string',
      'StackName': 'string',
      'LogicalResourceId': 'string',
      'PhysicalResourceId': 'string',
      'ResourceType': 'string',
      'Timestamp': datetime(2015, 1, 1),
      'ResourceStatus': 'CREATE_IN_PROGRESS' | 'CREATE_FAILED' | 'CREATE_COMPLETE',
      'ResourceStatusReason': 'string',
      'ResourceProperties': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
The output for a DescribeStackEvents action.

– **StackEvents** (*list*) –

A list of `StackEvents` structures.

* (*dict*) –

The `StackEvent` data type.

· **StackId** (*string*) –

The unique ID name of the instance of the stack.

· **EventId** (*string*) –

The unique ID of this event.

· **StackName** (*string*) –

The name associated with a stack.

· **LogicalResourceId** (*string*) –

The logical name of the resource specified in the template.

· **PhysicalResourceId** (*string*) –

The name or unique identifier associated with the physical instance of the resource.

· **ResourceType** (*string*) –

Type of resource. (For more information, go to [AWS Resource Types Reference](#) in the AWS CloudFormation User Guide.)

· **Timestamp** (*datetime*) –

Time the status was updated.

· **ResourceStatus** (*string*) –

Current status of the resource.

· **ResourceStatusReason** (*string*) –

Success/failure message associated with the resource.

· **ResourceProperties** (*string*) –

BLOB of the properties used to create the resource.

– **NextToken** (*string*) –

String that identifies the start of the next list of events, if there is one.

describe_stack_resource (***kwargs*)

Returns a description of the specified resource in the specified stack.

For deleted stacks, `DescribeStackResource` returns resource information for up to 90 days after the stack has been deleted.

Request Syntax

```
response = client.describe_stack_resource(
    StackName='string',
    LogicalResourceId='string'
)
```

Parameters

- **StackName** (*string*) – **[REQUIRED]**

The name or the unique stack ID that is associated with the stack, which are not always interchangeable:

- Running stacks: You can specify either the stack’s name or its unique stack ID.

- Deleted stacks: You must specify the unique stack ID.
Default: There is no default value.

- **LogicalResourceId** (*string*) – [REQUIRED]

The logical name of the resource as specified in the template.

Default: There is no default value.

Return type dict

Returns

Response Syntax

```
{
  'StackResourceDetail': {
    'StackName': 'string',
    'StackId': 'string',
    'LogicalResourceId': 'string',
    'PhysicalResourceId': 'string',
    'ResourceType': 'string',
    'LastUpdatedTimestamp': datetime(2015, 1, 1),
    'ResourceStatus': 'CREATE_IN_PROGRESS' | 'CREATE_FAILED' | 'CREATE_COMPLETE' | 'DELETE_IN_PROGRESS' | 'DELETE_FAILED' | 'DELETE_COMPLETE',
    'ResourceStatusReason': 'string',
    'Description': 'string',
    'Metadata': 'string'
  }
}
```

Response Structure

- (*dict*) –

The output for a DescribeStackResource action.

- **StackResourceDetail** (*dict*) –

A `StackResourceDetail` structure containing the description of the specified resource in the specified stack.

- * **StackName** (*string*) –

The name associated with the stack.

- * **StackId** (*string*) –

Unique identifier of the stack.

- * **LogicalResourceId** (*string*) –

The logical name of the resource specified in the template.

- * **PhysicalResourceId** (*string*) –

The name or unique identifier that corresponds to a physical instance ID of a resource supported by AWS CloudFormation.

- * **ResourceType** (*string*) –

Type of resource. ((For more information, go to [AWS Resource Types Reference](#) in the AWS CloudFormation User Guide.)

- * **LastUpdatedTimestamp** (*datetime*) –

Time the status was updated.

- * **ResourceStatus** (*string*) –

Current status of the resource.

- * **ResourceStatusReason** (*string*) –

Success/failure message associated with the resource.

- * **Description** (*string*) –

User defined description associated with the resource.

- * **Metadata** (*string*) –

The JSON format content of the `Metadata` attribute declared for the resource. For more information, see [Metadata Attribute](#) in the AWS CloudFormation User Guide.

describe_stack_resources (***kwargs*)

Returns AWS resource descriptions for running and deleted stacks. If `StackName` is specified, all the associated resources that are part of the stack are returned. If `PhysicalResourceId` is specified, the associated resources of the stack that the resource belongs to are returned.

Note: Only the first 100 resources will be returned. If your stack has more resources than this, you should use `ListStackResources` instead.

For deleted stacks, `DescribeStackResources` returns resource information for up to 90 days after the stack has been deleted.

You must specify either `StackName` or `PhysicalResourceId`, but not both. In addition, you can specify `LogicalResourceId` to filter the returned result. For more information about resources, the `LogicalResourceId` and `PhysicalResourceId`, go to the [AWS CloudFormation User Guide](#).

Note: A `ValidationError` is returned if you specify both `StackName` and `PhysicalResourceId` in the same request.

Request Syntax

```
response = client.describe_stack_resources(  
    StackName='string',  
    LogicalResourceId='string',  
    PhysicalResourceId='string'  
)
```

Parameters

- **StackName** (*string*) – The name or the unique stack ID that is associated with the stack, which are not always interchangeable:
 - Running stacks: You can specify either the stack’s name or its unique stack ID.
 - Deleted stacks: You must specify the unique stack ID.

Default: There is no default value.

Required: Conditional. If you do not specify `StackName`, you must specify `PhysicalResourceId`.

- **LogicalResourceId** (*string*) – The logical name of the resource as specified in the template.

Default: There is no default value.

- **PhysicalResourceId** (*string*) – The name or unique identifier that corresponds to a physical instance ID of a resource supported by AWS CloudFormation.

For example, for an Amazon Elastic Compute Cloud (EC2) instance, `PhysicalResourceId` corresponds to the `InstanceId`. You can pass the EC2 `InstanceId` to `DescribeStackResources` to find which stack the instance belongs to and what other resources are part of the stack.

Required: Conditional. If you do not specify `PhysicalResourceId`, you must specify `StackName`.

Default: There is no default value.

Return type dict

Returns

Response Syntax

```
{
  'StackResources': [
    {
      'StackName': 'string',
      'StackId': 'string',
      'LogicalResourceId': 'string',
      'PhysicalResourceId': 'string',
      'ResourceType': 'string',
      'Timestamp': datetime(2015, 1, 1),
      'ResourceStatus': 'CREATE_IN_PROGRESS' | 'CREATE_FAILED' | 'CREATE_COMPLETE',
      'ResourceStatusReason': 'string',
      'Description': 'string'
    },
  ],
}
```

Response Structure

- (*dict*) –

The output for a `DescribeStackResources` action.

- **StackResources** (*list*) –

A list of `StackResource` structures.

- * (*dict*) –

The `StackResource` data type.

- **StackName** (*string*) –

The name associated with the stack.

- **StackId** (*string*) –

Unique identifier of the stack.

- **LogicalResourceId** (*string*) –

The logical name of the resource specified in the template.

- **PhysicalResourceId** (*string*) –

The name or unique identifier that corresponds to a physical instance ID of a resource supported by AWS CloudFormation.

- **ResourceType** (*string*) –

Type of resource. (For more information, go to [AWS Resource Types Reference](#) in the AWS CloudFormation User Guide.)

- **Timestamp** (*datetime*) –

Time the status was updated.

- **ResourceStatus** (*string*) –

Current status of the resource.

- **ResourceStatusReason** (*string*) –

Success/failure message associated with the resource.

- **Description** (*string*) –

User defined description associated with the resource.

describe_stacks (***kwargs*)

Returns the description for the specified stack; if no stack name was specified, then it returns the description for all the stacks created.

Request Syntax

```
response = client.describe_stacks(
    StackName='string',
    NextToken='string'
)
```

Parameters

- **StackName** (*string*) – The name or the unique stack ID that is associated with the stack, which are not always interchangeable:
 - Running stacks: You can specify either the stack’s name or its unique stack ID.
 - Deleted stacks: You must specify the unique stack ID.Default: There is no default value.
- **NextToken** (*string*) – String that identifies the start of the next list of stacks, if there is one.

Return type dict

Returns**Response Syntax**

```
{
    'Stacks': [
        {
            'StackId': 'string',
            'StackName': 'string',
            'Description': 'string',
            'Parameters': [
                {
                    'ParameterKey': 'string',
                    'ParameterValue': 'string',
                    'UsePreviousValue': True|False
                },
            ],
            'CreationTime': datetime(2015, 1, 1),
            'LastUpdatedTime': datetime(2015, 1, 1),
            'StackStatus': 'CREATE_IN_PROGRESS'|'CREATE_FAILED'|'CREATE_COMPLETE'|
            'StackStatusReason': 'string',
            'DisableRollback': True|False,
            'NotificationARNs': [
                'string',
            ],
            'TimeoutInMinutes': 123,
            'Capabilities': [
                'CAPABILITY_IAM',
            ],
            'Outputs': [
                {
                    'OutputKey': 'string',
                    'OutputValue': 'string',
```



```

        'Description': 'string'
    },
],
'Tags': [
    {
        'Key': 'string',
        'Value': 'string'
    },
]
},
],
'NextToken': 'string'
}

```

Response Structure

- (*dict*) –

The output for a DescribeStacks action.

- **Stacks** (*list*) –

A list of stack structures.

- * (*dict*) –

The Stack data type.

- **StackId** (*string*) –

Unique identifier of the stack.

- **StackName** (*string*) –

The name associated with the stack.

- **Description** (*string*) –

User defined description associated with the stack.

- **Parameters** (*list*) –

A list of Parameter structures.

- (*dict*) –

The Parameter data type.

- **ParameterKey** (*string*) –

The key associated with the parameter. If you don't specify a key and value for a particular parameter, AWS CloudFormation uses the default value that is specified in your template.

- **ParameterValue** (*string*) –

The value associated with the parameter.

- **UsePreviousValue** (*boolean*) –

During a stack update, use the existing parameter value that the stack is using for a given parameter key. If you specify `true`, do not specify a parameter value.

- **CreationTime** (*datetime*) –

Time at which the stack was created.

- **LastUpdatedTime** (*datetime*) –

The time the stack was last updated. This field will only be returned if the stack has been updated at least once.

- **StackStatus** (*string*) –

Current status of the stack.

- **StackStatusReason** (*string*) –
Success/failure message associated with the stack status.
- **DisableRollback** (*boolean*) –
Boolean to enable or disable rollback on stack creation failures:
 - `true` : disable rollback
 - `false` : enable rollback
- **NotificationARNs** (*list*) –
SNS topic ARNs to which stack related events are published.
 - (*string*) –
- **TimeoutInMinutes** (*integer*) –
The amount of time within which stack creation should complete.
- **Capabilities** (*list*) –
The capabilities allowed in the stack.
 - (*string*) –
- **Outputs** (*list*) –
A list of output structures.
 - (*dict*) –
The Output data type.
- **OutputKey** (*string*) –
The key associated with the output.
- **OutputValue** (*string*) –
The value associated with the output.
- **Description** (*string*) –
User defined description associated with the output.
- **Tags** (*list*) –
A list of `Tag`s that specify cost allocation information for the stack.
 - (*dict*) –
The `Tag` type is used by `CreateStack` in the `Tags` parameter. It allows you to specify a key/value pair that can be used to store information related to cost allocation for an AWS CloudFormation stack.
- **Key** (*string*) –
Required . A string used to identify this tag. You can specify a maximum of 128 characters for a tag key. Tags owned by Amazon Web Services (AWS) have the reserved prefix: `aws:`
 -
- **Value** (*string*) –
Required . A string containing the value for this tag. You can specify a maximum of 256 characters for a tag value.
- **NextToken** (*string*) – String that identifies the start of the next list of stacks, if there is one.

estimate_template_cost (***kwargs*)

Returns the estimated monthly cost of a template. The return value is an AWS Simple Monthly Calculator

URL with a query string that describes the resources required to run the template.

Request Syntax

```
response = client.estimate_template_cost(
    TemplateBody='string',
    TemplateURL='string',
    Parameters=[
        {
            'ParameterKey': 'string',
            'ParameterValue': 'string',
            'UsePreviousValue': True|False
        },
    ]
)
```

Parameters

- **TemplateBody** (*string*) – Structure containing the template body with a minimum length of 1 byte and a maximum length of 51,200 bytes. (For more information, go to [Template Anatomy](#) in the AWS CloudFormation User Guide.)

Conditional: You must pass `TemplateBody` or `TemplateURL`. If both are passed, only `TemplateBody` is used.

- **TemplateURL** (*string*) – Location of file containing the template body. The URL must point to a template located in an S3 bucket in the same region as the stack. For more information, go to [Template Anatomy](#) in the AWS CloudFormation User Guide.

Conditional: You must pass `TemplateURL` or `TemplateBody`. If both are passed, only `TemplateBody` is used.

- **Parameters** (*list*) – A list of `Parameter` structures that specify input parameters.

– (*dict*) –

The Parameter data type.

- * **ParameterKey** (*string*) –

The key associated with the parameter. If you don't specify a key and value for a particular parameter, AWS CloudFormation uses the default value that is specified in your template.

- * **ParameterValue** (*string*) –

The value associated with the parameter.

- * **UsePreviousValue** (*boolean*) –

During a stack update, use the existing parameter value that the stack is using for a given parameter key. If you specify `true`, do not specify a parameter value.

Return type dict

Returns

Response Syntax

```
{
    'Url': 'string'
}
```

Response Structure

- (*dict*) –

The output for a EstimateTemplateCost action.

- **Url** (*string*) –

An AWS Simple Monthly Calculator URL with a query string that describes the resources required to run the template.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to *ClientMethod*.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type L{botocore.paginate.Paginator}

Returns A paginator object.

get_stack_policy (***kwargs*)

Returns the stack policy for a specified stack. If a stack doesn't have a policy, a null value is returned.

Request Syntax

```
response = client.get_stack_policy(  
    StackName='string'  
)
```

Parameters **StackName** (*string*) – [REQUIRED]

The name or unique stack ID that is associated with the stack whose policy you want to get.

Return type dict

Returns

Response Syntax

```
{  
    'StackPolicyBody': 'string'  
}
```

Response Structure

- (*dict*) –

The output for the GetStackPolicy action.

– **StackPolicyBody** (*string*) –

Structure containing the stack policy body. (For more information, go to [Prevent Updates to Stack Resources](#) in the AWS CloudFormation User Guide.)

get_template (***kwargs*)

Returns the template body for a specified stack. You can get the template for running or deleted stacks. For deleted stacks, GetTemplate returns the template for up to 90 days after the stack has been deleted.

Note: If the template does not exist, a `ValidationError` is returned.

Request Syntax

```
response = client.get_template(
    StackName='string'
)
```

Parameters **StackName** (*string*) – **[REQUIRED]**

The name or the unique stack ID that is associated with the stack, which are not always interchangeable:

- Running stacks: You can specify either the stack's name or its unique stack ID.
- Deleted stacks: You must specify the unique stack ID.

Default: There is no default value.

Return type dict

Returns

Response Syntax

```
{
    'TemplateBody': 'string'
}
```

Response Structure

- (*dict*) –

The output for GetTemplate action.

– **TemplateBody** (*string*) –

Structure containing the template body. (For more information, go to [Template Anatomy](#) in the AWS CloudFormation User Guide.)

get_template_summary (***kwargs*)

Returns information about a new or existing template. The `GetTemplateSummary` action is useful for viewing parameter information, such as default parameter values and parameter types, before you create or update a stack.

You can use the `GetTemplateSummary` action when you submit a template, or you can get template information for a running or deleted stack.

For deleted stacks, `GetTemplateSummary` returns the template information for up to 90 days after the stack has been deleted. If the template does not exist, a `ValidationError` is returned.

Request Syntax

```
response = client.get_template_summary(
    TemplateBody='string',
    TemplateURL='string',
```

```
StackName='string'  
)
```

Parameters

- **TemplateBody** (*string*) – Structure containing the template body with a minimum length of 1 byte and a maximum length of 51,200 bytes. For more information about templates, see [Template Anatomy](#) in the AWS CloudFormation User Guide.

Conditional: You must specify only one of the following parameters: `StackName`, `TemplateBody`, or `TemplateURL`.

- **TemplateURL** (*string*) – Location of file containing the template body. The URL must point to a template (max size: 460,800 bytes) located in an Amazon S3 bucket. For more information about templates, see [Template Anatomy](#) in the AWS CloudFormation User Guide.

Conditional: You must specify only one of the following parameters: `StackName`, `TemplateBody`, or `TemplateURL`.

- **StackName** (*string*) – The name or the stack ID that is associated with the stack, which are not always interchangeable. For running stacks, you can specify either the stack's name or its unique stack ID. For deleted stack, you must specify the unique stack ID.

Conditional: You must specify only one of the following parameters: `StackName`, `TemplateBody`, or `TemplateURL`.

Return type dict

Returns

Response Syntax

```
{  
  'Parameters': [  
    {  
      'ParameterKey': 'string',  
      'DefaultValue': 'string',  
      'ParameterType': 'string',  
      'NoEcho': True|False,  
      'Description': 'string',  
      'ParameterConstraints': {  
        'AllowedValues': [  
          'string',  
        ]  
      }  
    },  
  ],  
  'Description': 'string',  
  'Capabilities': [  
    'CAPABILITY_IAM',  
  ],  
  'CapabilitiesReason': 'string',  
  'Version': 'string',  
  'Metadata': 'string'  
}
```

Response Structure

- (*dict*) –

The output for the `GetTemplateSummary` action.

– **Parameters** (*list*) –

A list of parameter declarations that describe various properties for each parameter.

* (*dict*) –

The ParameterDeclaration data type.

· **ParameterKey** (*string*) –

The name that is associated with the parameter.

· **DefaultValue** (*string*) –

The default value of the parameter.

· **ParameterType** (*string*) –

The type of parameter.

· **NoEcho** (*boolean*) –

Flag that indicates whether the parameter value is shown as plain text in logs and in the AWS Management Console.

· **Description** (*string*) –

The description that is associate with the parameter.

· **ParameterConstraints** (*dict*) –

The criteria that AWS CloudFormation uses to validate parameter values.

· **AllowedValues** (*list*) –

A list of values that are permitted for a parameter.

· (*string*) –

– **Description** (*string*) –

The value that is defined in the `Description` property of the template.

– **Capabilities** (*list*) –

The capabilities found within the template. Currently, AWS CloudFormation supports only the `CAPABILITY_IAM` capability. If your template contains IAM resources, you must specify the `CAPABILITY_IAM` value for this parameter when you use the `CreateStack` or `UpdateStack` actions with your template; otherwise, those actions return an `InsufficientCapabilities` error.

* (*string*) –

– **CapabilitiesReason** (*string*) –

The list of resources that generated the values in the `Capabilities` response element.

– **Version** (*string*) –

The AWS template format version, which identifies the capabilities of the template.

– **Metadata** (*string*) –

The value that is defined for the `Metadata` property of the template.

get_waiter (*waiter_name*)

list_stack_resources (***kwargs*)

Returns descriptions of all resources of the specified stack.

For deleted stacks, `ListStackResources` returns resource information for up to 90 days after the stack has been deleted.

Request Syntax

```
response = client.list_stack_resources(
    StackName='string',
    NextToken='string'
)
```

Parameters

- **StackName** (*string*) – **[REQUIRED]**

The name or the unique stack ID that is associated with the stack, which are not always interchangeable:

- Running stacks: You can specify either the stack’s name or its unique stack ID.
- Deleted stacks: You must specify the unique stack ID.

Default: There is no default value.

- **NextToken** (*string*) – String that identifies the start of the next list of stack resource summaries, if there is one.

Default: There is no default value.

Return type dict

Returns

Response Syntax

```
{
  'StackResourceSummaries': [
    {
      'LogicalResourceId': 'string',
      'PhysicalResourceId': 'string',
      'ResourceType': 'string',
      'LastUpdatedTimestamp': datetime(2015, 1, 1),
      'ResourceStatus': 'CREATE_IN_PROGRESS' | 'CREATE_FAILED' | 'CREATE_COMPLETE',
      'ResourceStatusReason': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

The output for a ListStackResources action.

- **StackResourceSummaries** (*list*) –

A list of StackResourceSummary structures.

- * (*dict*) –

Contains high-level information about the specified stack resource.

- **LogicalResourceId** (*string*) –

The logical name of the resource specified in the template.

- **PhysicalResourceId** (*string*) –

The name or unique identifier that corresponds to a physical instance ID of the resource.

- **ResourceType** (*string*) –

Type of resource. (For more information, go to [AWS Resource Types Reference](#) in the AWS CloudFormation User Guide.)

- **LastUpdatedTimestamp** (*datetime*) –

Time the status was updated.

- **ResourceStatus** (*string*) –

Current status of the resource.

- **ResourceStatusReason** (*string*) –

Success/failure message associated with the resource.

- **NextToken** (*string*) –

String that identifies the start of the next list of stack resources, if there is one.

list_stacks (***kwargs*)

Returns the summary information for stacks whose status matches the specified `StackStatusFilter`. Summary information for stacks that have been deleted is kept for 90 days after the stack is deleted. If no `StackStatusFilter` is specified, summary information for all stacks is returned (including existing stacks and stacks that have been deleted).

Request Syntax

```
response = client.list_stacks(
    NextToken='string',
    StackStatusFilter=[
        'CREATE_IN_PROGRESS' | 'CREATE_FAILED' | 'CREATE_COMPLETE' | 'ROLLBACK_IN_PROGRESS' | 'ROLLBACK_COMPLETE'
    ]
)
```

Parameters

- **NextToken** (*string*) – String that identifies the start of the next list of stacks, if there is one.

Default: There is no default value.

- **StackStatusFilter** (*list*) – Stack status to use as a filter. Specify one or more stack status codes to list only stacks with the specified status codes. For a complete list of stack status codes, see the `StackStatus` parameter of the `Stack` data type.
- (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'StackSummaries': [
        {
            'StackId': 'string',
            'StackName': 'string',
            'TemplateDescription': 'string',
            'CreationTime': datetime(2015, 1, 1),
            'LastUpdatedTime': datetime(2015, 1, 1),
            'DeletionTime': datetime(2015, 1, 1),
            'StackStatus': 'CREATE_IN_PROGRESS' | 'CREATE_FAILED' | 'CREATE_COMPLETE' | 'ROLLBACK_IN_PROGRESS' | 'ROLLBACK_COMPLETE',
            'StackStatusReason': 'string'
        },
    ],
}
```

```
],  
  'NextToken': 'string'  
}
```

Response Structure

- (*dict*) –

The output for ListStacks action.

- **StackSummaries** (*list*) –

A list of `StackSummary` structures containing information about the specified stacks.

- * (*dict*) –

The StackSummary Data Type

- **StackId** (*string*) –

Unique stack identifier.

- **StackName** (*string*) –

The name associated with the stack.

- **TemplateDescription** (*string*) –

The template description of the template used to create the stack.

- **CreationTime** (*datetime*) –

The time the stack was created.

- **LastUpdatedTime** (*datetime*) –

The time the stack was last updated. This field will only be returned if the stack has been updated at least once.

- **DeletionTime** (*datetime*) –

The time the stack was deleted.

- **StackStatus** (*string*) –

The current status of the stack.

- **StackStatusReason** (*string*) –

Success/Failure message associated with the stack status.

- **NextToken** (*string*) –

String that identifies the start of the next list of stacks, if there is one.

set_stack_policy (***kwargs*)

Sets a stack policy for a specified stack.

Request Syntax

```
response = client.set_stack_policy(  
    StackName='string',  
    StackPolicyBody='string',  
    StackPolicyURL='string'  
)
```

Parameters

- **StackName** (*string*) – [REQUIRED]

The name or unique stack ID that you want to associate a policy with.

- **StackPolicyBody** (*string*) – Structure containing the stack policy body. For more information, go to [Prevent Updates to Stack Resources](#) in the AWS CloudFormation User Guide. You can specify either the `StackPolicyBody` or the `StackPolicyURL` parameter, but not both.
- **StackPolicyURL** (*string*) – Location of a file containing the stack policy. The URL must point to a policy (max size: 16KB) located in an S3 bucket in the same region as the stack. You can specify either the `StackPolicyBody` or the `StackPolicyURL` parameter, but not both.

Returns None

signal_resource (***kwargs*)

Sends a signal to the specified resource with a success or failure status. You can use the `SignalResource` API in conjunction with a creation policy or update policy. AWS CloudFormation doesn't proceed with a stack creation or update until resources receive the required number of signals or the timeout period is exceeded. The `SignalResource` API is useful in cases where you want to send signals from anywhere other than an Amazon EC2 instance.

Request Syntax

```
response = client.signal_resource(
    StackName='string',
    LogicalResourceId='string',
    UniqueId='string',
    Status='SUCCESS'|'FAILURE'
)
```

Parameters

- **StackName** (*string*) – [REQUIRED]
The stack name or unique stack ID that includes the resource that you want to signal.
- **LogicalResourceId** (*string*) – [REQUIRED]
The logical ID of the resource that you want to signal. The logical ID is the name of the resource that given in the template.
- **UniqueId** (*string*) – [REQUIRED]
A unique ID of the signal. When you signal Amazon EC2 instances or Auto Scaling groups, specify the instance ID that you are signaling as the unique ID. If you send multiple signals to a single resource (such as signaling a wait condition), each signal requires a different unique ID.
- **Status** (*string*) – [REQUIRED]
The status of the signal, which is either success or failure. A failure signal causes AWS CloudFormation to immediately fail the stack creation or update.

Returns None

update_stack (***kwargs*)

Updates a stack as specified in the template. After the call completes successfully, the stack update starts. You can check the status of the stack via the `DescribeStacks` action.

To get a copy of the template for an existing stack, you can use the `GetTemplate` action.

Tags that were associated with this stack during creation time will still be associated with the stack after an `UpdateStack` operation.

For more information about creating an update template, updating a stack, and monitoring the progress of the update, see [Updating a Stack](#).

Request Syntax

```
response = client.update_stack(  
    StackName='string',  
    TemplateBody='string',  
    TemplateURL='string',  
    UsePreviousTemplate=True|False,  
    StackPolicyDuringUpdateBody='string',  
    StackPolicyDuringUpdateURL='string',  
    Parameters=[  
        {  
            'ParameterKey': 'string',  
            'ParameterValue': 'string',  
            'UsePreviousValue': True|False  
        },  
    ],  
    Capabilities=[  
        'CAPABILITY_IAM',  
    ],  
    StackPolicyBody='string',  
    StackPolicyURL='string',  
    NotificationARNs=[  
        'string',  
    ]  
)
```

Parameters

- **StackName** (*string*) – [REQUIRED]

The name or unique stack ID of the stack to update.

- **TemplateBody** (*string*) – Structure containing the template body with a minimum length of 1 byte and a maximum length of 51,200 bytes. (For more information, go to [Template Anatomy](#) in the AWS CloudFormation User Guide.)

Conditional: You must specify either the `TemplateBody` or the `TemplateURL` parameter, but not both.

- **TemplateURL** (*string*) – Location of file containing the template body. The URL must point to a template located in an S3 bucket in the same region as the stack. For more information, go to [Template Anatomy](#) in the AWS CloudFormation User Guide.

Conditional: You must specify either the `TemplateBody` or the `TemplateURL` parameter, but not both.

- **UsePreviousTemplate** (*boolean*) – Reuse the existing template that is associated with the stack that you are updating.
- **StackPolicyDuringUpdateBody** (*string*) – Structure containing the temporary overriding stack policy body. You can specify either the `StackPolicyDuringUpdateBody` or the `StackPolicyDuringUpdateURL` parameter, but not both.

If you want to update protected resources, specify a temporary overriding stack policy during this update. If you do not specify a stack policy, the current policy that is associated with the stack will be used.

- **StackPolicyDuringUpdateURL** (*string*) – Location of a file containing the temporary overriding stack policy. The URL must point to a policy (max size: 16KB) located in an S3 bucket in the same region as the stack. You can specify either the `StackPolicyDuringUpdateBody` or the `StackPolicyDuringUpdateURL` parameter, but not both.

If you want to update protected resources, specify a temporary overriding stack

policy during this update. If you do not specify a stack policy, the current policy that is associated with the stack will be used.

- **Parameters** (*list*) – A list of `Parameter` structures that specify input parameters for the stack. For more information, see the [Parameter](#) data type.
 - (*dict*) –

The Parameter data type.

- * **ParameterKey** (*string*) –

The key associated with the parameter. If you don't specify a key and value for a particular parameter, AWS CloudFormation uses the default value that is specified in your template.

- * **ParameterValue** (*string*) –

The value associated with the parameter.

- * **UsePreviousValue** (*boolean*) –

During a stack update, use the existing parameter value that the stack is using for a given parameter key. If you specify `true`, do not specify a parameter value.

- **Capabilities** (*list*) – A list of capabilities that you must specify before AWS CloudFormation can create or update certain stacks. Some stack templates might include resources that can affect permissions in your AWS account. For those stacks, you must explicitly acknowledge their capabilities by specifying this parameter. Currently, the only valid value is `CAPABILITY_IAM`, which is required for the following resources: `AWS::IAM::AccessKey`, `AWS::IAM::Group`, `AWS::IAM::InstanceProfile`, `AWS::IAM::Policy`, `AWS::IAM::Role`, `AWS::IAM::User`, and `AWS::IAM::UserToGroupAddition`. If your stack template contains these resources, we recommend that you review any permissions associated with them. If you don't specify this parameter, this action returns an `InsufficientCapabilities` error.

– (*string*) –

- **StackPolicyBody** (*string*) – Structure containing a new stack policy body. You can specify either the `StackPolicyBody` or the `StackPolicyURL` parameter, but not both.

You might update the stack policy, for example, in order to protect a new resource that you created during a stack update. If you do not specify a stack policy, the current policy that is associated with the stack is unchanged.

- **StackPolicyURL** (*string*) – Location of a file containing the updated stack policy. The URL must point to a policy (max size: 16KB) located in an S3 bucket in the same region as the stack. You can specify either the `StackPolicyBody` or the `StackPolicyURL` parameter, but not both.

You might update the stack policy, for example, in order to protect a new resource that you created during a stack update. If you do not specify a stack policy, the current policy that is associated with the stack is unchanged.

- **NotificationARNs** (*list*) – Update the ARNs for the Amazon SNS topics that are associated with the stack.

– (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'StackId': 'string'
```

```
}
```

Response Structure

- (*dict*) –
The output for a UpdateStack action.
 - **StackId** (*string*) –
Unique identifier of the stack.

validate_template (***kwargs*)
Validates a specified template.

Request Syntax

```
response = client.validate_template(  
    TemplateBody='string',  
    TemplateURL='string'  
)
```

Parameters

- **TemplateBody** (*string*) – Structure containing the template body with a minimum length of 1 byte and a maximum length of 51,200 bytes. For more information, go to [Template Anatomy](#) in the AWS CloudFormation User Guide.

Conditional: You must pass `TemplateURL` or `TemplateBody` . If both are passed, only `TemplateBody` is used.

- **TemplateURL** (*string*) – Location of file containing the template body. The URL must point to a template (max size: 460,800 bytes) located in an S3 bucket in the same region as the stack. For more information, go to [Template Anatomy](#) in the AWS CloudFormation User Guide.

Conditional: You must pass `TemplateURL` or `TemplateBody` . If both are passed, only `TemplateBody` is used.

Return type dict

Returns

Response Syntax

```
{  
    'Parameters': [  
        {  
            'ParameterKey': 'string',  
            'DefaultValue': 'string',  
            'NoEcho': True|False,  
            'Description': 'string'  
        },  
    ],  
    'Description': 'string',  
    'Capabilities': [  
        'CAPABILITY_IAM',  
    ],  
    'CapabilitiesReason': 'string'  
}
```

Response Structure

- (*dict*) –
The output for ValidateTemplate action.

– **Parameters** (*list*) –

A list of `TemplateParameter` structures.

* (*dict*) –

The `TemplateParameter` data type.

· **ParameterKey** (*string*) –

The name associated with the parameter.

· **DefaultValue** (*string*) –

The default value associated with the parameter.

· **NoEcho** (*boolean*) –

Flag indicating whether the parameter should be displayed as plain text in logs and UIs.

· **Description** (*string*) –

User defined description associated with the parameter.

– **Description** (*string*) –

The description found within the template.

– **Capabilities** (*list*) –

The capabilities found within the template. Currently, AWS CloudFormation supports only the `CAPABILITY_IAM` capability. If your template contains IAM resources, you must specify the `CAPABILITY_IAM` value for this parameter when you use the `CreateStack` or `UpdateStack` actions with your template; otherwise, those actions return an `InsufficientCapabilities` error.

* (*string*) –

– **CapabilitiesReason** (*string*) –

The list of resources that generated the values in the `Capabilities` response element.

Paginators

The available paginators are:

- `CloudFormation.Paginator.DescribeStackEvents`
- `CloudFormation.Paginator.DescribeStacks`
- `CloudFormation.Paginator.ListStackResources`
- `CloudFormation.Paginator.ListStacks`

class `CloudFormation.Paginator.DescribeStackEvents`

```
paginator = client.get_paginator('describe_stack_events')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `CloudFormation.Client.describe_stack_events()`.

Request Syntax

```
response_iterator = paginator.paginate(
    StackName='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **StackName** (*string*) – The name or the unique stack ID that is associated with the stack, which are not always interchangeable:
 - Running stacks: You can specify either the stack’s name or its unique stack ID.
 - Deleted stacks: You must specify the unique stack ID.

Default: There is no default value.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
    'StackEvents': [
        {
            'StackId': 'string',
            'EventId': 'string',
            'StackName': 'string',
            'LogicalResourceId': 'string',
            'PhysicalResourceId': 'string',
            'ResourceType': 'string',
            'Timestamp': datetime(2015, 1, 1),
            'ResourceStatus': 'CREATE_IN_PROGRESS' | 'CREATE_FAILED' | 'CREATE_COMPLETE',
            'ResourceStatusReason': 'string',
            'ResourceProperties': 'string'
        },
    ],
}
```

Response Structure

- (*dict*) –

The output for a `DescribeStackEvents` action.

- **StackEvents** (*list*) –

A list of `StackEvents` structures.

* (*dict*) –

The `StackEvent` data type.

- **StackId** (*string*) –

The unique ID name of the instance of the stack.

- **EventId** (*string*) –

The unique ID of this event.

- **StackName** (*string*) –

The name associated with a stack.

- **LogicalResourceId** (*string*) –

The logical name of the resource specified in the template.

- **PhysicalResourceId** (*string*) –

The name or unique identifier associated with the physical instance of the resource.

- **ResourceType** (*string*) –

Type of resource. (For more information, go to [AWS Resource Types Reference](#) in the AWS CloudFormation User Guide.)

- **Timestamp** (*datetime*) –

Time the status was updated.

- **ResourceStatus** (*string*) –

Current status of the resource.

- **ResourceStatusReason** (*string*) –

Success/failure message associated with the resource.

- **ResourceProperties** (*string*) –

BLOB of the properties used to create the resource.

class `CloudFormation.Paginator.DescribeStacks`

```
paginator = client.get_paginator('describe_stacks')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `CloudFormation.Client.describe_stacks()`.

Request Syntax

```
response_iterator = paginator.paginate(
    StackName='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **StackName** (*string*) – The name or the unique stack ID that is associated with the stack, which are not always interchangeable:

- Running stacks: You can specify either the stack’s name or its unique stack ID.
- Deleted stacks: You must specify the unique stack ID.

Default: There is no default value.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Stacks': [
    {
      'StackId': 'string',
      'StackName': 'string',
      'Description': 'string',
      'Parameters': [
        {
          'ParameterKey': 'string',
          'ParameterValue': 'string',
          'UsePreviousValue': True|False
        },
      ],
      'CreationTime': datetime(2015, 1, 1),
      'LastUpdatedTime': datetime(2015, 1, 1),
      'StackStatus': 'CREATE_IN_PROGRESS' | 'CREATE_FAILED' | 'CREATE_COMPLETE' |
      'StackStatusReason': 'string',
      'DisableRollback': True|False,
      'NotificationARNs': [
        'string',
      ],
      'TimeoutInMinutes': 123,
      'Capabilities': [
        'CAPABILITY_IAM',
      ],
      'Outputs': [
        {
          'OutputKey': 'string',
          'OutputValue': 'string',
          'Description': 'string'
        },
      ],
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        },
      ],
    },
  ],
}
```

```

    },
    ],
  },
],
}

```

Response Structure

- *(dict)* –

The output for a DescribeStacks action.

- **Stacks** (*list*) –

A list of stack structures.

- * *(dict)* –

The Stack data type.

- **StackId** (*string*) –

Unique identifier of the stack.

- **StackName** (*string*) –

The name associated with the stack.

- **Description** (*string*) –

User defined description associated with the stack.

- **Parameters** (*list*) –

A list of `Parameter` structures.

- *(dict)* –

The Parameter data type.

- **ParameterKey** (*string*) –

The key associated with the parameter. If you don't specify a key and value for a particular parameter, AWS CloudFormation uses the default value that is specified in your template.

- **ParameterValue** (*string*) –

The value associated with the parameter.

- **UsePreviousValue** (*boolean*) –

During a stack update, use the existing parameter value that the stack is using for a given parameter key. If you specify `true`, do not specify a parameter value.

- **CreationTime** (*datetime*) –

Time at which the stack was created.

- **LastUpdatedTime** (*datetime*) –

The time the stack was last updated. This field will only be returned if the stack has been updated at least once.

- **StackStatus** (*string*) –

Current status of the stack.

- **StackStatusReason** (*string*) –

Success/failure message associated with the stack status.

- **DisableRollback** (*boolean*) –

Boolean to enable or disable rollback on stack creation failures:

- `true` : disable rollback
- `false` : enable rollback
- **NotificationARNs** (*list*) –
SNS topic ARNs to which stack related events are published.
- (*string*) –
- **TimeoutInMinutes** (*integer*) –
The amount of time within which stack creation should complete.
- **Capabilities** (*list*) –
The capabilities allowed in the stack.
- (*string*) –
- **Outputs** (*list*) –
A list of output structures.
- (*dict*) –
The Output data type.
- **OutputKey** (*string*) –
The key associated with the output.
- **OutputValue** (*string*) –
The value associated with the output.
- **Description** (*string*) –
User defined description associated with the output.
- **Tags** (*list*) –
A list of `Tag`s that specify cost allocation information for the stack.
- (*dict*) –
The `Tag` type is used by `CreateStack` in the `Tags` parameter. It allows you to specify a key/value pair that can be used to store information related to cost allocation for an AWS CloudFormation stack.
- **Key** (*string*) –
Required . A string used to identify this tag. You can specify a maximum of 128 characters for a tag key. Tags owned by Amazon Web Services (AWS) have the reserved prefix: `aws:`.
- **Value** (*string*) –
Required . A string containing the value for this tag. You can specify a maximum of 256 characters for a tag value.

class CloudFormation.Paginator.**ListStackResources**

```
paginator = client.get_paginator('list_stack_resources')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `CloudFormation.Client.list_stack_resources()`.

Request Syntax

```

response_iterator = paginator.paginate(
    StackName='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)

```

Parameters

- **StackName** (*string*) – **[REQUIRED]**

The name or the unique stack ID that is associated with the stack, which are not always interchangeable:

- Running stacks: You can specify either the stack’s name or its unique stack ID.
- Deleted stacks: You must specify the unique stack ID.

Default: There is no default value.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```

{
    'StackResourceSummaries': [
        {
            'LogicalResourceId': 'string',
            'PhysicalResourceId': 'string',
            'ResourceType': 'string',
            'LastUpdatedTimestamp': datetime(2015, 1, 1),
            'ResourceStatus': 'CREATE_IN_PROGRESS' | 'CREATE_FAILED' | 'CREATE_COMPLETE',
            'ResourceStatusReason': 'string'
        },
    ],
}

```

Response Structure

- (*dict*) –

The output for a `ListStackResources` action.

- **StackResourceSummaries** (*list*) –

A list of `StackResourceSummary` structures.

* (*dict*) –

Contains high-level information about the specified stack resource.

- **LogicalResourceId** (*string*) –

The logical name of the resource specified in the template.

- **PhysicalResourceId** (*string*) –

The name or unique identifier that corresponds to a physical instance ID of the resource.

- **ResourceType** (*string*) –

Type of resource. (For more information, go to [AWS Resource Types Reference](#) in the AWS CloudFormation User Guide.)

- **LastUpdatedTimestamp** (*datetime*) –

Time the status was updated.

- **ResourceStatus** (*string*) –

Current status of the resource.

- **ResourceStatusReason** (*string*) –

Success/failure message associated with the resource.

class CloudFormation.Paginator.**ListStacks**

```
paginator = client.get_paginator('list_stacks')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `CloudFormation.Client.list_stacks()`.

Request Syntax

```
response_iterator = paginator.paginate(
    StackStatusFilter=[
        'CREATE_IN_PROGRESS' | 'CREATE_FAILED' | 'CREATE_COMPLETE' | 'ROLLBACK_IN_PROGRESS' | 'ROLLBACK_COMPLETE' | 'DELETE_IN_PROGRESS' | 'DELETE_FAILED' | 'DELETE_COMPLETE' | 'UPDATE_IN_PROGRESS' | 'UPDATE_FAILED' | 'UPDATE_COMPLETE' | 'UPDATE_COMPLETE_WITH_ROLLBACK',
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **StackStatusFilter** (*list*) – Stack status to use as a filter. Specify one or more stack status codes to list only stacks with the specified status codes. For a complete list of stack status codes, see the `StackStatus` parameter of the `Stack` data type.
 - (*string*) –
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –
The size of each page.
- **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'StackSummaries': [
    {
      'StackId': 'string',
      'StackName': 'string',
      'TemplateDescription': 'string',
      'CreationTime': datetime(2015, 1, 1),
      'LastUpdatedTime': datetime(2015, 1, 1),
      'DeletionTime': datetime(2015, 1, 1),
      'StackStatus': 'CREATE_IN_PROGRESS' | 'CREATE_FAILED' | 'CREATE_COMPLETE' |
      'StackStatusReason': 'string'
    },
  ],
}
```

Response Structure

- (*dict*) –

The output for `ListStacks` action.

- **StackSummaries** (*list*) –

A list of `StackSummary` structures containing information about the specified stacks.

- * (*dict*) –

The `StackSummary` Data Type

- **StackId** (*string*) –
Unique stack identifier.
- **StackName** (*string*) –
The name associated with the stack.
- **TemplateDescription** (*string*) –
The template description of the template used to create the stack.
- **CreationTime** (*datetime*) –
The time the stack was created.
- **LastUpdatedTime** (*datetime*) –
The time the stack was last updated. This field will only be returned if the stack has been updated at least once.
- **DeletionTime** (*datetime*) –
The time the stack was deleted.
- **StackStatus** (*string*) –
The current status of the stack.

- **StackStatusReason** (*string*) –
Success/Failure message associated with the stack status.

Service Resource

class CloudFormation.**ServiceResource**

A resource representing AWS CloudFormation:

```
import boto3

cloudformation = boto3.resource('cloudformation')
```

These are the resource's available actions:

- `create_stack()`

These are the resource's available sub-resources:

- `Event()`
- `Stack()`
- `StackResource()`
- `StackResourceSummary()`

These are the resource's available collections:

- `stacks`

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

create_stack (***kwargs*)

Creates a stack as specified in the template. After the call completes successfully, the stack creation starts. You can check the status of the stack via the DescribeStacks API.

Request Syntax

```
stack = cloudformation.create_stack(
    StackName='string',
    TemplateBody='string',
    TemplateURL='string',
    Parameters=[
        {
            'ParameterKey': 'string',
            'ParameterValue': 'string',
            'UsePreviousValue': True|False
        },
    ],
    DisableRollback=True|False,
    TimeoutInMinutes=123,
    NotificationARNs=[
        'string',
    ],
    Capabilities=[
        'CAPABILITY_IAM',
    ],
    OnFailure='DO_NOTHING'|'ROLLBACK'|'DELETE',
    StackPolicyBody='string',
    StackPolicyURL='string',
    Tags=[
        {
```



```

        'Key': 'string',
        'Value': 'string'
    },
]
)

```

Parameters

- **StackName** (*string*) – [REQUIRED]

The name that is associated with the stack. The name must be unique in the region in which you are creating the stack.

Note: A stack name can contain only alphanumeric characters (case sensitive) and hyphens. It must start with an alphabetic character and cannot be longer than 255 characters.

- **TemplateBody** (*string*) – Structure containing the template body with a minimum length of 1 byte and a maximum length of 51,200 bytes. For more information, go to [Template Anatomy](#) in the AWS CloudFormation User Guide.

Conditional: You must specify either the `TemplateBody` or the `TemplateURL` parameter, but not both.

- **TemplateURL** (*string*) – Location of file containing the template body. The URL must point to a template (max size: 460,800 bytes) located in an S3 bucket in the same region as the stack. For more information, go to the [Template Anatomy](#) in the AWS CloudFormation User Guide.

Conditional: You must specify either the `TemplateBody` or the `TemplateURL` parameter, but not both.

- **Parameters** (*list*) – A list of `Parameter` structures that specify input parameters for the stack.
 - (*dict*) –

The Parameter data type.

- * **ParameterKey** (*string*) –

The key associated with the parameter. If you don't specify a key and value for a particular parameter, AWS CloudFormation uses the default value that is specified in your template.

- * **ParameterValue** (*string*) –

The value associated with the parameter.

- * **UsePreviousValue** (*boolean*) –

During a stack update, use the existing parameter value that the stack is using for a given parameter key. If you specify `true`, do not specify a parameter value.

- **DisableRollback** (*boolean*) – Set to `true` to disable rollback of the stack if stack creation failed. You can specify either `DisableRollback` or `OnFailure`, but not both.

Default: `false`

- **TimeoutInMinutes** (*integer*) – The amount of time that can pass before the stack status becomes `CREATE_FAILED`; if `DisableRollback` is not set or is set to `false`, the stack will be rolled back.
- **NotificationARNs** (*list*) – The Simple Notification Service (SNS) topic ARNs to publish stack related events. You can find your SNS topic ARNs using the [SNS console](#) or your Command Line Interface (CLI).

– (*string*) –

- **Capabilities** (*list*) – A list of capabilities that you must specify before AWS CloudFormation can create or update certain stacks. Some stack templates might include resources that can affect permissions in your AWS account. For those stacks, you must explicitly acknowledge their capabilities by specifying this parameter.

Currently, the only valid value is `CAPABILITY_IAM`, which is required for the following resources: `AWS::IAM::AccessKey`, `AWS::IAM::Group`, `AWS::IAM::InstanceProfile`, `AWS::IAM::Policy`, `AWS::IAM::Role`, `AWS::IAM::User`, and `AWS::IAM::UserToGroupAddition`. If your stack template contains these resources, we recommend that you review any permissions associated with them. If you don't specify this parameter, this action returns an `InsufficientCapabilities` error.

– (*string*) –

- **OnFailure** (*string*) – Determines what action will be taken if stack creation fails. This must be one of: `DO_NOTHING`, `ROLLBACK`, or `DELETE`. You can specify either `OnFailure` or `DisableRollback`, but not both.

Default: `ROLLBACK`

- **StackPolicyBody** (*string*) – Structure containing the stack policy body. For more information, go to [Prevent Updates to Stack Resources](#) in the AWS CloudFormation User Guide. You can specify either the `StackPolicyBody` or the `StackPolicyURL` parameter, but not both.
- **StackPolicyURL** (*string*) – Location of a file containing the stack policy. The URL must point to a policy (max size: 16KB) located in an S3 bucket in the same region as the stack. You can specify either the `StackPolicyBody` or the `StackPolicyURL` parameter, but not both.
- **Tags** (*list*) – A set of user-defined `Tags` to associate with this stack, represented by key/value pairs. Tags defined for the stack are propagated to EC2 resources that are created as part of the stack. A maximum number of 10 tags can be specified.

– (*dict*) –

The `Tag` type is used by `CreateStack` in the `Tags` parameter. It allows you to specify a key/value pair that can be used to store information related to cost allocation for an AWS CloudFormation stack.

* **Key** (*string*) –

Required. A string used to identify this tag. You can specify a maximum of 128 characters for a tag key. Tags owned by Amazon Web Services (AWS) have the reserved prefix: `aws:`.

* **Value** (*string*) –

Required. A string containing the value for this tag. You can specify a maximum of 256 characters for a tag value.

Return type `cloudformation.Stack`

Returns Stack resource

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Event (*id*)

Creates a Event resource.:

```
event = cloudformation.Event('id')
```

Parameters `id` (*string*) – The Event’s id identifier. This **must** be set.

Return type `CloudFormation.Event`

Returns A Event resource

Stack (*name*)

Creates a Stack resource.:

```
stack = cloudformation.Stack('name')
```

Parameters `name` (*string*) – The Stack’s name identifier. This **must** be set.

Return type `CloudFormation.Stack`

Returns A Stack resource

StackResource (*stack_name, logical_id*)

Creates a StackResource resource.:

```
stack_resource = cloudformation.StackResource('stack_name', 'logical_id')
```

Parameters

- **stack_name** (*string*) – The StackResource’s stack_name identifier. This **must** be set.
- **logical_id** (*string*) – The StackResource’s logical_id identifier. This **must** be set.

Return type `CloudFormation.StackResource`

Returns A StackResource resource

StackResourceSummary (*stack_name, logical_id*)

Creates a StackResourceSummary resource.:

```
stack_resource_summary = cloudformation.StackResourceSummary('stack_name', 'logical_id')
```

Parameters

- **stack_name** (*string*) – The StackResourceSummary’s stack_name identifier. This **must** be set.
- **logical_id** (*string*) – The StackResourceSummary’s logical_id identifier. This **must** be set.

Return type `CloudFormation.StackResourceSummary`

Returns A StackResourceSummary resource

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

stacks

all ()

Creates an iterable of all Stack resources in the collection.

Request Syntax

```
stack_iterator = cloudformation.stacks.all()
```

Return type `list(cloudformation.Stack)`

Returns A list of Stack resources

filter (***kwargs*)

Creates an iterable of all Stack resources in the collection filtered by kwargs passed to method.

Request Syntax

```
stack_iterator = cloudformation.stacks.filter(  
    StackName='string',  
    NextToken='string'  
)
```

Parameters

- **StackName** (*string*) – The name or the unique stack ID that is associated with the stack, which are not always interchangeable:
 - Running stacks: You can specify either the stack’s name or its unique stack ID.
 - Deleted stacks: You must specify the unique stack ID.Default: There is no default value.
- **NextToken** (*string*) – String that identifies the start of the next list of stacks, if there is one.

Return type list(`cloudformation.Stack`)

Returns A list of Stack resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Stack resources in the collection.

Request Syntax

```
stack_iterator = cloudformation.stacks.limit(  
    count=123  
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type list(`cloudformation.Stack`)

Returns A list of Stack resources

page_size (***kwargs*)

Creates an iterable of all Stack resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
stack_iterator = cloudformation.stacks.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type list(`cloudformation.Stack`)

Returns A list of Stack resources

Event

class `CloudFormation.Event` (*id*)

A resource representing an AWS CloudFormation Event:

```
import boto3

cloudformation = boto3.resource('cloudformation')
event = cloudformation.Event('id')
```

Parameters `id` (*string*) – The Event’s id identifier. This **must** be set.

These are the resource’s available identifiers:

- `id`

These are the resource’s available attributes:

- `event_id`
- `logical_resource_id`
- `physical_resource_id`
- `resource_properties`
- `resource_status`
- `resource_status_reason`
- `resource_type`
- `stack_id`
- `stack_name`
- `timestamp`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

id

(*string*) The Event’s id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

event_id

(*string*)

The unique ID of this event.

logical_resource_id

(*string*)

The logical name of the resource specified in the template.

physical_resource_id

(*string*)

The name or unique identifier associated with the physical instance of the resource.

resource_properties

(*string*)

BLOB of the properties used to create the resource.

resource_status

(*string*)

Current status of the resource.

resource_status_reason

(*string*)

Success/failure message associated with the resource.

resource_type*(string)*

Type of resource. (For more information, go to [AWS Resource Types Reference](#) in the AWS CloudFormation User Guide.)

stack_id*(string)*

The unique ID name of the instance of the stack.

stack_name*(string)*

The name associated with a stack.

timestamp*(datetime)*

Time the status was updated.

Stack

class CloudFormation.**Stack** (*name*)

A resource representing an AWS CloudFormation Stack:

```
import boto3
```

```
cloudformation = boto3.resource('cloudformation')
stack = cloudformation.Stack('name')
```

Parameters *name* (*string*) – The Stack’s name identifier. This **must** be set.

These are the resource’s available identifiers:

- *name*

These are the resource’s available attributes:

- *capabilities*
- *creation_time*
- *description*
- *disable_rollback*
- *last_updated_time*
- *notification_arns*
- *outputs*
- *parameters*
- *stack_id*
- *stack_name*
- *stack_status*
- *stack_status_reason*
- *tags*
- *timeout_in_minutes*

These are the resource’s available actions:

- *cancel_update()*
- *delete()*
- *load()*
- *reload()*
- *update()*

These are the resource’s available sub-resources:

- `Resource()`

These are the resource's available collections:

- `events`
- `resource_summaries`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

name

(*string*) The Stack's name identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

capabilities

(*list*)

The capabilities allowed in the stack.

creation_time

(*datetime*)

Time at which the stack was created.

description

(*string*)

User defined description associated with the stack.

disable_rollback

(*boolean*)

Boolean to enable or disable rollback on stack creation failures:

- `true`: disable rollback
- `false`: enable rollback

last_updated_time

(*datetime*)

The time the stack was last updated. This field will only be returned if the stack has been updated at least once.

notification_arns

(*list*)

SNS topic ARNs to which stack related events are published.

outputs

(*list*)

A list of output structures.

parameters

(*list*)

A list of `Parameter` structures.

stack_id

(*string*)

Unique identifier of the stack.

stack_name*(string)*

The name associated with the stack.

stack_status*(string)*

Current status of the stack.

stack_status_reason*(string)*

Success/failure message associated with the stack status.

tags*(list)*

A list of `Tag s` that specify cost allocation information for the stack.

timeout_in_minutes*(integer)*

The amount of time within which stack creation should complete.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

cancel_update()

Cancels an update on the specified stack. If the call completes successfully, the stack will roll back the update and revert to the previous stack configuration.

Note: Only stacks that are in the `UPDATE_IN_PROGRESS` state can be canceled.

Request Syntax

```
response = stack.cancel_update()
```

Returns None

delete()

Deletes a specified stack. Once the call completes successfully, stack deletion starts. Deleted stacks do not show up in the `DescribeStacks` API if the deletion has been completed successfully.

Request Syntax

```
response = stack.delete()
```

Returns None

load()

Calls `cloudformation.Client.describe_stacks()` to update the attributes of the Stack resource

Request Syntax

```
stack.load()
```

Returns None

reload()

Calls `cloudformation.Client.describe_stacks()` to update the attributes of the Stack resource

Request Syntax

```
stack.reload()
```

Returns None

update(kwargs)**

Updates a stack as specified in the template. After the call completes successfully, the stack update starts. You can check the status of the stack via the `DescribeStacks` action.

To get a copy of the template for an existing stack, you can use the `GetTemplate` action.

Tags that were associated with this stack during creation time will still be associated with the stack after an `UpdateStack` operation.

For more information about creating an update template, updating a stack, and monitoring the progress of the update, see [Updating a Stack](#).

Request Syntax

```
response = stack.update(
    TemplateBody='string',
    TemplateURL='string',
    UsePreviousTemplate=True|False,
    StackPolicyDuringUpdateBody='string',
    StackPolicyDuringUpdateURL='string',
    Parameters=[
        {
            'ParameterKey': 'string',
            'ParameterValue': 'string',
            'UsePreviousValue': True|False
        },
    ],
    Capabilities=[
        'CAPABILITY_IAM',
    ],
    StackPolicyBody='string',
    StackPolicyURL='string',
    NotificationARNs=[
        'string',
    ]
)
```

Parameters

- **TemplateBody** (*string*) – Structure containing the template body with a minimum length of 1 byte and a maximum length of 51,200 bytes. (For more information, go to [Template Anatomy](#) in the AWS CloudFormation User Guide.)

Conditional: You must specify either the `TemplateBody` or the `TemplateURL` parameter, but not both.

- **TemplateURL** (*string*) – Location of file containing the template body. The URL must point to a template located in an S3 bucket in the same region as the stack. For more information, go to [Template Anatomy](#) in the AWS CloudFormation User Guide.

Conditional: You must specify either the `TemplateBody` or the `TemplateURL` parameter, but not both.

- **UsePreviousTemplate** (*boolean*) – Reuse the existing template that is associated with the stack that you are updating.
- **StackPolicyDuringUpdateBody** (*string*) – Structure containing the temporary overriding stack policy body. You can specify either the `StackPolicyDuringUpdateBody` or the `StackPolicyDuringUpdateURL` parameter, but not both.

If you want to update protected resources, specify a temporary overriding stack policy during this update. If you do not specify a stack policy, the current policy that is associated with the stack will be used.

- **StackPolicyDuringUpdateURL** (*string*) – Location of a file containing the temporary overriding stack policy. The URL must point to a policy (max size: 16KB) located in an S3 bucket in the same region as the stack. You can specify either the `StackPolicyDuringUpdateBody` or the `StackPolicyDuringUpdateURL` parameter, but not both.

If you want to update protected resources, specify a temporary overriding stack policy during this update. If you do not specify a stack policy, the current policy that is associated with the stack will be used.

- **Parameters** (*list*) – A list of `Parameter` structures that specify input parameters for the stack. For more information, see the [Parameter](#) data type.
 - (*dict*) –

The `Parameter` data type.

- * **ParameterKey** (*string*) –

The key associated with the parameter. If you don't specify a key and value for a particular parameter, AWS CloudFormation uses the default value that is specified in your template.

- * **ParameterValue** (*string*) –

The value associated with the parameter.

- * **UsePreviousValue** (*boolean*) –

During a stack update, use the existing parameter value that the stack is using for a given parameter key. If you specify `true`, do not specify a parameter value.

- **Capabilities** (*list*) – A list of capabilities that you must specify before AWS CloudFormation can create or update certain stacks. Some stack templates might include resources that can affect permissions in your AWS account. For those stacks, you must explicitly acknowledge their capabilities by specifying this parameter. Currently, the only valid value is `CAPABILITY_IAM`, which is required for the following resources: `AWS::IAM::AccessKey`, `AWS::IAM::Group`, `AWS::IAM::InstanceProfile`, `AWS::IAM::Policy`, `AWS::IAM::Role`, `AWS::IAM::User`, and `AWS::IAM::UserToGroupAddition`. If your stack template contains these resources, we recommend that you review any permissions associated with them. If you don't specify this parameter, this action returns an `InsufficientCapabilities` error.
 - (*string*) –

- **StackPolicyBody** (*string*) – Structure containing a new stack policy body. You can specify either the `StackPolicyBody` or the `StackPolicyURL` parameter, but not both.

You might update the stack policy, for example, in order to protect a new resource that you created during a stack update. If you do not specify a stack policy, the

current policy that is associated with the stack is unchanged.

- **StackPolicyURL** (*string*) – Location of a file containing the updated stack policy. The URL must point to a policy (max size: 16KB) located in an S3 bucket in the same region as the stack. You can specify either the `StackPolicyBody` or the `StackPolicyURL` parameter, but not both.

You might update the stack policy, for example, in order to protect a new resource that you created during a stack update. If you do not specify a stack policy, the current policy that is associated with the stack is unchanged.

- **NotificationARNs** (*list*) – Update the ARNs for the Amazon SNS topics that are associated with the stack.
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'StackId': 'string'
}
```

Response Structure

- (*dict*) –

The output for a `UpdateStack` action.

- **StackId** (*string*) –

Unique identifier of the stack.

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Resource (*logical_id*)

Creates a `StackResource` resource.:

```
stack_resource = stack.Resource('logical_id')
```

Parameters **logical_id** (*string*) – The Resource's `logical_id` identifier. This **must** be set.

Return type `CloudFormation.StackResource`

Returns A `StackResource` resource

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

events

all()

Creates an iterable of all Event resources in the collection.

Request Syntax

```
event_iterator = stack.events.all()
```

Return type `list(CloudFormation.Event)`

Returns A list of Event resources

filter (***kwargs*)

Creates an iterable of all Event resources in the collection filtered by kwargs passed to method.

Request Syntax

```
event_iterator = stack.events.filter(  
    NextToken='string'  
)
```

Parameters **NextToken** (*string*) – String that identifies the start of the next list of events, if there is one.

Default: There is no default value.

Return type list(`cloudformation.Event`)

Returns A list of Event resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Event resources in the collection.

Request Syntax

```
event_iterator = stack.events.limit(  
    count=123  
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type list(`cloudformation.Event`)

Returns A list of Event resources

page_size (***kwargs*)

Creates an iterable of all Event resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
event_iterator = stack.events.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type list(`cloudformation.Event`)

Returns A list of Event resources

resource_summaries

all ()

Creates an iterable of all StackResourceSummary resources in the collection.

Request Syntax

```
stack_resource_summary_iterator = stack.resource_summaries.all()
```

Return type list(`cloudformation.StackResourceSummary`)

Returns A list of StackResourceSummary resources

filter (***kwargs*)

Creates an iterable of all StackResourceSummary resources in the collection filtered by kwargs passed to method.

Request Syntax

```
stack_resource_summary_iterator = stack.resource_summaries.filter(
    NextToken='string'
)
```

Parameters **NextToken** (*string*) – String that identifies the start of the next list of stack resource summaries, if there is one.

Default: There is no default value.

Return type list(`cloudformation.StackResourceSummary`)

Returns A list of `StackResourceSummary` resources

limit (***kwargs*)

Creates an iterable up to a specified amount of `StackResourceSummary` resources in the collection.

Request Syntax

```
stack_resource_summary_iterator = stack.resource_summaries.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type list(`cloudformation.StackResourceSummary`)

Returns A list of `StackResourceSummary` resources

page_size (***kwargs*)

Creates an iterable of all `StackResourceSummary` resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
stack_resource_summary_iterator = stack.resource_summaries.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type list(`cloudformation.StackResourceSummary`)

Returns A list of `StackResourceSummary` resources

StackResource

class `CloudFormation.StackResource` (*stack_name*, *logical_id*)

A resource representing an AWS CloudFormation `StackResource`:

```
import boto3
```

```
cloudformation = boto3.resource('cloudformation')
stack_resource = cloudformation.StackResource('stack_name', 'logical_id')
```

Parameters

- **stack_name** (*string*) – The `StackResource`'s `stack_name` identifier. This **must** be set.
- **logical_id** (*string*) – The `StackResource`'s `logical_id` identifier. This **must** be set.

These are the resource's available identifiers:

- `stack_name`
- `logical_id`

These are the resource's available attributes:

- `description`
- `last_updated_timestamp`
- `logical_resource_id`
- `metadata`
- `physical_resource_id`
- `resource_status`
- `resource_status_reason`
- `resource_type`
- `stack_id`

These are the resource's available sub-resources:

- `Stack()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

stack_name

(*string*) The StackResource's `stack_name` identifier. This **must** be set.

logical_id

(*string*) The StackResource's `logical_id` identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

description

(*string*)

User defined description associated with the resource.

last_updated_timestamp

(*datetime*)

Time the status was updated.

logical_resource_id

(*string*)

The logical name of the resource specified in the template.

metadata

(*string*)

The JSON format content of the `Metadata` attribute declared for the resource. For more information, see [Metadata Attribute](#) in the AWS CloudFormation User Guide.

physical_resource_id

(*string*)

The name or unique identifier that corresponds to a physical instance ID of a resource supported by AWS CloudFormation.

resource_status

(*string*)

Current status of the resource.

resource_status_reason

(*string*)

Success/failure message associated with the resource.

resource_type
(*string*)

Type of resource. ((For more information, go to [AWS Resource Types Reference](#) in the AWS CloudFormation User Guide.)

stack_id
(*string*)

Unique identifier of the stack.

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Stack()
Creates a Stack resource.:

```
stack = stack_resource.Stack()
```

Return type `CloudFormation.Stack`
Returns A Stack resource

StackResourceSummary

class `CloudFormation.StackResourceSummary`(*stack_name*, *logical_id*)
A resource representing an AWS CloudFormation StackResourceSummary:

```
import boto3
```

```
cloudformation = boto3.resource('cloudformation')
stack_resource_summary = cloudformation.StackResourceSummary('stack_name', 'logical_id')
```

Parameters

- **stack_name** (*string*) – The StackResourceSummary's stack_name identifier. This **must** be set.
- **logical_id** (*string*) – The StackResourceSummary's logical_id identifier. This **must** be set.

These are the resource's available identifiers:

- `stack_name`
- `logical_id`

These are the resource's available attributes:

- `last_updated_timestamp`
- `logical_resource_id`
- `physical_resource_id`
- `resource_status`
- `resource_status_reason`
- `resource_type`

These are the resource's available sub-resources:

- `Resource()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

stack_name

(*string*) The StackResourceSummary's stack_name identifier. This **must** be set.

logical_id

(*string*) The StackResourceSummary's logical_id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

last_updated_timestamp

(*datetime*)

Time the status was updated.

logical_resource_id

(*string*)

The logical name of the resource specified in the template.

physical_resource_id

(*string*)

The name or unique identifier that corresponds to a physical instance ID of the resource.

resource_status

(*string*)

Current status of the resource.

resource_status_reason

(*string*)

Success/failure message associated with the resource.

resource_type

(*string*)

Type of resource. (For more information, go to [AWS Resource Types Reference](#) in the AWS CloudFormation User Guide.)

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Resource()

Creates a StackResource resource.:

```
stack_resource = stack_resource_summary.Resource()
```

Return type `CloudFormation.StackResource`

Returns A StackResource resource

CloudFront

Table of Contents

- CloudFront
 - Client
 - Paginators
 - Waiters

Client**class CloudFront.Client**

A low-level client representing Amazon CloudFront:

```
import boto3
```

```
client = boto3.client('cloudfront')
```

These are the available methods:

- `can_paginate()`
- `create_cloud_front_origin_access_identity()`
- `create_distribution()`
- `create_invalidation()`
- `create_streaming_distribution()`
- `delete_cloud_front_origin_access_identity()`
- `delete_distribution()`
- `delete_streaming_distribution()`
- `generate_presigned_url()`
- `get_cloud_front_origin_access_identity()`
- `get_cloud_front_origin_access_identity_config()`
- `get_distribution()`
- `get_distribution_config()`
- `get_invalidation()`
- `get_paginator()`
- `get_streaming_distribution()`
- `get_streaming_distribution_config()`
- `get_waiter()`
- `list_cloud_front_origin_access_identities()`
- `list_distributions()`
- `list_invalidations()`
- `list_streaming_distributions()`
- `update_cloud_front_origin_access_identity()`
- `update_distribution()`
- `update_streaming_distribution()`

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

create_cloud_front_origin_access_identity (**kwargs)

Create a new origin access identity.

Request Syntax

```
response = client.create_cloud_front_origin_access_identity(
    CloudFrontOriginAccessIdentityConfig={
        'CallerReference': 'string',
        'Comment': 'string'
    }
)
```

Parameters **CloudFrontOriginAccessIdentityConfig** (*dict*) – **[REQUIRED]** The origin access identity’s configuration information.

- **CallerReference** (*string*) – **[REQUIRED]** A unique number that ensures the request can’t be replayed. If the CallerReference is new (no matter the content of the CloudFrontOriginAccessIdentityConfig object), a new origin access identity is created. If the CallerReference is a value you already sent in a previous request to create an identity, and the content of the CloudFrontOriginAccessIdentityConfig is identical to the original request (ignoring white space), the response includes the same information returned to the original request. If the CallerReference is a value you already sent in a previous request to create an identity but the content of the CloudFrontOriginAccessIdentityConfig is different from the original request, CloudFront returns a CloudFrontOriginAccessIdentityAlreadyExists error.
- **Comment** (*string*) – **[REQUIRED]** Any comments you want to include about the origin access identity.

Return type dict

Returns**Response Syntax**

```
{
    'CloudFrontOriginAccessIdentity': {
        'Id': 'string',
        'S3CanonicalUserId': 'string',
        'CloudFrontOriginAccessIdentityConfig': {
            'CallerReference': 'string',
            'Comment': 'string'
        }
    },
    'Location': 'string',
    'ETag': 'string'
}
```

Response Structure

- (*dict*) – The returned result of the corresponding request.
 - **CloudFrontOriginAccessIdentity** (*dict*) – The origin access identity’s information.
 - * **Id** (*string*) – The ID for the origin access identity. For example: E74FTE3AJFJ256A.
 - * **S3CanonicalUserId** (*string*) – The Amazon S3 canonical user ID for the origin access identity, which you use when giving the origin access identity read permission to an object in Amazon S3.
 - * **CloudFrontOriginAccessIdentityConfig** (*dict*) – The current configuration information for the identity.

- **CallerReference** (*string*) – A unique number that ensures the request can't be replayed. If the CallerReference is new (no matter the content of the CloudFrontOriginAccessIdentityConfig object), a new origin access identity is created. If the CallerReference is a value you already sent in a previous request to create an identity, and the content of the CloudFrontOriginAccessIdentityConfig is identical to the original request (ignoring white space), the response includes the same information returned to the original request. If the CallerReference is a value you already sent in a previous request to create an identity but the content of the CloudFrontOriginAccessIdentityConfig is different from the original request, CloudFront returns a CloudFrontOriginAccessIdentityAlreadyExists error.
- **Comment** (*string*) – Any comments you want to include about the origin access identity.
- **Location** (*string*) – The fully qualified URI of the new origin access identity just created. For example: <https://cloudfront.amazonaws.com/2010-11-01/origin-access-identity/cloudfront/E74FTE3AJFJ256A>.
- **ETag** (*string*) – The current version of the origin access identity created.

create_distribution (**kwargs)

Create a new distribution.

Request Syntax

```
response = client.create_distribution(
    DistributionConfig={
        'CallerReference': 'string',
        'Aliases': {
            'Quantity': 123,
            'Items': [
                'string',
            ]
        },
        'DefaultRootObject': 'string',
        'Origins': {
            'Quantity': 123,
            'Items': [
                {
                    'Id': 'string',
                    'DomainName': 'string',
                    'OriginPath': 'string',
                    'S3OriginConfig': {
                        'OriginAccessIdentity': 'string'
                    },
                    'CustomOriginConfig': {
                        'HTTPPort': 123,
                        'HTTPSPort': 123,
                        'OriginProtocolPolicy': 'http-only'|'match-viewer'
                    }
                }
            ]
        },
        'DefaultCacheBehavior': {
            'TargetOriginId': 'string',
            'ForwardedValues': {
```

```
    'QueryString': True|False,
    'Cookies': {
        'Forward': 'none'|'whitelist'|'all',
        'WhitelistedNames': {
            'Quantity': 123,
            'Items': [
                'string',
            ]
        }
    },
    'Headers': {
        'Quantity': 123,
        'Items': [
            'string',
        ]
    }
},
'TrustedSigners': {
    'Enabled': True|False,
    'Quantity': 123,
    'Items': [
        'string',
    ]
},
'ViewerProtocolPolicy': 'allow-all'|'https-only'|'redirect-to-https',
'MinTTL': 123,
'AllowedMethods': {
    'Quantity': 123,
    'Items': [
        'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
    ],
    'CachedMethods': {
        'Quantity': 123,
        'Items': [
            'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
        ]
    }
},
'SmoothStreaming': True|False,
'DefaultTTL': 123,
'MaxTTL': 123
},
'CacheBehaviors': {
    'Quantity': 123,
    'Items': [
        {
            'PathPattern': 'string',
            'TargetOriginId': 'string',
            'ForwardedValues': {
                'QueryString': True|False,
                'Cookies': {
                    'Forward': 'none'|'whitelist'|'all',
                    'WhitelistedNames': {
                        'Quantity': 123,
                        'Items': [
                            'string',
                        ]
                    }
                }
            }
        }
    ]
}
```

```

    },
    'Headers': {
        'Quantity': 123,
        'Items': [
            'string',
        ]
    }
},
'TrustedSigners': {
    'Enabled': True|False,
    'Quantity': 123,
    'Items': [
        'string',
    ]
},
'ViewerProtocolPolicy': 'allow-all'|'https-only'|'redirect-to-https',
'MinTTL': 123,
'AllowedMethods': {
    'Quantity': 123,
    'Items': [
        'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
    ],
    'CachedMethods': {
        'Quantity': 123,
        'Items': [
            'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
        ]
    }
},
'SmoothStreaming': True|False,
'DefaultTTL': 123,
'MaxTTL': 123
},
]
},
'CustomErrorResponses': {
    'Quantity': 123,
    'Items': [
        {
            'ErrorCode': 123,
            'ResponsePagePath': 'string',
            'ResponseCode': 'string',
            'ErrorCachingMinTTL': 123
        }
    ],
},
'Comment': 'string',
'Logging': {
    'Enabled': True|False,
    'IncludeCookies': True|False,
    'Bucket': 'string',
    'Prefix': 'string'
},
'PriceClass': 'PriceClass_100'|'PriceClass_200'|'PriceClass_All',
'Enabled': True|False,
'ViewerCertificate': {
    'IAMCertificateId': 'string',
    'CloudFrontDefaultCertificate': True|False,

```

```
        'SSLSupportMethod': 'sni-only'|'vip',
        'MinimumProtocolVersion': 'SSLv3'|'TLSv1'
    },
    'Restrictions': {
        'GeoRestriction': {
            'RestrictionType': 'blacklist'|'whitelist'|'none',
            'Quantity': 123,
            'Items': [
                'string',
            ]
        }
    }
}
```

Parameters **DistributionConfig** (*dict*) – **[REQUIRED]** The distribution’s configuration information.

- **CallerReference** (*string*) – **[REQUIRED]** A unique number that ensures the request can’t be replayed. If the CallerReference is new (no matter the content of the DistributionConfig object), a new distribution is created. If the CallerReference is a value you already sent in a previous request to create a distribution, and the content of the DistributionConfig is identical to the original request (ignoring white space), the response includes the same information returned to the original request. If the CallerReference is a value you already sent in a previous request to create a distribution but the content of the DistributionConfig is different from the original request, CloudFront returns a DistributionAlreadyExists error.
- **Aliases** (*dict*) – A complex type that contains information about CNAMEs (alternate domain names), if any, for this distribution.
 - **Quantity** (*integer*) – **[REQUIRED]** The number of CNAMEs, if any, for this distribution.
 - **Items** (*list*) – Optional: A complex type that contains CNAME elements, if any, for this distribution. If Quantity is 0, you can omit Items.
 - * (*string*) –
- **DefaultRootObject** (*string*) – The object that you want CloudFront to return (for example, index.html) when an end user requests the root URL for your distribution (<http://www.example.com>) instead of an object in your distribution (<http://www.example.com/index.html>). Specifying a default root object avoids exposing the contents of your distribution. If you don’t want to specify a default root object when you create a distribution, include an empty DefaultRootObject element. To delete the default root object from an existing distribution, update the distribution configuration and include an empty DefaultRootObject element. To replace the default root object, update the distribution configuration and specify the new object.
- **Origins** (*dict*) – **[REQUIRED]** A complex type that contains information about origins for this distribution.
 - **Quantity** (*integer*) – **[REQUIRED]** The number of origins for this distribution.
 - **Items** (*list*) – A complex type that contains origins for this distribution.
 - * (*dict*) – A complex type that describes the Amazon S3 bucket or the HTTP server (for example, a web server) from which CloudFront gets your files. You must create at least one origin.
 - **Id** (*string*) – **[REQUIRED]** A unique identifier for the origin. The value of Id must be unique within the distribution. You use the value of Id when you create a cache behavior. The Id identifies the origin that CloudFront routes a request to when

- the request matches the path pattern for that cache behavior.
- **DomainName** (*string*) – **[REQUIRED]** Amazon S3 origins: The DNS name of the Amazon S3 bucket from which you want CloudFront to get objects for this origin, for example, myawsbucket.s3.amazonaws.com. Custom origins: The DNS domain name for the HTTP server from which you want CloudFront to get objects for this origin, for example, www.example.com.
 - **OriginPath** (*string*) – An optional element that causes CloudFront to request your content from a directory in your Amazon S3 bucket or your custom origin. When you include the OriginPath element, specify the directory name, beginning with a /. CloudFront appends the directory name to the value of DomainName.
 - **S3OriginConfig** (*dict*) – A complex type that contains information about the Amazon S3 origin. If the origin is a custom origin, use the CustomOriginConfig element instead.
 - **OriginAccessIdentity** (*string*) – **[REQUIRED]** The CloudFront origin access identity to associate with the origin. Use an origin access identity to configure the origin so that end users can only access objects in an Amazon S3 bucket through CloudFront. If you want end users to be able to access objects using either the CloudFront URL or the Amazon S3 URL, specify an empty OriginAccessIdentity element. To delete the origin access identity from an existing distribution, update the distribution configuration and include an empty OriginAccessIdentity element. To replace the origin access identity, update the distribution configuration and specify the new origin access identity. Use the format origin-access-identity/cloudfront/Id where Id is the value that CloudFront returned in the Id element when you created the origin access identity.
 - **CustomOriginConfig** (*dict*) – A complex type that contains information about a custom origin. If the origin is an Amazon S3 bucket, use the S3OriginConfig element instead.
 - **HTTPPort** (*integer*) – **[REQUIRED]** The HTTP port the custom origin listens on.
 - **HTTPSPort** (*integer*) – **[REQUIRED]** The HTTPS port the custom origin listens on.
 - **OriginProtocolPolicy** (*string*) – **[REQUIRED]** The origin protocol policy to apply to your origin.
 - **DefaultCacheBehavior** (*dict*) – **[REQUIRED]** A complex type that describes the default cache behavior if you do not specify a CacheBehavior element or if files don't match any of the values of PathPattern in CacheBehavior elements. You must create exactly one default cache behavior.
 - **TargetOriginId** (*string*) – **[REQUIRED]** The value of ID for the origin that you want CloudFront to route requests to when a request matches the path pattern either for a cache behavior or for the default cache behavior.
 - **ForwardedValues** (*dict*) – **[REQUIRED]** A complex type that specifies how CloudFront handles query strings, cookies and headers.
 - * **QueryString** (*boolean*) – **[REQUIRED]** Indicates whether you want CloudFront to forward query strings to the origin that is associated with this cache behavior. If so, specify true; if not, specify false.

- * **Cookies** (*dict*) – **[REQUIRED]** A complex type that specifies how CloudFront handles cookies.
 - **Forward** (*string*) – **[REQUIRED]** Use this element to specify whether you want CloudFront to forward cookies to the origin that is associated with this cache behavior. You can specify all, none or whitelist. If you choose All, CloudFront forwards all cookies regardless of how many your application uses.
 - **WhitelistedNames** (*dict*) – A complex type that specifies the whitelisted cookies, if any, that you want CloudFront to forward to your origin that is associated with this cache behavior.
 - **Quantity** (*integer*) – **[REQUIRED]** The number of whitelisted cookies for this cache behavior.
 - **Items** (*list*) – Optional: A complex type that contains whitelisted cookies for this cache behavior. If Quantity is 0, you can omit Items.
 - (*string*) –
- * **Headers** (*dict*) – A complex type that specifies the Headers, if any, that you want CloudFront to vary upon for this cache behavior.
 - **Quantity** (*integer*) – **[REQUIRED]** The number of different headers that you want CloudFront to forward to the origin and to vary on for this cache behavior. The maximum number of headers that you can specify by name is 10. If you want CloudFront to forward all headers to the origin and vary on all of them, specify 1 for Quantity and * for Name. If you don't want CloudFront to forward any additional headers to the origin or to vary on any headers, specify 0 for Quantity and omit Items.
 - **Items** (*list*) – Optional: A complex type that contains a Name element for each header that you want CloudFront to forward to the origin and to vary on for this cache behavior. If Quantity is 0, omit Items.
 - (*string*) –
- **TrustedSigners** (*dict*) – **[REQUIRED]** A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the PathPattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don't want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more trusted signers, change Enabled to true (if it's currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.
 - * **Enabled** (*boolean*) – **[REQUIRED]** Specifies whether you want to require end users to use signed URLs to access the files specified by PathPattern and TargetOriginId.
 - * **Quantity** (*integer*) – **[REQUIRED]** The number of trusted signers for this cache behavior.
 - * **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit Items.
 - (*string*) –
- **ViewerProtocolPolicy** (*string*) – **[REQUIRED]** Use this element to spec-

ify the protocol that users can use to access the files in the origin specified by `TargetOriginId` when a request matches the path pattern in `PathPattern`. If you want CloudFront to allow end users to use any available protocol, specify `allow-all`. If you want CloudFront to require HTTPS, specify `https`. If you want CloudFront to respond to an HTTP request with an HTTP status code of 301 (Moved Permanently) and the HTTPS URL, specify `redirect-to-https`. The viewer then resubmits the request using the HTTPS URL.

- **MinTTL** (*integer*) – **[REQUIRED]** The minimum amount of time that you want objects to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **AllowedMethods** (*dict*) – A complex type that controls which HTTP methods CloudFront processes and forwards to your Amazon S3 bucket or your custom origin. There are three choices:
 - CloudFront forwards only GET and HEAD requests.
 - CloudFront forwards only GET, HEAD and OPTIONS requests.
 - CloudFront forwards GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests. If you pick the third choice, you may need to restrict access to your Amazon S3 bucket or to your custom origin so users can't perform operations that you don't want them to. For example, you may not want users to have permission to delete objects from your origin.
- * **Quantity** (*integer*) – **[REQUIRED]** The number of HTTP methods that you want CloudFront to forward to your origin. Valid values are 2 (for GET and HEAD requests), 3 (for GET, HEAD and OPTIONS requests) and 7 (for GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests).
 - * **Items** (*list*) – **[REQUIRED]** A complex type that contains the HTTP methods that you want CloudFront to process and forward to your origin.
 - (*string*) –
 - * **CachedMethods** (*dict*) – A complex type that controls whether CloudFront caches the response to requests using the specified HTTP methods. There are two choices:
 - CloudFront caches responses to GET and HEAD requests.
 - CloudFront caches responses to GET, HEAD, and OPTIONS requests. If you pick the second choice for your S3 Origin, you may need to forward `Access-Control-Request-Method`, `Access-Control-Request-Headers` and `Origin` headers for the responses to be cached correctly.
 - **Quantity** (*integer*) – **[REQUIRED]** The number of HTTP methods for which you want CloudFront to cache responses. Valid values are 2 (for caching responses to GET and HEAD requests) and 3 (for caching responses to GET, HEAD, and OPTIONS requests).
 - **Items** (*list*) – **[REQUIRED]** A complex type that contains the HTTP methods that you want CloudFront to cache responses to.
 - (*string*) –
- **SmoothStreaming** (*boolean*) – Indicates whether you want to distribute media files in Microsoft Smooth Streaming format using the origin that is associated with this cache behavior. If so, specify `true`; if not, specify `false`.
- **DefaultTTL** (*integer*) – If you don't configure your origin to add a `Cache-Control max-age` directive or an `Expires` header, `DefaultTTL` is the default

amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin does not add HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).

- **MaxTTL** (*integer*) – The maximum amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin adds HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **CacheBehaviors** (*dict*) – A complex type that contains zero or more CacheBehavior elements.
 - **Quantity** (*integer*) – **[REQUIRED]** The number of cache behaviors for this distribution.
 - **Items** (*list*) – Optional: A complex type that contains cache behaviors for this distribution. If Quantity is 0, you can omit Items.
 - * (*dict*) – A complex type that describes how CloudFront processes requests. You can create up to 10 cache behaviors. You must create at least as many cache behaviors (including the default cache behavior) as you have origins if you want CloudFront to distribute objects from all of the origins. Each cache behavior specifies the one origin from which you want CloudFront to get objects. If you have two origins and only the default cache behavior, the default cache behavior will cause CloudFront to get objects from one of the origins, but the other origin will never be used. If you don't want to specify any cache behaviors, include only an empty CacheBehaviors element. Don't include an empty CacheBehavior element, or CloudFront returns a MalformedXML error. To delete all cache behaviors in an existing distribution, update the distribution configuration and include only an empty CacheBehaviors element. To add, change, or remove one or more cache behaviors, update the distribution configuration and specify all of the cache behaviors that you want to include in the updated distribution.
 - **PathPattern** (*string*) – **[REQUIRED]** The pattern (for example, images/*.jpg) that specifies which requests you want this cache behavior to apply to. When CloudFront receives an end-user request, the requested path is compared with path patterns in the order in which cache behaviors are listed in the distribution. The path pattern for the default cache behavior is * and cannot be changed. If the request for an object does not match the path pattern for any cache behaviors, CloudFront applies the behavior in the default cache behavior.
 - **TargetOriginId** (*string*) – **[REQUIRED]** The value of ID for the origin that you want CloudFront to route requests to when a request matches the path pattern either for a cache behavior or for the default cache behavior.
 - **ForwardedValues** (*dict*) – **[REQUIRED]** A complex type that specifies how CloudFront handles query strings, cookies and headers.
 - **QueryString** (*boolean*) – **[REQUIRED]** Indicates whether you want CloudFront to forward query strings to the origin

that is associated with this cache behavior. If so, specify true; if not, specify false.

- **Cookies** (*dict*) – **[REQUIRED]** A complex type that specifies how CloudFront handles cookies.
- **Forward** (*string*) – **[REQUIRED]** Use this element to specify whether you want CloudFront to forward cookies to the origin that is associated with this cache behavior. You can specify all, none or whitelist. If you choose All, CloudFront forwards all cookies regardless of how many your application uses.
- **WhitelistedNames** (*dict*) – A complex type that specifies the whitelisted cookies, if any, that you want CloudFront to forward to your origin that is associated with this cache behavior.
- **Quantity** (*integer*) – **[REQUIRED]** The number of whitelisted cookies for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains whitelisted cookies for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **Headers** (*dict*) – A complex type that specifies the Headers, if any, that you want CloudFront to vary upon for this cache behavior.
- **Quantity** (*integer*) – **[REQUIRED]** The number of different headers that you want CloudFront to forward to the origin and to vary on for this cache behavior. The maximum number of headers that you can specify by name is 10. If you want CloudFront to forward all headers to the origin and vary on all of them, specify 1 for Quantity and * for Name. If you don't want CloudFront to forward any additional headers to the origin or to vary on any headers, specify 0 for Quantity and omit Items.
- **Items** (*list*) – Optional: A complex type that contains a Name element for each header that you want CloudFront to forward to the origin and to vary on for this cache behavior. If Quantity is 0, omit Items.
- (*string*) –
- **TrustedSigners** (*dict*) – **[REQUIRED]** A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the PathPattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don't want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more trusted signers, change Enabled to true (if it's currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.
- **Enabled** (*boolean*) – **[REQUIRED]** Specifies whether you want to require end users to use signed URLs to access the files specified by PathPattern and TargetOriginId.

- **Quantity** (*integer*) – **[REQUIRED]** The number of trusted signers for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **ViewerProtocolPolicy** (*string*) – **[REQUIRED]** Use this element to specify the protocol that users can use to access the files in the origin specified by TargetOriginId when a request matches the path pattern in PathPattern. If you want CloudFront to allow end users to use any available protocol, specify allow-all. If you want CloudFront to require HTTPS, specify https. If you want CloudFront to respond to an HTTP request with an HTTP status code of 301 (Moved Permanently) and the HTTPS URL, specify redirect-to-https. The viewer then resubmits the request using the HTTPS URL.
- **MinTTL** (*integer*) – **[REQUIRED]** The minimum amount of time that you want objects to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **AllowedMethods** (*dict*) – A complex type that controls which HTTP methods CloudFront processes and forwards to your Amazon S3 bucket or your custom origin. There are three choices: - CloudFront forwards only GET and HEAD requests. - CloudFront forwards only GET, HEAD and OPTIONS requests. - CloudFront forwards GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests. If you pick the third choice, you may need to restrict access to your Amazon S3 bucket or to your custom origin so users can't perform operations that you don't want them to. For example, you may not want users to have permission to delete objects from your origin.
- **Quantity** (*integer*) – **[REQUIRED]** The number of HTTP methods that you want CloudFront to forward to your origin. Valid values are 2 (for GET and HEAD requests), 3 (for GET, HEAD and OPTIONS requests) and 7 (for GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests).
- **Items** (*list*) – **[REQUIRED]** A complex type that contains the HTTP methods that you want CloudFront to process and forward to your origin.
- (*string*) –
- **CachedMethods** (*dict*) – A complex type that controls whether CloudFront caches the response to requests using the specified HTTP methods. There are two choices: - CloudFront caches responses to GET and HEAD requests. - CloudFront caches responses to GET, HEAD, and OPTIONS requests. If you pick the second choice for your S3 Origin, you may need to forward Access-Control-Request-Method, Access-Control-Request-Headers and Origin headers for the responses to be cached correctly.
- **Quantity** (*integer*) – **[REQUIRED]** The number of HTTP methods for which you want CloudFront to cache responses. Valid values are 2 (for caching responses to GET and HEAD

- requests) and 3 (for caching responses to GET, HEAD, and OPTIONS requests).
- **Items** (*list*) – **[REQUIRED]** A complex type that contains the HTTP methods that you want CloudFront to cache responses to.
- (*string*) –
- **SmoothStreaming** (*boolean*) – Indicates whether you want to distribute media files in Microsoft Smooth Streaming format using the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
- **DefaultTTL** (*integer*) – If you don't configure your origin to add a Cache-Control max-age directive or an Expires header, DefaultTTL is the default amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin does not add HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **MaxTTL** (*integer*) – The maximum amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin adds HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **CustomErrorResponses** (*dict*) – A complex type that contains zero or more CustomErrorResponse elements.
 - **Quantity** (*integer*) – **[REQUIRED]** The number of custom error responses for this distribution.
 - **Items** (*list*) – Optional: A complex type that contains custom error responses for this distribution. If Quantity is 0, you can omit Items.
 - * (*dict*) – A complex type that describes how you'd prefer CloudFront to respond to requests that result in either a 4xx or 5xx response. You can control whether a custom error page should be displayed, what the desired response code should be for this error page and how long should the error response be cached by CloudFront. If you don't want to specify any custom error responses, include only an empty CustomErrorResponses element. To delete all custom error responses in an existing distribution, update the distribution configuration and include only an empty CustomErrorResponses element. To add, change, or remove one or more custom error responses, update the distribution configuration and specify all of the custom error responses that you want to include in the updated distribution.
 - **ErrorCode** (*integer*) – **[REQUIRED]** The 4xx or 5xx HTTP status code that you want to customize. For a list of HTTP status codes that you can customize, see CloudFront documentation.
 - **ResponsePagePath** (*string*) – The path of the custom error page (for example, /custom_404.html). The path is relative to the distribution and must begin with a slash (/). If the path includes any non-ASCII characters or unsafe characters as de-

defined in RFC 1783 (<http://www.ietf.org/rfc/rfc1738.txt>), URL encode those characters. Do not URL encode any other characters in the path, or CloudFront will not return the custom error page to the viewer.

- **ResponseCode** (*string*) – The HTTP status code that you want CloudFront to return with the custom error page to the viewer. For a list of HTTP status codes that you can replace, see CloudFront Documentation.
- **ErrorCachingMinTTL** (*integer*) – The minimum amount of time you want HTTP error codes to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 31,536,000.
- **Comment** (*string*) – **[REQUIRED]** Any comments you want to include about the distribution.
- **Logging** (*dict*) – A complex type that controls whether access logs are written for the distribution.
 - **Enabled** (*boolean*) – **[REQUIRED]** Specifies whether you want CloudFront to save access logs to an Amazon S3 bucket. If you do not want to enable logging when you create a distribution or if you want to disable logging for an existing distribution, specify false for Enabled, and specify empty Bucket and Prefix elements. If you specify false for Enabled but you specify values for Bucket, prefix and IncludeCookies, the values are automatically deleted.
 - **IncludeCookies** (*boolean*) – **[REQUIRED]** Specifies whether you want CloudFront to include cookies in access logs, specify true for IncludeCookies. If you choose to include cookies in logs, CloudFront logs all cookies regardless of how you configure the cache behaviors for this distribution. If you do not want to include cookies when you create a distribution or if you want to disable include cookies for an existing distribution, specify false for IncludeCookies.
 - **Bucket** (*string*) – **[REQUIRED]** The Amazon S3 bucket to store the access logs in, for example, myawslogbucket.s3.amazonaws.com.
 - **Prefix** (*string*) – **[REQUIRED]** An optional string that you want CloudFront to prefix to the access log filenames for this distribution, for example, myprefix/. If you want to enable logging, but you do not want to specify a prefix, you still must include an empty Prefix element in the Logging element.
- **PriceClass** (*string*) – A complex type that contains information about price class for this distribution.
- **Enabled** (*boolean*) – **[REQUIRED]** Whether the distribution is enabled to accept end user requests for content.
- **ViewerCertificate** (*dict*) – A complex type that contains information about viewer certificates for this distribution.
 - **IAMCertificateId** (*string*) – If you want viewers to use HTTPS to request your objects and you're using an alternate domain name in your object URLs (for example, <https://example.com/logo.jpg>), specify the IAM certificate identifier of the custom viewer certificate for this distribution. Specify either this value or CloudFrontDefaultCertificate.
 - **CloudFrontDefaultCertificate** (*boolean*) – If you want viewers to use HTTPS to request your objects and you're using the CloudFront domain name of your distribution in your object URLs (for example, <https://d1111111abcdef8.cloudfront.net/logo.jpg>), set to true. Omit this value if you are setting an IAMCertificateId.

- **SSLSupportMethod** (*string*) – If you specify a value for `IAMCertificateId`, you must also specify how you want CloudFront to serve HTTPS requests. Valid values are `vip` and `sni-only`. If you specify `vip`, CloudFront uses dedicated IP addresses for your content and can respond to HTTPS requests from any viewer. However, you must request permission to use this feature, and you incur additional monthly charges. If you specify `sni-only`, CloudFront can only respond to HTTPS requests from viewers that support Server Name Indication (SNI). All modern browsers support SNI, but some browsers still in use don't support SNI. Do not specify a value for `SSLSupportMethod` if you specified `true` for `CloudFrontDefaultCertificate`.
- **MinimumProtocolVersion** (*string*) – Specify the minimum version of the SSL protocol that you want CloudFront to use, `SSLv3` or `TLSv1`, for HTTPS connections. CloudFront will serve your objects only to browsers or devices that support at least the SSL version that you specify. The `TLSv1` protocol is more secure, so we recommend that you specify `SSLv3` only if your users are using browsers or devices that don't support `TLSv1`. If you're using a custom certificate (if you specify a value for `IAMCertificateId`) and if you're using dedicated IP (if you specify `vip` for `SSLSupportMethod`), you can choose `SSLv3` or `TLSv1` as the `MinimumProtocolVersion`. If you're using a custom certificate (if you specify a value for `IAMCertificateId`) and if you're using SNI (if you specify `sni-only` for `SSLSupportMethod`), you must specify `TLSv1` for `MinimumProtocolVersion`.
- **Restrictions** (*dict*) – A complex type that identifies ways in which you want to restrict distribution of your content.
 - **GeoRestriction** (*dict*) – **[REQUIRED]** A complex type that controls the countries in which your content is distributed. For more information about geo restriction, go to Customizing Error Responses in the Amazon CloudFront Developer Guide. CloudFront determines the location of your users using MaxMind GeoIP databases. For information about the accuracy of these databases, see *How accurate are your GeoIP databases?* on the MaxMind website.
 - * **RestrictionType** (*string*) – **[REQUIRED]** The method that you want to use to restrict distribution of your content by country:
 - `none`: No geo restriction is enabled, meaning access to content is not restricted by client geo location.
 - `blacklist`: The `Location` elements specify the countries in which you do not want CloudFront to distribute your content.
 - `whitelist`: The `Location` elements specify the countries in which you want CloudFront to distribute your content.
 - * **Quantity** (*integer*) – **[REQUIRED]** When geo restriction is enabled, this is the number of countries in your whitelist or blacklist. Otherwise, when it is not enabled, `Quantity` is 0, and you can omit `Items`.
 - * **Items** (*list*) – A complex type that contains a `Location` element for each country in which you want CloudFront either to distribute your content (whitelist) or not distribute your content (blacklist). The `Location` element is a two-letter, uppercase country code for a country that you want to include in your blacklist or whitelist. Include one `Location` element for each country. CloudFront and MaxMind both use ISO 3166 country codes. For the current list of countries and the corresponding codes, see ISO 3166-1-alpha-2 code on the International Organization for Standardization website. You can also refer

to the country list in the CloudFront console, which includes both country names and codes.

· (string)–

Return type dict

Returns

Response Syntax

```
{
  'Distribution': {
    'Id': 'string',
    'Status': 'string',
    'LastModifiedTime': datetime(2015, 1, 1),
    'InProgressInvalidationBatches': 123,
    'DomainName': 'string',
    'ActiveTrustedSigners': {
      'Enabled': True|False,
      'Quantity': 123,
      'Items': [
        {
          'AwsAccountNumber': 'string',
          'KeyPairIds': {
            'Quantity': 123,
            'Items': [
              'string',
            ]
          },
        },
      ]
    },
  },
  'DistributionConfig': {
    'CallerReference': 'string',
    'Aliases': {
      'Quantity': 123,
      'Items': [
        'string',
      ]
    },
    'DefaultRootObject': 'string',
    'Origins': {
      'Quantity': 123,
      'Items': [
        {
          'Id': 'string',
          'DomainName': 'string',
          'OriginPath': 'string',
          'S3OriginConfig': {
            'OriginAccessIdentity': 'string'
          },
          'CustomOriginConfig': {
            'HTTPPort': 123,
            'HTTPSPort': 123,
            'OriginProtocolPolicy': 'http-only'|'match-viewer'
          },
        },
      ],
    },
  },
  'DefaultCacheBehavior': {
```



```

'TargetOriginId': 'string',
'ForwardedValues': {
    'QueryString': True|False,
    'Cookies': {
        'Forward': 'none'|'whitelist'|'all',
        'WhitelistedNames': {
            'Quantity': 123,
            'Items': [
                'string',
            ]
        }
    },
    'Headers': {
        'Quantity': 123,
        'Items': [
            'string',
        ]
    }
},
'TrustedSigners': {
    'Enabled': True|False,
    'Quantity': 123,
    'Items': [
        'string',
    ]
},
'ViewerProtocolPolicy': 'allow-all'|'https-only'|'redirect-to-http',
'MinTTL': 123,
'AllowedMethods': {
    'Quantity': 123,
    'Items': [
        'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
    ],
    'CachedMethods': {
        'Quantity': 123,
        'Items': [
            'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
        ]
    }
},
'SmoothStreaming': True|False,
'DefaultTTL': 123,
'MaxTTL': 123
},
'CacheBehaviors': {
    'Quantity': 123,
    'Items': [
        {
            'PathPattern': 'string',
            'TargetOriginId': 'string',
            'ForwardedValues': {
                'QueryString': True|False,
                'Cookies': {
                    'Forward': 'none'|'whitelist'|'all',
                    'WhitelistedNames': {
                        'Quantity': 123,
                        'Items': [
                            'string',
                        ]
                    }
                }
            }
        }
    ]
}

```

```
        ]
    },
    'Headers': {
        'Quantity': 123,
        'Items': [
            'string',
        ]
    },
    'TrustedSigners': {
        'Enabled': True|False,
        'Quantity': 123,
        'Items': [
            'string',
        ]
    },
    'ViewerProtocolPolicy': 'allow-all'|'https-only'|'redirect',
    'MinTTL': 123,
    'AllowedMethods': {
        'Quantity': 123,
        'Items': [
            'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
        ],
        'CachedMethods': {
            'Quantity': 123,
            'Items': [
                'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
            ]
        }
    },
    'SmoothStreaming': True|False,
    'DefaultTTL': 123,
    'MaxTTL': 123
},
]
},
'CustomErrorResponses': {
    'Quantity': 123,
    'Items': [
        {
            'ErrorCode': 123,
            'ResponsePagePath': 'string',
            'ResponseCode': 'string',
            'ErrorCachingMinTTL': 123
        },
    ],
},
],
'Comment': 'string',
'Logging': {
    'Enabled': True|False,
    'IncludeCookies': True|False,
    'Bucket': 'string',
    'Prefix': 'string'
},
'PriceClass': 'PriceClass_100'|'PriceClass_200'|'PriceClass_All',
'Enabled': True|False,
'ViewerCertificate': {
```

```

        'IAMCertificateId': 'string',
        'CloudFrontDefaultCertificate': True|False,
        'SSLSupportMethod': 'sni-only'|'vip',
        'MinimumProtocolVersion': 'SSLv3'|'TLSv1'
    },
    'Restrictions': {
        'GeoRestriction': {
            'RestrictionType': 'blacklist'|'whitelist'|'none',
            'Quantity': 123,
            'Items': [
                'string',
            ]
        }
    }
},
'Location': 'string',
'ETag': 'string'
}

```

Response Structure

- (*dict*) – The returned result of the corresponding request.
 - **Distribution** (*dict*) – The distribution’s information.
 - * **Id** (*string*) – The identifier for the distribution. For example: EDFD-VBD632BHDS5.
 - * **Status** (*string*) – This response element indicates the current status of the distribution. When the status is Deployed, the distribution’s information is fully propagated throughout the Amazon CloudFront system.
 - * **LastModifiedTime** (*datetime*) – The date and time the distribution was last modified.
 - * **InProgressInvalidationBatches** (*integer*) – The number of invalidation batches currently in progress.
 - * **DomainName** (*string*) – The domain name corresponding to the distribution. For example: d604721fxaaqy9.cloudfront.net.
 - * **ActiveTrustedSigners** (*dict*) – CloudFront automatically adds this element to the response only if you’ve set up the distribution to serve private content with signed URLs. The element lists the key pair IDs that CloudFront is aware of for each trusted signer. The Signer child element lists the AWS account number of the trusted signer (or an empty Self element if the signer is you). The Signer element also includes the IDs of any active key pairs associated with the trusted signer’s AWS account. If no KeyPairId element appears for a Signer, that signer can’t create working signed URLs.
 - **Enabled** (*boolean*) – Each active trusted signer.
 - **Quantity** (*integer*) – The number of unique trusted signers included in all cache behaviors. For example, if three cache behaviors all list the same three AWS accounts, the value of Quantity for ActiveTrustedSigners will be 3.
 - **Items** (*list*) – A complex type that contains one Signer complex type for each unique trusted signer that is specified in the TrustedSigners complex type, including trusted signers in the default cache behavior and in all of the other cache behaviors.
 - (*dict*) – A complex type that lists the AWS accounts that were included in the TrustedSigners complex type, as well as their

- active CloudFront key pair IDs, if any.
- **AwsAccountNumber** (*string*) – Specifies an AWS account that can create signed URLs. Values: self, which indicates that the AWS account that was used to create the distribution can create signed URLs, or an AWS account number. Omit the dashes in the account number.
- **KeyPairIds** (*dict*) – A complex type that lists the active CloudFront key pairs, if any, that are associated with AwsAccountNumber.
- **Quantity** (*integer*) – The number of active CloudFront key pairs for AwsAccountNumber.
- **Items** (*list*) – A complex type that lists the active CloudFront key pairs, if any, that are associated with AwsAccountNumber.
- (*string*) –
- * **DistributionConfig** (*dict*) – The current configuration information for the distribution.
 - **CallerReference** (*string*) – A unique number that ensures the request can't be replayed. If the CallerReference is new (no matter the content of the DistributionConfig object), a new distribution is created. If the CallerReference is a value you already sent in a previous request to create a distribution, and the content of the DistributionConfig is identical to the original request (ignoring white space), the response includes the same information returned to the original request. If the CallerReference is a value you already sent in a previous request to create a distribution but the content of the DistributionConfig is different from the original request, CloudFront returns a DistributionAlreadyExists error.
 - **Aliases** (*dict*) – A complex type that contains information about CNAMEs (alternate domain names), if any, for this distribution.
 - **Quantity** (*integer*) – The number of CNAMEs, if any, for this distribution.
 - **Items** (*list*) – Optional: A complex type that contains CNAME elements, if any, for this distribution. If Quantity is 0, you can omit Items.
 - (*string*) –
 - **DefaultRootObject** (*string*) – The object that you want CloudFront to return (for example, index.html) when an end user requests the root URL for your distribution (<http://www.example.com>) instead of an object in your distribution (<http://www.example.com/index.html>). Specifying a default root object avoids exposing the contents of your distribution. If you don't want to specify a default root object when you create a distribution, include an empty DefaultRootObject element. To delete the default root object from an existing distribution, update the distribution configuration and include an empty DefaultRootObject element. To replace the default root object, update the distribution configuration and specify the new object.
 - **Origins** (*dict*) – A complex type that contains information about origins for this distribution.
 - **Quantity** (*integer*) – The number of origins for this distribu-

tion.

- **Items** (*list*) – A complex type that contains origins for this distribution.
- (*dict*) – A complex type that describes the Amazon S3 bucket or the HTTP server (for example, a web server) from which CloudFront gets your files. You must create at least one origin.
- **Id** (*string*) – A unique identifier for the origin. The value of Id must be unique within the distribution. You use the value of Id when you create a cache behavior. The Id identifies the origin that CloudFront routes a request to when the request matches the path pattern for that cache behavior.
- **DomainName** (*string*) – Amazon S3 origins: The DNS name of the Amazon S3 bucket from which you want CloudFront to get objects for this origin, for example, myaws-bucket.s3.amazonaws.com. Custom origins: The DNS domain name for the HTTP server from which you want CloudFront to get objects for this origin, for example, www.example.com.
- **OriginPath** (*string*) – An optional element that causes CloudFront to request your content from a directory in your Amazon S3 bucket or your custom origin. When you include the OriginPath element, specify the directory name, beginning with a /. CloudFront appends the directory name to the value of DomainName.
- **S3OriginConfig** (*dict*) – A complex type that contains information about the Amazon S3 origin. If the origin is a custom origin, use the CustomOriginConfig element instead.
- **OriginAccessIdentity** (*string*) – The CloudFront origin access identity to associate with the origin. Use an origin access identity to configure the origin so that end users can only access objects in an Amazon S3 bucket through CloudFront. If you want end users to be able to access objects using either the CloudFront URL or the Amazon S3 URL, specify an empty OriginAccessIdentity element. To delete the origin access identity from an existing distribution, update the distribution configuration and include an empty OriginAccessIdentity element. To replace the origin access identity, update the distribution configuration and specify the new origin access identity. Use the format origin-access-identity/cloudfront/Id where Id is the value that CloudFront returned in the Id element when you created the origin access identity.
- **CustomOriginConfig** (*dict*) – A complex type that contains information about a custom origin. If the origin is an Amazon S3 bucket, use the S3OriginConfig element instead.
- **HTTPPort** (*integer*) – The HTTP port the custom origin listens on.
- **HTTPSPort** (*integer*) – The HTTPS port the custom origin listens on.
- **OriginProtocolPolicy** (*string*) – The origin protocol policy to apply to your origin.
- **DefaultCacheBehavior** (*dict*) – A complex type that describes the default cache behavior if you do not specify a CacheBehavior element or if files don't match any of the values of PathPattern in CacheBehavior elements. You must cre-

ate exactly one default cache behavior.

- **TargetOriginId** (*string*) – The value of ID for the origin that you want CloudFront to route requests to when a request matches the path pattern either for a cache behavior or for the default cache behavior.
- **ForwardedValues** (*dict*) – A complex type that specifies how CloudFront handles query strings, cookies and headers.
- **QueryString** (*boolean*) – Indicates whether you want CloudFront to forward query strings to the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
- **Cookies** (*dict*) – A complex type that specifies how CloudFront handles cookies.
- **Forward** (*string*) – Use this element to specify whether you want CloudFront to forward cookies to the origin that is associated with this cache behavior. You can specify all, none or whitelist. If you choose All, CloudFront forwards all cookies regardless of how many your application uses.
- **WhitelistedNames** (*dict*) – A complex type that specifies the whitelisted cookies, if any, that you want CloudFront to forward to your origin that is associated with this cache behavior.
- **Quantity** (*integer*) – The number of whitelisted cookies for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains whitelisted cookies for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **Headers** (*dict*) – A complex type that specifies the Headers, if any, that you want CloudFront to vary upon for this cache behavior.
- **Quantity** (*integer*) – The number of different headers that you want CloudFront to forward to the origin and to vary on for this cache behavior. The maximum number of headers that you can specify by name is 10. If you want CloudFront to forward all headers to the origin and vary on all of them, specify 1 for Quantity and * for Name. If you don't want CloudFront to forward any additional headers to the origin or to vary on any headers, specify 0 for Quantity and omit Items.
- **Items** (*list*) – Optional: A complex type that contains a Name element for each header that you want CloudFront to forward to the origin and to vary on for this cache behavior. If Quantity is 0, omit Items.
- (*string*) –
- **TrustedSigners** (*dict*) – A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the PathPattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don't want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more

trusted signers, change Enabled to true (if it's currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.

- **Enabled** (*boolean*) – Specifies whether you want to require end users to use signed URLs to access the files specified by PathPattern and TargetOriginId.
- **Quantity** (*integer*) – The number of trusted signers for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **ViewerProtocolPolicy** (*string*) – Use this element to specify the protocol that users can use to access the files in the origin specified by TargetOriginId when a request matches the path pattern in PathPattern. If you want CloudFront to allow end users to use any available protocol, specify allow-all. If you want CloudFront to require HTTPS, specify https. If you want CloudFront to respond to an HTTP request with an HTTP status code of 301 (Moved Permanently) and the HTTPS URL, specify redirect-to-https. The viewer then resubmits the request using the HTTPS URL.
- **MinTTL** (*integer*) – The minimum amount of time that you want objects to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **AllowedMethods** (*dict*) – A complex type that controls which HTTP methods CloudFront processes and forwards to your Amazon S3 bucket or your custom origin. There are three choices:
 - CloudFront forwards only GET and HEAD requests.
 - CloudFront forwards only GET, HEAD and OPTIONS requests.
 - CloudFront forwards GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests.
 If you pick the third choice, you may need to restrict access to your Amazon S3 bucket or to your custom origin so users can't perform operations that you don't want them to. For example, you may not want users to have permission to delete objects from your origin.
- **Quantity** (*integer*) – The number of HTTP methods that you want CloudFront to forward to your origin. Valid values are 2 (for GET and HEAD requests), 3 (for GET, HEAD and OPTIONS requests) and 7 (for GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests).
- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to process and forward to your origin.
- (*string*) –
- **CachedMethods** (*dict*) – A complex type that controls whether CloudFront caches the response to requests using the specified HTTP methods. There are two choices:
 - CloudFront caches responses to GET and HEAD requests.
 - CloudFront caches responses to GET, HEAD, and OPTIONS requests.
 If you pick the second choice for your S3 Origin,

you may need to forward Access-Control-Request-Method, Access-Control-Request-Headers and Origin headers for the responses to be cached correctly.

- **Quantity** (*integer*) – The number of HTTP methods for which you want CloudFront to cache responses. Valid values are 2 (for caching responses to GET and HEAD requests) and 3 (for caching responses to GET, HEAD, and OPTIONS requests).
- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to cache responses to.
- (*string*) –
- **SmoothStreaming** (*boolean*) – Indicates whether you want to distribute media files in Microsoft Smooth Streaming format using the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
- **DefaultTTL** (*integer*) – If you don't configure your origin to add a Cache-Control max-age directive or an Expires header, DefaultTTL is the default amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin does not add HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **MaxTTL** (*integer*) – The maximum amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin adds HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **CacheBehaviors** (*dict*) – A complex type that contains zero or more CacheBehavior elements.
- **Quantity** (*integer*) – The number of cache behaviors for this distribution.
- **Items** (*list*) – Optional: A complex type that contains cache behaviors for this distribution. If Quantity is 0, you can omit Items.
- (*dict*) – A complex type that describes how CloudFront processes requests. You can create up to 10 cache behaviors. You must create at least as many cache behaviors (including the default cache behavior) as you have origins if you want CloudFront to distribute objects from all of the origins. Each cache behavior specifies the one origin from which you want CloudFront to get objects. If you have two origins and only the default cache behavior, the default cache behavior will cause CloudFront to get objects from one of the origins, but the other origin will never be used. If you don't want to specify any cache behaviors, include only an empty CacheBehaviors element. Don't include an empty CacheBehavior element, or CloudFront returns a MalformedXML error. To delete all cache behaviors in an existing distribution, update the distribution configuration and include only an

empty CacheBehaviors element. To add, change, or remove one or more cache behaviors, update the distribution configuration and specify all of the cache behaviors that you want to include in the updated distribution.

- **PathPattern** (*string*) – The pattern (for example, images/*.jpg) that specifies which requests you want this cache behavior to apply to. When CloudFront receives an end-user request, the requested path is compared with path patterns in the order in which cache behaviors are listed in the distribution. The path pattern for the default cache behavior is * and cannot be changed. If the request for an object does not match the path pattern for any cache behaviors, CloudFront applies the behavior in the default cache behavior.
- **TargetOriginId** (*string*) – The value of ID for the origin that you want CloudFront to route requests to when a request matches the path pattern either for a cache behavior or for the default cache behavior.
- **ForwardedValues** (*dict*) – A complex type that specifies how CloudFront handles query strings, cookies and headers.
- **QueryString** (*boolean*) – Indicates whether you want CloudFront to forward query strings to the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
- **Cookies** (*dict*) – A complex type that specifies how CloudFront handles cookies.
- **Forward** (*string*) – Use this element to specify whether you want CloudFront to forward cookies to the origin that is associated with this cache behavior. You can specify all, none or whitelist. If you choose All, CloudFront forwards all cookies regardless of how many your application uses.
- **WhitelistedNames** (*dict*) – A complex type that specifies the whitelisted cookies, if any, that you want CloudFront to forward to your origin that is associated with this cache behavior.
- **Quantity** (*integer*) – The number of whitelisted cookies for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains whitelisted cookies for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **Headers** (*dict*) – A complex type that specifies the Headers, if any, that you want CloudFront to vary upon for this cache behavior.
- **Quantity** (*integer*) – The number of different headers that you want CloudFront to forward to the origin and to vary on for this cache behavior. The maximum number of headers that you can specify by name is 10. If you want CloudFront to forward all headers to the origin and vary on all of them, specify 1 for Quantity and * for Name. If you don't want CloudFront to forward any additional headers to the origin or to vary on any headers, specify 0 for Quantity and omit Items.
- **Items** (*list*) – Optional: A complex type that contains a Name element for each header that you want CloudFront to forward to the origin and to vary on for this cache behavior. If Quantity is 0, omit Items.

- *(string)* –
- **TrustedSigners** (*dict*) – A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the PathPattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don't want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more trusted signers, change Enabled to true (if it's currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.
- **Enabled** (*boolean*) – Specifies whether you want to require end users to use signed URLs to access the files specified by PathPattern and TargetOriginId.
- **Quantity** (*integer*) – The number of trusted signers for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit Items.
- *(string)* –
- **ViewerProtocolPolicy** (*string*) – Use this element to specify the protocol that users can use to access the files in the origin specified by TargetOriginId when a request matches the path pattern in PathPattern. If you want CloudFront to allow end users to use any available protocol, specify allow-all. If you want CloudFront to require HTTPS, specify https. If you want CloudFront to respond to an HTTP request with an HTTP status code of 301 (Moved Permanently) and the HTTPS URL, specify redirect-to-https. The viewer then resubmits the request using the HTTPS URL.
- **MinTTL** (*integer*) – The minimum amount of time that you want objects to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **AllowedMethods** (*dict*) – A complex type that controls which HTTP methods CloudFront processes and forwards to your Amazon S3 bucket or your custom origin. There are three choices: - CloudFront forwards only GET and HEAD requests. - CloudFront forwards only GET, HEAD and OPTIONS requests. - CloudFront forwards GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests. If you pick the third choice, you may need to restrict access to your Amazon S3 bucket or to your custom origin so users can't perform operations that you don't want them to. For example, you may not want users to have permission to delete objects from your origin.
- **Quantity** (*integer*) – The number of HTTP methods that you want CloudFront to forward to your origin. Valid values are 2 (for GET and HEAD requests), 3 (for GET, HEAD and OP-

TIONS requests) and 7 (for GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests).

- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to process and forward to your origin.
- (*string*) –
- **CachedMethods** (*dict*) – A complex type that controls whether CloudFront caches the response to requests using the specified HTTP methods. There are two choices: - CloudFront caches responses to GET and HEAD requests. - CloudFront caches responses to GET, HEAD, and OPTIONS requests. If you pick the second choice for your S3 Origin, you may need to forward Access-Control-Request-Method, Access-Control-Request-Headers and Origin headers for the responses to be cached correctly.
- **Quantity** (*integer*) – The number of HTTP methods for which you want CloudFront to cache responses. Valid values are 2 (for caching responses to GET and HEAD requests) and 3 (for caching responses to GET, HEAD, and OPTIONS requests).
- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to cache responses to.
- (*string*) –
- **SmoothStreaming** (*boolean*) – Indicates whether you want to distribute media files in Microsoft Smooth Streaming format using the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
- **DefaultTTL** (*integer*) – If you don't configure your origin to add a Cache-Control max-age directive or an Expires header, DefaultTTL is the default amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin does not add HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **MaxTTL** (*integer*) – The maximum amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin adds HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **CustomErrorResponses** (*dict*) – A complex type that contains zero or more CustomErrorResponse elements.
- **Quantity** (*integer*) – The number of custom error responses for this distribution.
- **Items** (*list*) – Optional: A complex type that contains custom error responses for this distribution. If Quantity is 0, you can omit Items.
- (*dict*) – A complex type that describes how you'd prefer CloudFront to respond to requests that result in either a 4xx or 5xx response. You can control whether a custom error page

should be displayed, what the desired response code should be for this error page and how long should the error response be cached by CloudFront. If you don't want to specify any custom error responses, include only an empty `CustomErrorResponses` element. To delete all custom error responses in an existing distribution, update the distribution configuration and include only an empty `CustomErrorResponses` element. To add, change, or remove one or more custom error responses, update the distribution configuration and specify all of the custom error responses that you want to include in the updated distribution.

- **ErrorCode** (*integer*) – The 4xx or 5xx HTTP status code that you want to customize. For a list of HTTP status codes that you can customize, see CloudFront documentation.
- **ResponsePagePath** (*string*) – The path of the custom error page (for example, `/custom_404.html`). The path is relative to the distribution and must begin with a slash (/). If the path includes any non-ASCII characters or unsafe characters as defined in RFC 1783 (<http://www.ietf.org/rfc/rfc1738.txt>), URL encode those characters. Do not URL encode any other characters in the path, or CloudFront will not return the custom error page to the viewer.
- **ResponseCode** (*string*) – The HTTP status code that you want CloudFront to return with the custom error page to the viewer. For a list of HTTP status codes that you can replace, see CloudFront Documentation.
- **ErrorCachingMinTTL** (*integer*) – The minimum amount of time you want HTTP error codes to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 31,536,000.
- **Comment** (*string*) – Any comments you want to include about the distribution.
- **Logging** (*dict*) – A complex type that controls whether access logs are written for the distribution.
- **Enabled** (*boolean*) – Specifies whether you want CloudFront to save access logs to an Amazon S3 bucket. If you do not want to enable logging when you create a distribution or if you want to disable logging for an existing distribution, specify `false` for `Enabled`, and specify empty `Bucket` and `Prefix` elements. If you specify `false` for `Enabled` but you specify values for `Bucket`, `prefix` and `IncludeCookies`, the values are automatically deleted.
- **IncludeCookies** (*boolean*) – Specifies whether you want CloudFront to include cookies in access logs, specify `true` for `IncludeCookies`. If you choose to include cookies in logs, CloudFront logs all cookies regardless of how you configure the cache behaviors for this distribution. If you do not want to include cookies when you create a distribution or if you want to disable include cookies for an existing distribution, specify `false` for `IncludeCookies`.
- **Bucket** (*string*) – The Amazon S3 bucket to store the access logs in, for example, `myawslogbucket.s3.amazonaws.com`.
- **Prefix** (*string*) – An optional string that you want CloudFront

to prefix to the access log filenames for this distribution, for example, `myprefix/`. If you want to enable logging, but you do not want to specify a prefix, you still must include an empty `Prefix` element in the `Logging` element.

- **PriceClass** (*string*) – A complex type that contains information about price class for this distribution.
- **Enabled** (*boolean*) – Whether the distribution is enabled to accept end user requests for content.
- **ViewerCertificate** (*dict*) – A complex type that contains information about viewer certificates for this distribution.
- **IAMCertificateId** (*string*) – If you want viewers to use HTTPS to request your objects and you’re using an alternate domain name in your object URLs (for example, <https://example.com/logo.jpg>), specify the IAM certificate identifier of the custom viewer certificate for this distribution. Specify either this value or `CloudFrontDefaultCertificate`.
- **CloudFrontDefaultCertificate** (*boolean*) – If you want viewers to use HTTPS to request your objects and you’re using the CloudFront domain name of your distribution in your object URLs (for example, <https://d1111111abcdef8.cloudfront.net/logo.jpg>), set to `true`. Omit this value if you are setting an `IAMCertificateId`.
- **SSLSupportMethod** (*string*) – If you specify a value for `IAMCertificateId`, you must also specify how you want CloudFront to serve HTTPS requests. Valid values are `vip` and `sni-only`. If you specify `vip`, CloudFront uses dedicated IP addresses for your content and can respond to HTTPS requests from any viewer. However, you must request permission to use this feature, and you incur additional monthly charges. If you specify `sni-only`, CloudFront can only respond to HTTPS requests from viewers that support Server Name Indication (SNI). All modern browsers support SNI, but some browsers still in use don’t support SNI. Do not specify a value for `SSLSupportMethod` if you specified `true` for `CloudFrontDefaultCertificate`.
- **MinimumProtocolVersion** (*string*) – Specify the minimum version of the SSL protocol that you want CloudFront to use, `SSLv3` or `TLSv1`, for HTTPS connections. CloudFront will serve your objects only to browsers or devices that support at least the SSL version that you specify. The `TLSv1` protocol is more secure, so we recommend that you specify `SSLv3` only if your users are using browsers or devices that don’t support `TLSv1`. If you’re using a custom certificate (if you specify a value for `IAMCertificateId`) and if you’re using dedicated IP (if you specify `vip` for `SSLSupportMethod`), you can choose `SSLv3` or `TLSv1` as the `MinimumProtocolVersion`. If you’re using a custom certificate (if you specify a value for `IAMCertificateId`) and if you’re using SNI (if you specify `sni-only` for `SSLSupportMethod`), you must specify `TLSv1` for `MinimumProtocolVersion`.
- **Restrictions** (*dict*) – A complex type that identifies ways in which you want to restrict distribution of your content.
- **GeoRestriction** (*dict*) – A complex type that controls the countries in which your content is distributed. For more in-

information about geo restriction, go to Customizing Error Responses in the Amazon CloudFront Developer Guide. CloudFront determines the location of your users using MaxMind GeoIP databases. For information about the accuracy of these databases, see How accurate are your GeoIP databases? on the MaxMind website.

- **RestrictionType** (*string*) – The method that you want to use to restrict distribution of your content by country: - none: No geo restriction is enabled, meaning access to content is not restricted by client geo location. - blacklist: The Location elements specify the countries in which you do not want CloudFront to distribute your content. - whitelist: The Location elements specify the countries in which you want CloudFront to distribute your content.
- **Quantity** (*integer*) – When geo restriction is enabled, this is the number of countries in your whitelist or blacklist. Otherwise, when it is not enabled, Quantity is 0, and you can omit Items.
- **Items** (*list*) – A complex type that contains a Location element for each country in which you want CloudFront either to distribute your content (whitelist) or not distribute your content (blacklist). The Location element is a two-letter, uppercase country code for a country that you want to include in your blacklist or whitelist. Include one Location element for each country. CloudFront and MaxMind both use ISO 3166 country codes. For the current list of countries and the corresponding codes, see ISO 3166-1-alpha-2 code on the International Organization for Standardization website. You can also refer to the country list in the CloudFront console, which includes both country names and codes.
- (*string*) –
 - **Location** (*string*) – The fully qualified URI of the new distribution resource just created. For example: <https://cloudfront.amazonaws.com/2010-11-01/distribution/EDFDVBD632BHDS5>.
 - **ETag** (*string*) – The current version of the distribution created.

create_invalidation (**kwargs)

Create a new invalidation.

Request Syntax

```
response = client.create_invalidation(
    DistributionId='string',
    InvalidationBatch={
        'Paths': {
            'Quantity': 123,
            'Items': [
                'string',
            ]
        },
        'CallerReference': 'string'
    }
)
```

Parameters

- **DistributionId** (*string*) – [REQUIRED] The distribution’s id.
- **InvalidationBatch** (*dict*) – [REQUIRED] The batch information for the invalidation.
 - **Paths** (*dict*) – [REQUIRED] The path of the object to invalidate. The path is relative to the distribution and must begin with a slash (/). You must enclose each invalidation object with the Path element tags. If the path includes non-ASCII characters or unsafe characters as defined in RFC 1783 (<http://www.ietf.org/rfc/rfc1738.txt>), URL encode those characters. Do not URL encode any other characters in the path, or CloudFront will not invalidate the old version of the updated object.
 - * **Quantity** (*integer*) – [REQUIRED] The number of objects that you want to invalidate.
 - * **Items** (*list*) – A complex type that contains a list of the objects that you want to invalidate.
 - (*string*) –
 - **CallerReference** (*string*) – [REQUIRED] A unique name that ensures the request can’t be replayed. If the CallerReference is new (no matter the content of the Path object), a new distribution is created. If the CallerReference is a value you already sent in a previous request to create an invalidation batch, and the content of each Path element is identical to the original request, the response includes the same information returned to the original request. If the CallerReference is a value you already sent in a previous request to create a distribution but the content of any Path is different from the original request, CloudFront returns an InvalidationBatchAlreadyExists error.

Return type dict

Returns

Response Syntax

```
{
  'Location': 'string',
  'Invalidation': {
    'Id': 'string',
    'Status': 'string',
    'CreateTime': datetime(2015, 1, 1),
    'InvalidationBatch': {
      'Paths': {
        'Quantity': 123,
        'Items': [
          'string',
        ]
      },
      'CallerReference': 'string'
    }
  }
}
```

Response Structure

- (*dict*) – The returned result of the corresponding request.
 - **Location** (*string*) – The fully qualified URI of the distribution and invalidation batch request, including the Invalidation ID.
 - **Invalidation** (*dict*) – The invalidation’s information.
 - * **Id** (*string*) – The identifier for the invalidation request. For example: IDFDVBD632BHDS5.

- * **Status** (*string*) – The status of the invalidation request. When the invalidation batch is finished, the status is `Completed`.
- * **CreateTime** (*datetime*) – The date and time the invalidation request was first made.
- * **InvalidationBatch** (*dict*) – The current invalidation information for the batch request.
 - **Paths** (*dict*) – The path of the object to invalidate. The path is relative to the distribution and must begin with a slash (/). You must enclose each invalidation object with the Path element tags. If the path includes non-ASCII characters or unsafe characters as defined in RFC 1738 (<http://www.ietf.org/rfc/rfc1738.txt>), URL encode those characters. Do not URL encode any other characters in the path, or CloudFront will not invalidate the old version of the updated object.
 - **Quantity** (*integer*) – The number of objects that you want to invalidate.
 - **Items** (*list*) – A complex type that contains a list of the objects that you want to invalidate.
 - (*string*) –
 - **CallerReference** (*string*) – A unique name that ensures the request can't be replayed. If the CallerReference is new (no matter the content of the Path object), a new distribution is created. If the CallerReference is a value you already sent in a previous request to create an invalidation batch, and the content of each Path element is identical to the original request, the response includes the same information returned to the original request. If the CallerReference is a value you already sent in a previous request to create a distribution but the content of any Path is different from the original request, CloudFront returns an `InvalidationBatchAlreadyExists` error.

create_streaming_distribution (**kwargs)

Create a new streaming distribution.

Request Syntax

```
response = client.create_streaming_distribution(
    StreamingDistributionConfig={
        'CallerReference': 'string',
        'S3Origin': {
            'DomainName': 'string',
            'OriginAccessIdentity': 'string'
        },
        'Aliases': {
            'Quantity': 123,
            'Items': [
                'string',
            ]
        },
        'Comment': 'string',
        'Logging': {
            'Enabled': True|False,
            'Bucket': 'string',
            'Prefix': 'string'
        },
    },
```



```

    'TrustedSigners': {
        'Enabled': True|False,
        'Quantity': 123,
        'Items': [
            'string',
        ]
    },
    'PriceClass': 'PriceClass_100'|'PriceClass_200'|'PriceClass_All',
    'Enabled': True|False
}
)

```

Parameters **StreamingDistributionConfig** (*dict*) – **[REQUIRED]** The streaming distribution’s configuration information.

- **CallerReference** (*string*) – **[REQUIRED]** A unique number that ensures the request can’t be replayed. If the CallerReference is new (no matter the content of the StreamingDistributionConfig object), a new streaming distribution is created. If the CallerReference is a value you already sent in a previous request to create a streaming distribution, and the content of the StreamingDistributionConfig is identical to the original request (ignoring white space), the response includes the same information returned to the original request. If the CallerReference is a value you already sent in a previous request to create a streaming distribution but the content of the StreamingDistributionConfig is different from the original request, CloudFront returns a DistributionAlreadyExists error.
- **S3Origin** (*dict*) – **[REQUIRED]** A complex type that contains information about the Amazon S3 bucket from which you want CloudFront to get your media files for distribution.
 - **DomainName** (*string*) – **[REQUIRED]** The DNS name of the S3 origin.
 - **OriginAccessIdentity** (*string*) – **[REQUIRED]** Your S3 origin’s origin access identity.
- **Aliases** (*dict*) – A complex type that contains information about CNAMEs (alternate domain names), if any, for this streaming distribution.
 - **Quantity** (*integer*) – **[REQUIRED]** The number of CNAMEs, if any, for this distribution.
 - **Items** (*list*) – Optional: A complex type that contains CNAME elements, if any, for this distribution. If Quantity is 0, you can omit Items.
 - * (*string*) –
- **Comment** (*string*) – **[REQUIRED]** Any comments you want to include about the streaming distribution.
- **Logging** (*dict*) – A complex type that controls whether access logs are written for the streaming distribution.
 - **Enabled** (*boolean*) – **[REQUIRED]** Specifies whether you want CloudFront to save access logs to an Amazon S3 bucket. If you do not want to enable logging when you create a streaming distribution or if you want to disable logging for an existing streaming distribution, specify false for Enabled, and specify empty Bucket and Prefix elements. If you specify false for Enabled but you specify values for Bucket and Prefix, the values are automatically deleted.
 - **Bucket** (*string*) – **[REQUIRED]** The Amazon S3 bucket to store the access logs in, for example, myawslogbucket.s3.amazonaws.com.
 - **Prefix** (*string*) – **[REQUIRED]** An optional string that you want CloudFront to prefix to the access log filenames for this streaming distribution, for example, myprefix/. If you want to enable logging, but you do not want to specify a prefix, you still must include an empty Prefix element in the Logging element.

- **TrustedSigners** (*dict*) – **[REQUIRED]** A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the PathPattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don't want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more trusted signers, change Enabled to true (if it's currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.
 - **Enabled** (*boolean*) – **[REQUIRED]** Specifies whether you want to require end users to use signed URLs to access the files specified by PathPattern and TargetOriginId.
 - **Quantity** (*integer*) – **[REQUIRED]** The number of trusted signers for this cache behavior.
 - **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit Items.
 - * (*string*) –
- **PriceClass** (*string*) – A complex type that contains information about price class for this streaming distribution.
- **Enabled** (*boolean*) – **[REQUIRED]** Whether the streaming distribution is enabled to accept end user requests for content.

Return type dict

Returns

Response Syntax

```
{
  'StreamingDistribution': {
    'Id': 'string',
    'Status': 'string',
    'LastModifiedTime': datetime(2015, 1, 1),
    'DomainName': 'string',
    'ActiveTrustedSigners': {
      'Enabled': True|False,
      'Quantity': 123,
      'Items': [
        {
          'AwsAccountNumber': 'string',
          'KeyPairIds': {
            'Quantity': 123,
            'Items': [
              'string',
            ]
          }
        }
      ]
    },
  },
  'StreamingDistributionConfig': {
    'CallerReference': 'string',
    'S3Origin': {
      'DomainName': 'string',
      'OriginAccessIdentity': 'string'
    },
    'Aliases': {
```

```

        'Quantity': 123,
        'Items': [
            'string',
        ]
    },
    'Comment': 'string',
    'Logging': {
        'Enabled': True|False,
        'Bucket': 'string',
        'Prefix': 'string'
    },
    'TrustedSigners': {
        'Enabled': True|False,
        'Quantity': 123,
        'Items': [
            'string',
        ]
    },
    'PriceClass': 'PriceClass_100'|'PriceClass_200'|'PriceClass_All',
    'Enabled': True|False
}
},
'Location': 'string',
'ETag': 'string'
}

```

Response Structure

- (*dict*) – The returned result of the corresponding request.
 - **StreamingDistribution** (*dict*) – The streaming distribution’s information.
 - * **Id** (*string*) – The identifier for the streaming distribution. For example: EGTXBBD79H29TRA8.
 - * **Status** (*string*) – The current status of the streaming distribution. When the status is Deployed, the distribution’s information is fully propagated throughout the Amazon CloudFront system.
 - * **LastModifiedTime** (*datetime*) – The date and time the distribution was last modified.
 - * **DomainName** (*string*) – The domain name corresponding to the streaming distribution. For example: s5c39gqb8ow64r.cloudfront.net.
 - * **ActiveTrustedSigners** (*dict*) – CloudFront automatically adds this element to the response only if you’ve set up the distribution to serve private content with signed URLs. The element lists the key pair IDs that CloudFront is aware of for each trusted signer. The Signer child element lists the AWS account number of the trusted signer (or an empty Self element if the signer is you). The Signer element also includes the IDs of any active key pairs associated with the trusted signer’s AWS account. If no KeyPairId element appears for a Signer, that signer can’t create working signed URLs.
 - **Enabled** (*boolean*) – Each active trusted signer.
 - **Quantity** (*integer*) – The number of unique trusted signers included in all cache behaviors. For example, if three cache behaviors all list the same three AWS accounts, the value of Quantity for ActiveTrustedSigners will be 3.
 - **Items** (*list*) – A complex type that contains one Signer complex type for each unique trusted signer that is specified in the TrustedSigners complex type, including trusted signers in the

- default cache behavior and in all of the other cache behaviors.
- *(dict)* – A complex type that lists the AWS accounts that were included in the TrustedSigners complex type, as well as their active CloudFront key pair IDs, if any.
- **AwsAccountNumber** (*string*) – Specifies an AWS account that can create signed URLs. Values: self, which indicates that the AWS account that was used to create the distribution can create signed URLs, or an AWS account number. Omit the dashes in the account number.
- **KeyPairIds** (*dict*) – A complex type that lists the active CloudFront key pairs, if any, that are associated with AwsAccountNumber.
- **Quantity** (*integer*) – The number of active CloudFront key pairs for AwsAccountNumber.
- **Items** (*list*) – A complex type that lists the active CloudFront key pairs, if any, that are associated with AwsAccountNumber.
- (*string*) –
- * **StreamingDistributionConfig** (*dict*) – The current configuration information for the streaming distribution.
 - **CallerReference** (*string*) – A unique number that ensures the request can't be replayed. If the CallerReference is new (no matter the content of the StreamingDistributionConfig object), a new streaming distribution is created. If the CallerReference is a value you already sent in a previous request to create a streaming distribution, and the content of the StreamingDistributionConfig is identical to the original request (ignoring white space), the response includes the same information returned to the original request. If the CallerReference is a value you already sent in a previous request to create a streaming distribution but the content of the StreamingDistributionConfig is different from the original request, CloudFront returns a DistributionAlreadyExists error.
 - **S3Origin** (*dict*) – A complex type that contains information about the Amazon S3 bucket from which you want CloudFront to get your media files for distribution.
 - **DomainName** (*string*) – The DNS name of the S3 origin.
 - **OriginAccessIdentity** (*string*) – Your S3 origin's origin access identity.
 - **Aliases** (*dict*) – A complex type that contains information about CNAMEs (alternate domain names), if any, for this streaming distribution.
 - **Quantity** (*integer*) – The number of CNAMEs, if any, for this distribution.
 - **Items** (*list*) – Optional: A complex type that contains CNAME elements, if any, for this distribution. If Quantity is 0, you can omit Items.
 - (*string*) –
 - **Comment** (*string*) – Any comments you want to include about the streaming distribution.
 - **Logging** (*dict*) – A complex type that controls whether access logs are written for the streaming distribution.
 - **Enabled** (*boolean*) – Specifies whether you want CloudFront to save access logs to an Amazon S3 bucket. If you do not

want to enable logging when you create a streaming distribution or if you want to disable logging for an existing streaming distribution, specify `false` for `Enabled`, and specify empty `Bucket` and `Prefix` elements. If you specify `false` for `Enabled` but you specify values for `Bucket` and `Prefix`, the values are automatically deleted.

- **Bucket** (*string*) – The Amazon S3 bucket to store the access logs in, for example, `myawslogbucket.s3.amazonaws.com`.
- **Prefix** (*string*) – An optional string that you want CloudFront to prefix to the access log filenames for this streaming distribution, for example, `myprefix/`. If you want to enable logging, but you do not want to specify a prefix, you still must include an empty `Prefix` element in the `Logging` element.
- **TrustedSigners** (*dict*) – A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the `PathPattern` for this cache behavior, specify `true` for `Enabled`, and specify the applicable values for `Quantity` and `Items`. For more information, go to [Using a Signed URL to Serve Private Content](#) in the Amazon CloudFront Developer Guide. If you don't want to require signed URLs in requests for objects that match `PathPattern`, specify `false` for `Enabled` and `0` for `Quantity`. Omit `Items`. To add, change, or remove one or more trusted signers, change `Enabled` to `true` (if it's currently `false`), change `Quantity` as applicable, and specify all of the trusted signers that you want to include in the updated distribution.
- **Enabled** (*boolean*) – Specifies whether you want to require end users to use signed URLs to access the files specified by `PathPattern` and `TargetOriginId`.
- **Quantity** (*integer*) – The number of trusted signers for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If `Quantity` is `0`, you can omit `Items`.
- (*string*) –
- **PriceClass** (*string*) – A complex type that contains information about price class for this streaming distribution.
- **Enabled** (*boolean*) – Whether the streaming distribution is enabled to accept end user requests for content.
- **Location** (*string*) – The fully qualified URI of the new streaming distribution resource just created. For example: <https://cloudfront.amazonaws.com/2010-11-01/streaming-distribution/EGTXBD79H29TRA8>.
- **ETag** (*string*) – The current version of the streaming distribution created.

delete_cloud_front_origin_access_identity (**kwargs)

Delete an origin access identity.

Request Syntax

```
response = client.delete_cloud_front_origin_access_identity(
    Id='string',
    IfMatch='string'
)
```

Parameters

- **Id** (*string*) – **[REQUIRED]** The origin access identity's id.
- **IfMatch** (*string*) – The value of the ETag header you received from a previous GET or PUT request. For example: E2QWRUHAPOMQZL.

Returns None

delete_distribution (***kwargs*)

Delete a distribution.

Request Syntax

```
response = client.delete_distribution(  
    Id='string',  
    IfMatch='string'  
)
```

Parameters

- **Id** (*string*) – **[REQUIRED]** The distribution id.
- **IfMatch** (*string*) – The value of the ETag header you received when you disabled the distribution. For example: E2QWRUHAPOMQZL.

Returns None

delete_streaming_distribution (***kwargs*)

Delete a streaming distribution.

Request Syntax

```
response = client.delete_streaming_distribution(  
    Id='string',  
    IfMatch='string'  
)
```

Parameters

- **Id** (*string*) – **[REQUIRED]** The distribution id.
- **IfMatch** (*string*) – The value of the ETag header you received when you disabled the streaming distribution. For example: E2QWRUHAPOMQZL.

Returns None

generate_presigned_url (*ClientMethod, Params=None, ExpiresIn=3600, HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to *ClientMethod*.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_cloud_front_origin_access_identity (***kwargs*)

Get the information about an origin access identity.

Request Syntax

```
response = client.get_cloud_front_origin_access_identity(  
    Id='string'  
)
```

Parameters **Id** (*string*) – **[REQUIRED]** The identity's id.

Return type dict

Returns

Response Syntax

```
{
  'CloudFrontOriginAccessIdentity': {
    'Id': 'string',
    'S3CanonicalUserId': 'string',
    'CloudFrontOriginAccessIdentityConfig': {
      'CallerReference': 'string',
      'Comment': 'string'
    }
  },
  'ETag': 'string'
}
```

Response Structure

- (*dict*) – The returned result of the corresponding request.
- **CloudFrontOriginAccessIdentity** (*dict*) – The origin access identity’s information.
 - * **Id** (*string*) – The ID for the origin access identity. For example: E74FTE3AJFJ256A.
 - * **S3CanonicalUserId** (*string*) – The Amazon S3 canonical user ID for the origin access identity, which you use when giving the origin access identity read permission to an object in Amazon S3.
 - * **CloudFrontOriginAccessIdentityConfig** (*dict*) – The current configuration information for the identity.
 - **CallerReference** (*string*) – A unique number that ensures the request can’t be replayed. If the CallerReference is new (no matter the content of the CloudFrontOriginAccessIdentityConfig object), a new origin access identity is created. If the CallerReference is a value you already sent in a previous request to create an identity, and the content of the CloudFrontOriginAccessIdentityConfig is identical to the original request (ignoring white space), the response includes the same information returned to the original request. If the CallerReference is a value you already sent in a previous request to create an identity but the content of the CloudFrontOriginAccessIdentityConfig is different from the original request, CloudFront returns a CloudFrontOriginAccessIdentityAlreadyExists error.
 - **Comment** (*string*) – Any comments you want to include about the origin access identity.
- **ETag** (*string*) – The current version of the origin access identity’s information. For example: E2QWRUHAPOMQZL.

get_cloud_front_origin_access_identity_config (***kwargs*)

Get the configuration information about an origin access identity.

Request Syntax

```
response = client.get_cloud_front_origin_access_identity_config(
    Id='string'
)
```

Parameters **Id** (*string*) – [REQUIRED] The identity’s id.

Return type dict

Returns

Response Syntax

```
{
    'CloudFrontOriginAccessIdentityConfig': {
        'CallerReference': 'string',
        'Comment': 'string'
    },
    'ETag': 'string'
}
```

Response Structure

- (*dict*) – The returned result of the corresponding request.
 - **CloudFrontOriginAccessIdentityConfig** (*dict*) – The origin access identity’s configuration information.
 - * **CallerReference** (*string*) – A unique number that ensures the request can’t be replayed. If the CallerReference is new (no matter the content of the CloudFrontOriginAccessIdentityConfig object), a new origin access identity is created. If the CallerReference is a value you already sent in a previous request to create an identity, and the content of the CloudFrontOriginAccessIdentityConfig is identical to the original request (ignoring white space), the response includes the same information returned to the original request. If the CallerReference is a value you already sent in a previous request to create an identity but the content of the CloudFrontOriginAccessIdentityConfig is different from the original request, CloudFront returns a CloudFrontOriginAccessIdentityAlreadyExists error.
 - * **Comment** (*string*) – Any comments you want to include about the origin access identity.
 - **ETag** (*string*) – The current version of the configuration. For example: E2QWRUHAPOMQZL.

get_distribution (***kwargs*)

Get the information about a distribution.

Request Syntax

```
response = client.get_distribution(
    Id='string'
)
```

Parameters **Id** (*string*) – [REQUIRED] The distribution’s id.

Return type dict

Returns

Response Syntax

```
{
    'Distribution': {
        'Id': 'string',
        'Status': 'string',
        'LastModifiedTime': datetime(2015, 1, 1),
        'InProgressInvalidationBatches': 123,
        'DomainName': 'string',
        'ActiveTrustedSigners': {
```



```

'Enabled': True|False,
'Quantity': 123,
'Items': [
    {
        'AwsAccountNumber': 'string',
        'KeyPairIds': {
            'Quantity': 123,
            'Items': [
                'string',
            ]
        },
    },
]
},
'DistributionConfig': {
    'CallerReference': 'string',
    'Aliases': {
        'Quantity': 123,
        'Items': [
            'string',
        ]
    },
    'DefaultRootObject': 'string',
    'Origins': {
        'Quantity': 123,
        'Items': [
            {
                'Id': 'string',
                'DomainName': 'string',
                'OriginPath': 'string',
                'S3OriginConfig': {
                    'OriginAccessIdentity': 'string'
                },
                'CustomOriginConfig': {
                    'HTTPPort': 123,
                    'HTTPSPort': 123,
                    'OriginProtocolPolicy': 'http-only'|'match-viewer'
                }
            },
        ],
    },
    'DefaultCacheBehavior': {
        'TargetOriginId': 'string',
        'ForwardedValues': {
            'QueryString': True|False,
            'Cookies': {
                'Forward': 'none'|'whitelist'|'all',
                'WhitelistedNames': {
                    'Quantity': 123,
                    'Items': [
                        'string',
                    ]
                }
            },
        },
        'Headers': {
            'Quantity': 123,
            'Items': [
                'string',
            ],
        },
    },
},

```

```
        ],
    },
    'TrustedSigners': {
        'Enabled': True|False,
        'Quantity': 123,
        'Items': [
            'string',
        ]
    },
    'ViewerProtocolPolicy': 'allow-all'|'https-only'|'redirect-to-http',
    'MinTTL': 123,
    'AllowedMethods': {
        'Quantity': 123,
        'Items': [
            'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
        ],
        'CachedMethods': {
            'Quantity': 123,
            'Items': [
                'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
            ]
        }
    },
    'SmoothStreaming': True|False,
    'DefaultTTL': 123,
    'MaxTTL': 123
},
'CacheBehaviors': {
    'Quantity': 123,
    'Items': [
        {
            'PathPattern': 'string',
            'TargetOriginId': 'string',
            'ForwardedValues': {
                'QueryString': True|False,
                'Cookies': {
                    'Forward': 'none'|'whitelist'|'all',
                    'WhitelistedNames': {
                        'Quantity': 123,
                        'Items': [
                            'string',
                        ]
                    }
                },
            },
            'Headers': {
                'Quantity': 123,
                'Items': [
                    'string',
                ]
            }
        },
    ],
    'TrustedSigners': {
        'Enabled': True|False,
        'Quantity': 123,
        'Items': [
            'string',
        ]
    }
}
```

3.1. Services 211

```
    'ETag': 'string'  
}
```

Response Structure

- (*dict*) – The returned result of the corresponding request.
 - **Distribution** (*dict*) – The distribution’s information.
 - * **Id** (*string*) – The identifier for the distribution. For example: EDFD-VBD632BHDS5.
 - * **Status** (*string*) – This response element indicates the current status of the distribution. When the status is Deployed, the distribution’s information is fully propagated throughout the Amazon CloudFront system.
 - * **LastModifiedTime** (*datetime*) – The date and time the distribution was last modified.
 - * **InProgressInvalidationBatches** (*integer*) – The number of invalidation batches currently in progress.
 - * **DomainName** (*string*) – The domain name corresponding to the distribution. For example: d604721fxaaqy9.cloudfront.net.
 - * **ActiveTrustedSigners** (*dict*) – CloudFront automatically adds this element to the response only if you’ve set up the distribution to serve private content with signed URLs. The element lists the key pair IDs that CloudFront is aware of for each trusted signer. The Signer child element lists the AWS account number of the trusted signer (or an empty Self element if the signer is you). The Signer element also includes the IDs of any active key pairs associated with the trusted signer’s AWS account. If no KeyPairId element appears for a Signer, that signer can’t create working signed URLs.
 - **Enabled** (*boolean*) – Each active trusted signer.
 - **Quantity** (*integer*) – The number of unique trusted signers included in all cache behaviors. For example, if three cache behaviors all list the same three AWS accounts, the value of Quantity for ActiveTrustedSigners will be 3.
 - **Items** (*list*) – A complex type that contains one Signer complex type for each unique trusted signer that is specified in the TrustedSigners complex type, including trusted signers in the default cache behavior and in all of the other cache behaviors.
 - (*dict*) – A complex type that lists the AWS accounts that were included in the TrustedSigners complex type, as well as their active CloudFront key pair IDs, if any.
 - **AwsAccountNumber** (*string*) – Specifies an AWS account that can create signed URLs. Values: self, which indicates that the AWS account that was used to create the distribution can create signed URLs, or an AWS account number. Omit the dashes in the account number.
 - **KeyPairIds** (*dict*) – A complex type that lists the active CloudFront key pairs, if any, that are associated with AwsAccountNumber.
 - **Quantity** (*integer*) – The number of active CloudFront key pairs for AwsAccountNumber.
 - **Items** (*list*) – A complex type that lists the active CloudFront key pairs, if any, that are associated with AwsAccountNumber.
 - (*string*) –
 - * **DistributionConfig** (*dict*) – The current configuration information

for the distribution.

- **CallerReference** (*string*) – A unique number that ensures the request can't be replayed. If the CallerReference is new (no matter the content of the DistributionConfig object), a new distribution is created. If the CallerReference is a value you already sent in a previous request to create a distribution, and the content of the DistributionConfig is identical to the original request (ignoring white space), the response includes the same information returned to the original request. If the CallerReference is a value you already sent in a previous request to create a distribution but the content of the DistributionConfig is different from the original request, CloudFront returns a DistributionAlreadyExists error.
- **Aliases** (*dict*) – A complex type that contains information about CNAMEs (alternate domain names), if any, for this distribution.
- **Quantity** (*integer*) – The number of CNAMEs, if any, for this distribution.
- **Items** (*list*) – Optional: A complex type that contains CNAME elements, if any, for this distribution. If Quantity is 0, you can omit Items.
- (*string*) –
- **DefaultRootObject** (*string*) – The object that you want CloudFront to return (for example, index.html) when an end user requests the root URL for your distribution (<http://www.example.com>) instead of an object in your distribution (<http://www.example.com/index.html>). Specifying a default root object avoids exposing the contents of your distribution. If you don't want to specify a default root object when you create a distribution, include an empty DefaultRootObject element. To delete the default root object from an existing distribution, update the distribution configuration and include an empty DefaultRootObject element. To replace the default root object, update the distribution configuration and specify the new object.
- **Origins** (*dict*) – A complex type that contains information about origins for this distribution.
- **Quantity** (*integer*) – The number of origins for this distribution.
- **Items** (*list*) – A complex type that contains origins for this distribution.
- (*dict*) – A complex type that describes the Amazon S3 bucket or the HTTP server (for example, a web server) from which CloudFront gets your files. You must create at least one origin.
- **Id** (*string*) – A unique identifier for the origin. The value of Id must be unique within the distribution. You use the value of Id when you create a cache behavior. The Id identifies the origin that CloudFront routes a request to when the request matches the path pattern for that cache behavior.
- **DomainName** (*string*) – Amazon S3 origins: The DNS name of the Amazon S3 bucket from which you want CloudFront to get objects for this origin, for example, myaws-bucket.s3.amazonaws.com. Custom origins: The DNS domain name for the HTTP server from which you want

CloudFront to get objects for this origin, for example, `www.example.com`.

- **OriginPath** (*string*) – An optional element that causes CloudFront to request your content from a directory in your Amazon S3 bucket or your custom origin. When you include the OriginPath element, specify the directory name, beginning with a `/`. CloudFront appends the directory name to the value of DomainName.
- **S3OriginConfig** (*dict*) – A complex type that contains information about the Amazon S3 origin. If the origin is a custom origin, use the CustomOriginConfig element instead.
- **OriginAccessIdentity** (*string*) – The CloudFront origin access identity to associate with the origin. Use an origin access identity to configure the origin so that end users can only access objects in an Amazon S3 bucket through CloudFront. If you want end users to be able to access objects using either the CloudFront URL or the Amazon S3 URL, specify an empty OriginAccessIdentity element. To delete the origin access identity from an existing distribution, update the distribution configuration and include an empty OriginAccessIdentity element. To replace the origin access identity, update the distribution configuration and specify the new origin access identity. Use the format `origin-access-identity/cloudfront/Id` where Id is the value that CloudFront returned in the Id element when you created the origin access identity.
- **CustomOriginConfig** (*dict*) – A complex type that contains information about a custom origin. If the origin is an Amazon S3 bucket, use the S3OriginConfig element instead.
- **HTTPPort** (*integer*) – The HTTP port the custom origin listens on.
- **HTTPSPort** (*integer*) – The HTTPS port the custom origin listens on.
- **OriginProtocolPolicy** (*string*) – The origin protocol policy to apply to your origin.
- **DefaultCacheBehavior** (*dict*) – A complex type that describes the default cache behavior if you do not specify a CacheBehavior element or if files don't match any of the values of PathPattern in CacheBehavior elements. You must create exactly one default cache behavior.
- **TargetOriginId** (*string*) – The value of ID for the origin that you want CloudFront to route requests to when a request matches the path pattern either for a cache behavior or for the default cache behavior.
- **ForwardedValues** (*dict*) – A complex type that specifies how CloudFront handles query strings, cookies and headers.
- **QueryString** (*boolean*) – Indicates whether you want CloudFront to forward query strings to the origin that is associated with this cache behavior. If so, specify `true`; if not, specify `false`.
- **Cookies** (*dict*) – A complex type that specifies how CloudFront handles cookies.
- **Forward** (*string*) – Use this element to specify whether you want CloudFront to forward cookies to the origin that is associated with this cache behavior. You can specify `all`, `none` or

whitelist. If you choose All, CloudFront forwards all cookies regardless of how many your application uses.

- **WhitelistedNames** (*dict*) – A complex type that specifies the whitelisted cookies, if any, that you want CloudFront to forward to your origin that is associated with this cache behavior.
- **Quantity** (*integer*) – The number of whitelisted cookies for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains whitelisted cookies for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **Headers** (*dict*) – A complex type that specifies the Headers, if any, that you want CloudFront to vary upon for this cache behavior.
- **Quantity** (*integer*) – The number of different headers that you want CloudFront to forward to the origin and to vary on for this cache behavior. The maximum number of headers that you can specify by name is 10. If you want CloudFront to forward all headers to the origin and vary on all of them, specify 1 for Quantity and * for Name. If you don't want CloudFront to forward any additional headers to the origin or to vary on any headers, specify 0 for Quantity and omit Items.
- **Items** (*list*) – Optional: A complex type that contains a Name element for each header that you want CloudFront to forward to the origin and to vary on for this cache behavior. If Quantity is 0, omit Items.
- (*string*) –
- **TrustedSigners** (*dict*) – A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the PathPattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don't want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more trusted signers, change Enabled to true (if it's currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.
- **Enabled** (*boolean*) – Specifies whether you want to require end users to use signed URLs to access the files specified by PathPattern and TargetOriginId.
- **Quantity** (*integer*) – The number of trusted signers for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **ViewerProtocolPolicy** (*string*) – Use this element to specify the protocol that users can use to access the files in the origin specified by TargetOriginId when a request matches the path pattern in PathPattern. If you want CloudFront to allow end

users to use any available protocol, specify allow-all. If you want CloudFront to require HTTPS, specify https. If you want CloudFront to respond to an HTTP request with an HTTP status code of 301 (Moved Permanently) and the HTTPS URL, specify redirect-to-https. The viewer then resubmits the request using the HTTPS URL.

- **MinTTL** (*integer*) – The minimum amount of time that you want objects to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **AllowedMethods** (*dict*) – A complex type that controls which HTTP methods CloudFront processes and forwards to your Amazon S3 bucket or your custom origin. There are three choices:
 - CloudFront forwards only GET and HEAD requests.
 - CloudFront forwards only GET, HEAD and OPTIONS requests.
 - CloudFront forwards GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests. If you pick the third choice, you may need to restrict access to your Amazon S3 bucket or to your custom origin so users can't perform operations that you don't want them to. For example, you may not want users to have permission to delete objects from your origin.
- **Quantity** (*integer*) – The number of HTTP methods that you want CloudFront to forward to your origin. Valid values are 2 (for GET and HEAD requests), 3 (for GET, HEAD and OPTIONS requests) and 7 (for GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests).
- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to process and forward to your origin.
- (*string*) –
- **CachedMethods** (*dict*) – A complex type that controls whether CloudFront caches the response to requests using the specified HTTP methods. There are two choices:
 - CloudFront caches responses to GET and HEAD requests.
 - CloudFront caches responses to GET, HEAD, and OPTIONS requests. If you pick the second choice for your S3 Origin, you may need to forward Access-Control-Request-Method, Access-Control-Request-Headers and Origin headers for the responses to be cached correctly.
- **Quantity** (*integer*) – The number of HTTP methods for which you want CloudFront to cache responses. Valid values are 2 (for caching responses to GET and HEAD requests) and 3 (for caching responses to GET, HEAD, and OPTIONS requests).
- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to cache responses to.
- (*string*) –
- **SmoothStreaming** (*boolean*) – Indicates whether you want to distribute media files in Microsoft Smooth Streaming format using the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
- **DefaultTTL** (*integer*) – If you don't configure your origin to add a Cache-Control max-age directive or an Expires header,

DefaultTTL is the default amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin does not add HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).

- **MaxTTL** (*integer*) – The maximum amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin adds HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **CacheBehaviors** (*dict*) – A complex type that contains zero or more CacheBehavior elements.
- **Quantity** (*integer*) – The number of cache behaviors for this distribution.
- **Items** (*list*) – Optional: A complex type that contains cache behaviors for this distribution. If Quantity is 0, you can omit Items.
- (*dict*) – A complex type that describes how CloudFront processes requests. You can create up to 10 cache behaviors. You must create at least as many cache behaviors (including the default cache behavior) as you have origins if you want CloudFront to distribute objects from all of the origins. Each cache behavior specifies the one origin from which you want CloudFront to get objects. If you have two origins and only the default cache behavior, the default cache behavior will cause CloudFront to get objects from one of the origins, but the other origin will never be used. If you don't want to specify any cache behaviors, include only an empty CacheBehaviors element. Don't include an empty CacheBehavior element, or CloudFront returns a MalformedXML error. To delete all cache behaviors in an existing distribution, update the distribution configuration and include only an empty CacheBehaviors element. To add, change, or remove one or more cache behaviors, update the distribution configuration and specify all of the cache behaviors that you want to include in the updated distribution.
- **PathPattern** (*string*) – The pattern (for example, images/* .jpg) that specifies which requests you want this cache behavior to apply to. When CloudFront receives an end-user request, the requested path is compared with path patterns in the order in which cache behaviors are listed in the distribution. The path pattern for the default cache behavior is * and cannot be changed. If the request for an object does not match the path pattern for any cache behaviors, CloudFront applies the behavior in the default cache behavior.
- **TargetOriginId** (*string*) – The value of ID for the origin that you want CloudFront to route requests to when a request matches the path pattern either for a cache behavior or for

the default cache behavior.

- **ForwardedValues** (*dict*) – A complex type that specifies how CloudFront handles query strings, cookies and headers.
- **QueryString** (*boolean*) – Indicates whether you want CloudFront to forward query strings to the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
- **Cookies** (*dict*) – A complex type that specifies how CloudFront handles cookies.
- **Forward** (*string*) – Use this element to specify whether you want CloudFront to forward cookies to the origin that is associated with this cache behavior. You can specify all, none or whitelist. If you choose All, CloudFront forwards all cookies regardless of how many your application uses.
- **WhitelistedNames** (*dict*) – A complex type that specifies the whitelisted cookies, if any, that you want CloudFront to forward to your origin that is associated with this cache behavior.
- **Quantity** (*integer*) – The number of whitelisted cookies for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains whitelisted cookies for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **Headers** (*dict*) – A complex type that specifies the Headers, if any, that you want CloudFront to vary upon for this cache behavior.
- **Quantity** (*integer*) – The number of different headers that you want CloudFront to forward to the origin and to vary on for this cache behavior. The maximum number of headers that you can specify by name is 10. If you want CloudFront to forward all headers to the origin and vary on all of them, specify 1 for Quantity and * for Name. If you don't want CloudFront to forward any additional headers to the origin or to vary on any headers, specify 0 for Quantity and omit Items.
- **Items** (*list*) – Optional: A complex type that contains a Name element for each header that you want CloudFront to forward to the origin and to vary on for this cache behavior. If Quantity is 0, omit Items.
- (*string*) –
- **TrustedSigners** (*dict*) – A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the PathPattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don't want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more trusted signers, change Enabled to true (if it's currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.
- **Enabled** (*boolean*) – Specifies whether you want to require

end users to use signed URLs to access the files specified by PathPattern and TargetOriginId.

- **Quantity** (*integer*) – The number of trusted signers for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **ViewerProtocolPolicy** (*string*) – Use this element to specify the protocol that users can use to access the files in the origin specified by TargetOriginId when a request matches the path pattern in PathPattern. If you want CloudFront to allow end users to use any available protocol, specify allow-all. If you want CloudFront to require HTTPS, specify https. If you want CloudFront to respond to an HTTP request with an HTTP status code of 301 (Moved Permanently) and the HTTPS URL, specify redirect-to-https. The viewer then resubmits the request using the HTTPS URL.
- **MinTTL** (*integer*) – The minimum amount of time that you want objects to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **AllowedMethods** (*dict*) – A complex type that controls which HTTP methods CloudFront processes and forwards to your Amazon S3 bucket or your custom origin. There are three choices:
 - CloudFront forwards only GET and HEAD requests.
 - CloudFront forwards only GET, HEAD and OPTIONS requests.
 - CloudFront forwards GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests.
 If you pick the third choice, you may need to restrict access to your Amazon S3 bucket or to your custom origin so users can't perform operations that you don't want them to. For example, you may not want users to have permission to delete objects from your origin.
- **Quantity** (*integer*) – The number of HTTP methods that you want CloudFront to forward to your origin. Valid values are 2 (for GET and HEAD requests), 3 (for GET, HEAD and OPTIONS requests) and 7 (for GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests).
- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to process and forward to your origin.
- (*string*) –
- **CachedMethods** (*dict*) – A complex type that controls whether CloudFront caches the response to requests using the specified HTTP methods. There are two choices:
 - CloudFront caches responses to GET and HEAD requests.
 - CloudFront caches responses to GET, HEAD, and OPTIONS requests.
 If you pick the second choice for your S3 Origin, you may need to forward Access-Control-Request-Method, Access-Control-Request-Headers and Origin headers for the responses to be cached correctly.
- **Quantity** (*integer*) – The number of HTTP methods for which

you want CloudFront to cache responses. Valid values are 2 (for caching responses to GET and HEAD requests) and 3 (for caching responses to GET, HEAD, and OPTIONS requests).

- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to cache responses to.
- (*string*) –
- **SmoothStreaming** (*boolean*) – Indicates whether you want to distribute media files in Microsoft Smooth Streaming format using the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
- **DefaultTTL** (*integer*) – If you don't configure your origin to add a Cache-Control max-age directive or an Expires header, DefaultTTL is the default amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin does not add HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **MaxTTL** (*integer*) – The maximum amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin adds HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **CustomErrorResponses** (*dict*) – A complex type that contains zero or more CustomErrorResponse elements.
- **Quantity** (*integer*) – The number of custom error responses for this distribution.
- **Items** (*list*) – Optional: A complex type that contains custom error responses for this distribution. If Quantity is 0, you can omit Items.
- (*dict*) – A complex type that describes how you'd prefer CloudFront to respond to requests that result in either a 4xx or 5xx response. You can control whether a custom error page should be displayed, what the desired response code should be for this error page and how long should the error response be cached by CloudFront. If you don't want to specify any custom error responses, include only an empty CustomErrorResponses element. To delete all custom error responses in an existing distribution, update the distribution configuration and include only an empty CustomErrorResponses element. To add, change, or remove one or more custom error responses, update the distribution configuration and specify all of the custom error responses that you want to include in the updated distribution.
- **ErrorCode** (*integer*) – The 4xx or 5xx HTTP status code that you want to customize. For a list of HTTP status codes that you can customize, see CloudFront documentation.
- **ResponsePagePath** (*string*) – The path of the custom error page (for example, /custom_404.html). The path is relative

to the distribution and must begin with a slash (/). If the path includes any non-ASCII characters or unsafe characters as defined in RFC 1783 (<http://www.ietf.org/rfc/rfc1738.txt>), URL encode those characters. Do not URL encode any other characters in the path, or CloudFront will not return the custom error page to the viewer.

- **ResponseCode** (*string*) – The HTTP status code that you want CloudFront to return with the custom error page to the viewer. For a list of HTTP status codes that you can replace, see CloudFront Documentation.
- **ErrorCachingMinTTL** (*integer*) – The minimum amount of time you want HTTP error codes to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 31,536,000.
- **Comment** (*string*) – Any comments you want to include about the distribution.
- **Logging** (*dict*) – A complex type that controls whether access logs are written for the distribution.
- **Enabled** (*boolean*) – Specifies whether you want CloudFront to save access logs to an Amazon S3 bucket. If you do not want to enable logging when you create a distribution or if you want to disable logging for an existing distribution, specify false for Enabled, and specify empty Bucket and Prefix elements. If you specify false for Enabled but you specify values for Bucket, prefix and IncludeCookies, the values are automatically deleted.
- **IncludeCookies** (*boolean*) – Specifies whether you want CloudFront to include cookies in access logs, specify true for IncludeCookies. If you choose to include cookies in logs, CloudFront logs all cookies regardless of how you configure the cache behaviors for this distribution. If you do not want to include cookies when you create a distribution or if you want to disable include cookies for an existing distribution, specify false for IncludeCookies.
- **Bucket** (*string*) – The Amazon S3 bucket to store the access logs in, for example, myawslogbucket.s3.amazonaws.com.
- **Prefix** (*string*) – An optional string that you want CloudFront to prefix to the access log filenames for this distribution, for example, myprefix/. If you want to enable logging, but you do not want to specify a prefix, you still must include an empty Prefix element in the Logging element.
- **PriceClass** (*string*) – A complex type that contains information about price class for this distribution.
- **Enabled** (*boolean*) – Whether the distribution is enabled to accept end user requests for content.
- **ViewerCertificate** (*dict*) – A complex type that contains information about viewer certificates for this distribution.
- **IAMCertificateId** (*string*) – If you want viewers to use HTTPS to request your objects and you're using an alternate domain name in your object URLs (for example, <https://example.com/logo.jpg>), specify the IAM certificate identifier of the custom viewer certificate for this distribution. Specify either this value or CloudFrontDefaultCertificate.

- **CloudFrontDefaultCertificate** (*boolean*) – If you want viewers to use HTTPS to request your objects and you’re using the CloudFront domain name of your distribution in your object URLs (for example, <https://d1111111abcdef8.cloudfront.net/logo.jpg>), set to true. Omit this value if you are setting an IAMCertificateId.
- **SSLSupportMethod** (*string*) – If you specify a value for IAMCertificateId, you must also specify how you want CloudFront to serve HTTPS requests. Valid values are vip and sni-only. If you specify vip, CloudFront uses dedicated IP addresses for your content and can respond to HTTPS requests from any viewer. However, you must request permission to use this feature, and you incur additional monthly charges. If you specify sni-only, CloudFront can only respond to HTTPS requests from viewers that support Server Name Indication (SNI). All modern browsers support SNI, but some browsers still in use don’t support SNI. Do not specify a value for SSLSupportMethod if you specified true for CloudFrontDefaultCertificate.
- **MinimumProtocolVersion** (*string*) – Specify the minimum version of the SSL protocol that you want CloudFront to use, SSLv3 or TLSv1, for HTTPS connections. CloudFront will serve your objects only to browsers or devices that support at least the SSL version that you specify. The TLSv1 protocol is more secure, so we recommend that you specify SSLv3 only if your users are using browsers or devices that don’t support TLSv1. If you’re using a custom certificate (if you specify a value for IAMCertificateId) and if you’re using dedicated IP (if you specify vip for SSLSupportMethod), you can choose SSLv3 or TLSv1 as the MinimumProtocolVersion. If you’re using a custom certificate (if you specify a value for IAMCertificateId) and if you’re using SNI (if you specify sni-only for SSLSupportMethod), you must specify TLSv1 for MinimumProtocolVersion.
- **Restrictions** (*dict*) – A complex type that identifies ways in which you want to restrict distribution of your content.
- **GeoRestriction** (*dict*) – A complex type that controls the countries in which your content is distributed. For more information about geo restriction, go to Customizing Error Responses in the Amazon CloudFront Developer Guide. CloudFront determines the location of your users using MaxMind GeoIP databases. For information about the accuracy of these databases, see How accurate are your GeoIP databases? on the MaxMind website.
- **RestrictionType** (*string*) – The method that you want to use to restrict distribution of your content by country: - none: No geo restriction is enabled, meaning access to content is not restricted by client geo location. - blacklist: The Location elements specify the countries in which you do not want CloudFront to distribute your content. - whitelist: The Location elements specify the countries in which you want CloudFront to distribute your content.
- **Quantity** (*integer*) – When geo restriction is enabled, this is the number of countries in your whitelist or blacklist. Other-

wise, when it is not enabled, Quantity is 0, and you can omit Items.

- **Items** (*list*) – A complex type that contains a Location element for each country in which you want CloudFront either to distribute your content (whitelist) or not distribute your content (blacklist). The Location element is a two-letter, uppercase country code for a country that you want to include in your blacklist or whitelist. Include one Location element for each country. CloudFront and MaxMind both use ISO 3166 country codes. For the current list of countries and the corresponding codes, see ISO 3166-1-alpha-2 code on the International Organization for Standardization website. You can also refer to the country list in the CloudFront console, which includes both country names and codes.
- (*string*) –
- **ETag** (*string*) – The current version of the distribution’s information. For example: E2QWRUHAPOMQZL.

get_distribution_config (**kwargs)

Get the configuration information about a distribution.

Request Syntax

```
response = client.get_distribution_config(
    Id='string'
)
```

Parameters **Id** (*string*) – **[REQUIRED]** The distribution’s id.

Return type dict

Returns

Response Syntax

```
{
  'DistributionConfig': {
    'CallerReference': 'string',
    'Aliases': {
      'Quantity': 123,
      'Items': [
        'string',
      ]
    },
    'DefaultRootObject': 'string',
    'Origins': {
      'Quantity': 123,
      'Items': [
        {
          'Id': 'string',
          'DomainName': 'string',
          'OriginPath': 'string',
          'S3OriginConfig': {
            'OriginAccessIdentity': 'string'
          },
          'CustomOriginConfig': {
            'HTTPPort': 123,
            'HTTPSPort': 123,
            'OriginProtocolPolicy': 'http-only'|'match-viewer'
```

```
    },
    ],
},
'DefaultCacheBehavior': {
    'TargetOriginId': 'string',
    'ForwardedValues': {
        'QueryString': True|False,
        'Cookies': {
            'Forward': 'none'|'whitelist'|'all',
            'WhitelistedNames': {
                'Quantity': 123,
                'Items': [
                    'string',
                ]
            }
        },
    },
    'Headers': {
        'Quantity': 123,
        'Items': [
            'string',
        ]
    },
},
'TrustedSigners': {
    'Enabled': True|False,
    'Quantity': 123,
    'Items': [
        'string',
    ]
},
'ViewerProtocolPolicy': 'allow-all'|'https-only'|'redirect-to-https',
'MinTTL': 123,
'AllowedMethods': {
    'Quantity': 123,
    'Items': [
        'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
    ],
    'CachedMethods': {
        'Quantity': 123,
        'Items': [
            'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
        ]
    },
},
'SmoothStreaming': True|False,
'DefaultTTL': 123,
'MaxTTL': 123
},
'CacheBehaviors': {
    'Quantity': 123,
    'Items': [
        {
            'PathPattern': 'string',
            'TargetOriginId': 'string',
            'ForwardedValues': {
                'QueryString': True|False,
                'Cookies': {
```



```

        'Forward': 'none'|'whitelist'|'all',
        'WhitelistedNames': {
            'Quantity': 123,
            'Items': [
                'string',
            ]
        },
    },
    'Headers': {
        'Quantity': 123,
        'Items': [
            'string',
        ]
    },
},
'TrustedSigners': {
    'Enabled': True|False,
    'Quantity': 123,
    'Items': [
        'string',
    ]
},
'ViewerProtocolPolicy': 'allow-all'|'https-only'|'redirect-to-https',
'MinTTL': 123,
'AllowedMethods': {
    'Quantity': 123,
    'Items': [
        'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
    ],
    'CachedMethods': {
        'Quantity': 123,
        'Items': [
            'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
        ]
    }
},
'SmoothStreaming': True|False,
'DefaultTTL': 123,
'MaxTTL': 123
},
]
},
'CustomErrorResponses': {
    'Quantity': 123,
    'Items': [
        {
            'ErrorCode': 123,
            'ResponsePagePath': 'string',
            'ResponseCode': 'string',
            'ErrorCachingMinTTL': 123
        },
    ],
},
],
'Comment': 'string',
'Logging': {
    'Enabled': True|False,
    'IncludeCookies': True|False,
    'Bucket': 'string',

```

```
    'Prefix': 'string'
},
'PriceClass': 'PriceClass_100'|'PriceClass_200'|'PriceClass_All',
'Enabled': True|False,
'ViewerCertificate': {
    'IAMCertificateId': 'string',
    'CloudFrontDefaultCertificate': True|False,
    'SSLSupportMethod': 'sni-only'|'vip',
    'MinimumProtocolVersion': 'SSLv3'|'TLSv1'
},
'Restrictions': {
    'GeoRestriction': {
        'RestrictionType': 'blacklist'|'whitelist'|'none',
        'Quantity': 123,
        'Items': [
            'string',
        ]
    }
},
'ETag': 'string'
}
```

Response Structure

- (*dict*) – The returned result of the corresponding request.
 - **DistributionConfig** (*dict*) – The distribution’s configuration information.
 - * **CallerReference** (*string*) – A unique number that ensures the request can’t be replayed. If the CallerReference is new (no matter the content of the DistributionConfig object), a new distribution is created. If the CallerReference is a value you already sent in a previous request to create a distribution, and the content of the DistributionConfig is identical to the original request (ignoring white space), the response includes the same information returned to the original request. If the CallerReference is a value you already sent in a previous request to create a distribution but the content of the DistributionConfig is different from the original request, CloudFront returns a DistributionAlreadyExists error.
 - * **Aliases** (*dict*) – A complex type that contains information about CNAMEs (alternate domain names), if any, for this distribution.
 - **Quantity** (*integer*) – The number of CNAMEs, if any, for this distribution.
 - **Items** (*list*) – Optional: A complex type that contains CNAME elements, if any, for this distribution. If Quantity is 0, you can omit Items.
 - (*string*) –
 - * **DefaultRootObject** (*string*) – The object that you want CloudFront to return (for example, index.html) when an end user requests the root URL for your distribution (<http://www.example.com>) instead of an object in your distribution (<http://www.example.com/index.html>). Specifying a default root object avoids exposing the contents of your distribution. If you don’t want to specify a default root object when you create a distribution, include an empty DefaultRootObject element. To delete the default root object from an existing distribution, update the distribution configuration and include an empty DefaultRootObject

element. To replace the default root object, update the distribution configuration and specify the new object.

- * **Origins** (*dict*) – A complex type that contains information about origins for this distribution.
 - **Quantity** (*integer*) – The number of origins for this distribution.
 - **Items** (*list*) – A complex type that contains origins for this distribution.
 - (*dict*) – A complex type that describes the Amazon S3 bucket or the HTTP server (for example, a web server) from which CloudFront gets your files. You must create at least one origin.
 - **Id** (*string*) – A unique identifier for the origin. The value of Id must be unique within the distribution. You use the value of Id when you create a cache behavior. The Id identifies the origin that CloudFront routes a request to when the request matches the path pattern for that cache behavior.
 - **DomainName** (*string*) – Amazon S3 origins: The DNS name of the Amazon S3 bucket from which you want CloudFront to get objects for this origin, for example, myaws-bucket.s3.amazonaws.com. Custom origins: The DNS domain name for the HTTP server from which you want CloudFront to get objects for this origin, for example, www.example.com.
 - **OriginPath** (*string*) – An optional element that causes CloudFront to request your content from a directory in your Amazon S3 bucket or your custom origin. When you include the OriginPath element, specify the directory name, beginning with a /. CloudFront appends the directory name to the value of DomainName.
 - **S3OriginConfig** (*dict*) – A complex type that contains information about the Amazon S3 origin. If the origin is a custom origin, use the CustomOriginConfig element instead.
 - **OriginAccessIdentity** (*string*) – The CloudFront origin access identity to associate with the origin. Use an origin access identity to configure the origin so that end users can only access objects in an Amazon S3 bucket through CloudFront. If you want end users to be able to access objects using either the CloudFront URL or the Amazon S3 URL, specify an empty OriginAccessIdentity element. To delete the origin access identity from an existing distribution, update the distribution configuration and include an empty OriginAccessIdentity element. To replace the origin access identity, update the distribution configuration and specify the new origin access identity. Use the format origin-access-identity/cloudfront/Id where Id is the value that CloudFront returned in the Id element when you created the origin access identity.
 - **CustomOriginConfig** (*dict*) – A complex type that contains information about a custom origin. If the origin is an Amazon S3 bucket, use the S3OriginConfig element instead.
 - **HTTPPort** (*integer*) – The HTTP port the custom origin listens on.
 - **HTTPSPort** (*integer*) – The HTTPS port the custom origin listens on.
 - **OriginProtocolPolicy** (*string*) – The origin protocol policy

to apply to your origin.

- * **DefaultCacheBehavior** (*dict*) – A complex type that describes the default cache behavior if you do not specify a CacheBehavior element or if files don't match any of the values of PathPattern in CacheBehavior elements. You must create exactly one default cache behavior.
 - **TargetOriginId** (*string*) – The value of ID for the origin that you want CloudFront to route requests to when a request matches the path pattern either for a cache behavior or for the default cache behavior.
 - **ForwardedValues** (*dict*) – A complex type that specifies how CloudFront handles query strings, cookies and headers.
 - **QueryString** (*boolean*) – Indicates whether you want CloudFront to forward query strings to the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
 - **Cookies** (*dict*) – A complex type that specifies how CloudFront handles cookies.
 - **Forward** (*string*) – Use this element to specify whether you want CloudFront to forward cookies to the origin that is associated with this cache behavior. You can specify all, none or whitelist. If you choose All, CloudFront forwards all cookies regardless of how many your application uses.
 - **WhitelistedNames** (*dict*) – A complex type that specifies the whitelisted cookies, if any, that you want CloudFront to forward to your origin that is associated with this cache behavior.
 - **Quantity** (*integer*) – The number of whitelisted cookies for this cache behavior.
 - **Items** (*list*) – Optional: A complex type that contains whitelisted cookies for this cache behavior. If Quantity is 0, you can omit Items.
 - (*string*) –
 - **Headers** (*dict*) – A complex type that specifies the Headers, if any, that you want CloudFront to vary upon for this cache behavior.
 - **Quantity** (*integer*) – The number of different headers that you want CloudFront to forward to the origin and to vary on for this cache behavior. The maximum number of headers that you can specify by name is 10. If you want CloudFront to forward all headers to the origin and vary on all of them, specify 1 for Quantity and * for Name. If you don't want CloudFront to forward any additional headers to the origin or to vary on any headers, specify 0 for Quantity and omit Items.
 - **Items** (*list*) – Optional: A complex type that contains a Name element for each header that you want CloudFront to forward to the origin and to vary on for this cache behavior. If Quantity is 0, omit Items.
 - (*string*) –
 - **TrustedSigners** (*dict*) – A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the PathPattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For

more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don't want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more trusted signers, change Enabled to true (if it's currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.

- **Enabled** (*boolean*) – Specifies whether you want to require end users to use signed URLs to access the files specified by PathPattern and TargetOriginId.
- **Quantity** (*integer*) – The number of trusted signers for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **ViewerProtocolPolicy** (*string*) – Use this element to specify the protocol that users can use to access the files in the origin specified by TargetOriginId when a request matches the path pattern in PathPattern. If you want CloudFront to allow end users to use any available protocol, specify allow-all. If you want CloudFront to require HTTPS, specify https. If you want CloudFront to respond to an HTTP request with an HTTP status code of 301 (Moved Permanently) and the HTTPS URL, specify redirect-to-https. The viewer then resubmits the request using the HTTPS URL.
- **MinTTL** (*integer*) – The minimum amount of time that you want objects to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **AllowedMethods** (*dict*) – A complex type that controls which HTTP methods CloudFront processes and forwards to your Amazon S3 bucket or your custom origin. There are three choices:
 - CloudFront forwards only GET and HEAD requests.
 - CloudFront forwards only GET, HEAD and OPTIONS requests.
 - CloudFront forwards GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests.
 If you pick the third choice, you may need to restrict access to your Amazon S3 bucket or to your custom origin so users can't perform operations that you don't want them to. For example, you may not want users to have permission to delete objects from your origin.
- **Quantity** (*integer*) – The number of HTTP methods that you want CloudFront to forward to your origin. Valid values are 2 (for GET and HEAD requests), 3 (for GET, HEAD and OPTIONS requests) and 7 (for GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests).
- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to process and forward to your origin.
- (*string*) –
- **CachedMethods** (*dict*) – A complex type that controls

whether CloudFront caches the response to requests using the specified HTTP methods. There are two choices:

- CloudFront caches responses to GET and HEAD requests.
- CloudFront caches responses to GET, HEAD, and OPTIONS requests. If you pick the second choice for your S3 Origin, you may need to forward Access-Control-Request-Method, Access-Control-Request-Headers and Origin headers for the responses to be cached correctly.

- **Quantity** (*integer*) – The number of HTTP methods for which you want CloudFront to cache responses. Valid values are 2 (for caching responses to GET and HEAD requests) and 3 (for caching responses to GET, HEAD, and OPTIONS requests).
- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to cache responses to.
- (*string*) –
- **SmoothStreaming** (*boolean*) – Indicates whether you want to distribute media files in Microsoft Smooth Streaming format using the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
- **DefaultTTL** (*integer*) – If you don't configure your origin to add a Cache-Control max-age directive or an Expires header, DefaultTTL is the default amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin does not add HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **MaxTTL** (*integer*) – The maximum amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin adds HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- * **CacheBehaviors** (*dict*) – A complex type that contains zero or more CacheBehavior elements.
 - **Quantity** (*integer*) – The number of cache behaviors for this distribution.
 - **Items** (*list*) – Optional: A complex type that contains cache behaviors for this distribution. If Quantity is 0, you can omit Items.
 - (*dict*) – A complex type that describes how CloudFront processes requests. You can create up to 10 cache behaviors. You must create at least as many cache behaviors (including the default cache behavior) as you have origins if you want CloudFront to distribute objects from all of the origins. Each cache behavior specifies the one origin from which you want CloudFront to get objects. If you have two origins and only the default cache behavior, the default cache behavior will cause CloudFront to get objects from one of the origins, but the other origin will never be used. If you don't

want to specify any cache behaviors, include only an empty `CacheBehaviors` element. Don't include an empty `CacheBehavior` element, or CloudFront returns a `MalformedXML` error. To delete all cache behaviors in an existing distribution, update the distribution configuration and include only an empty `CacheBehaviors` element. To add, change, or remove one or more cache behaviors, update the distribution configuration and specify all of the cache behaviors that you want to include in the updated distribution.

- **PathPattern** (*string*) – The pattern (for example, `images/*.jpg`) that specifies which requests you want this cache behavior to apply to. When CloudFront receives an end-user request, the requested path is compared with path patterns in the order in which cache behaviors are listed in the distribution. The path pattern for the default cache behavior is `*` and cannot be changed. If the request for an object does not match the path pattern for any cache behaviors, CloudFront applies the behavior in the default cache behavior.
- **TargetOriginId** (*string*) – The value of ID for the origin that you want CloudFront to route requests to when a request matches the path pattern either for a cache behavior or for the default cache behavior.
- **ForwardedValues** (*dict*) – A complex type that specifies how CloudFront handles query strings, cookies and headers.
- **QueryString** (*boolean*) – Indicates whether you want CloudFront to forward query strings to the origin that is associated with this cache behavior. If so, specify `true`; if not, specify `false`.
- **Cookies** (*dict*) – A complex type that specifies how CloudFront handles cookies.
- **Forward** (*string*) – Use this element to specify whether you want CloudFront to forward cookies to the origin that is associated with this cache behavior. You can specify `all`, `none` or `whitelist`. If you choose `All`, CloudFront forwards all cookies regardless of how many your application uses.
- **WhitelistedNames** (*dict*) – A complex type that specifies the whitelisted cookies, if any, that you want CloudFront to forward to your origin that is associated with this cache behavior.
- **Quantity** (*integer*) – The number of whitelisted cookies for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains whitelisted cookies for this cache behavior. If `Quantity` is 0, you can omit `Items`.
- (*string*) –
- **Headers** (*dict*) – A complex type that specifies the Headers, if any, that you want CloudFront to vary upon for this cache behavior.
- **Quantity** (*integer*) – The number of different headers that you want CloudFront to forward to the origin and to vary on for this cache behavior. The maximum number of headers that you can specify by name is 10. If you want CloudFront to forward all headers to the origin and vary on all of them, specify 1 for `Quantity` and `*` for `Name`. If you don't want CloudFront to forward any additional headers to the origin or to vary on

any headers, specify 0 for Quantity and omit Items.

- **Items** (*list*) – Optional: A complex type that contains a Name element for each header that you want CloudFront to forward to the origin and to vary on for this cache behavior. If Quantity is 0, omit Items.
- (*string*) –
- **TrustedSigners** (*dict*) – A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the PathPattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don't want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more trusted signers, change Enabled to true (if it's currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.
- **Enabled** (*boolean*) – Specifies whether you want to require end users to use signed URLs to access the files specified by PathPattern and TargetOriginId.
- **Quantity** (*integer*) – The number of trusted signers for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **ViewerProtocolPolicy** (*string*) – Use this element to specify the protocol that users can use to access the files in the origin specified by TargetOriginId when a request matches the path pattern in PathPattern. If you want CloudFront to allow end users to use any available protocol, specify allow-all. If you want CloudFront to require HTTPS, specify https. If you want CloudFront to respond to an HTTP request with an HTTP status code of 301 (Moved Permanently) and the HTTPS URL, specify redirect-to-https. The viewer then resubmits the request using the HTTPS URL.
- **MinTTL** (*integer*) – The minimum amount of time that you want objects to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **AllowedMethods** (*dict*) – A complex type that controls which HTTP methods CloudFront processes and forwards to your Amazon S3 bucket or your custom origin. There are three choices:
 - CloudFront forwards only GET and HEAD requests.
 - CloudFront forwards only GET, HEAD and OPTIONS requests.
 - CloudFront forwards GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests.If you pick the third choice, you may need to restrict access to your Amazon S3 bucket or to your custom origin so users can't perform operations that you don't want them to. For example,

you may not want users to have permission to delete objects from your origin.

- **Quantity** (*integer*) – The number of HTTP methods that you want CloudFront to forward to your origin. Valid values are 2 (for GET and HEAD requests), 3 (for GET, HEAD and OPTIONS requests) and 7 (for GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests).
- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to process and forward to your origin.
- (*string*) –
- **CachedMethods** (*dict*) – A complex type that controls whether CloudFront caches the response to requests using the specified HTTP methods. There are two choices: - CloudFront caches responses to GET and HEAD requests. - CloudFront caches responses to GET, HEAD, and OPTIONS requests. If you pick the second choice for your S3 Origin, you may need to forward Access-Control-Request-Method, Access-Control-Request-Headers and Origin headers for the responses to be cached correctly.
- **Quantity** (*integer*) – The number of HTTP methods for which you want CloudFront to cache responses. Valid values are 2 (for caching responses to GET and HEAD requests) and 3 (for caching responses to GET, HEAD, and OPTIONS requests).
- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to cache responses to.
- (*string*) –
- **SmoothStreaming** (*boolean*) – Indicates whether you want to distribute media files in Microsoft Smooth Streaming format using the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
- **DefaultTTL** (*integer*) – If you don't configure your origin to add a Cache-Control max-age directive or an Expires header, DefaultTTL is the default amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin does not add HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **MaxTTL** (*integer*) – The maximum amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin adds HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- * **CustomErrorResponses** (*dict*) – A complex type that contains zero or more CustomErrorResponse elements.
 - **Quantity** (*integer*) – The number of custom error responses for this distribution.
 - **Items** (*list*) – Optional: A complex type that contains custom

error responses for this distribution. If Quantity is 0, you can omit Items.

- ***dict*** – A complex type that describes how you'd prefer CloudFront to respond to requests that result in either a 4xx or 5xx response. You can control whether a custom error page should be displayed, what the desired response code should be for this error page and how long should the error response be cached by CloudFront. If you don't want to specify any custom error responses, include only an empty CustomErrorResponses element. To delete all custom error responses in an existing distribution, update the distribution configuration and include only an empty CustomErrorResponses element. To add, change, or remove one or more custom error responses, update the distribution configuration and specify all of the custom error responses that you want to include in the updated distribution.
- **ErrorCode** (*integer*) – The 4xx or 5xx HTTP status code that you want to customize. For a list of HTTP status codes that you can customize, see CloudFront documentation.
- **ResponsePagePath** (*string*) – The path of the custom error page (for example, /custom_404.html). The path is relative to the distribution and must begin with a slash (/). If the path includes any non-ASCII characters or unsafe characters as defined in RFC 1783 (<http://www.ietf.org/rfc/rfc1738.txt>), URL encode those characters. Do not URL encode any other characters in the path, or CloudFront will not return the custom error page to the viewer.
- **ResponseCode** (*string*) – The HTTP status code that you want CloudFront to return with the custom error page to the viewer. For a list of HTTP status codes that you can replace, see CloudFront Documentation.
- **ErrorCachingMinTTL** (*integer*) – The minimum amount of time you want HTTP error codes to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 31,536,000.
- * **Comment** (*string*) – Any comments you want to include about the distribution.
- * **Logging** (*dict*) – A complex type that controls whether access logs are written for the distribution.
 - **Enabled** (*boolean*) – Specifies whether you want CloudFront to save access logs to an Amazon S3 bucket. If you do not want to enable logging when you create a distribution or if you want to disable logging for an existing distribution, specify false for Enabled, and specify empty Bucket and Prefix elements. If you specify false for Enabled but you specify values for Bucket, prefix and IncludeCookies, the values are automatically deleted.
 - **IncludeCookies** (*boolean*) – Specifies whether you want CloudFront to include cookies in access logs, specify true for IncludeCookies. If you choose to include cookies in logs, CloudFront logs all cookies regardless of how you configure the cache behaviors for this distribution. If you do not want to include cookies when you create a distribution or if you want

to disable include cookies for an existing distribution, specify `false` for `IncludeCookies`.

- **Bucket** (*string*) – The Amazon S3 bucket to store the access logs in, for example, `myawslogbucket.s3.amazonaws.com`.
- **Prefix** (*string*) – An optional string that you want CloudFront to prefix to the access log filenames for this distribution, for example, `myprefix/`. If you want to enable logging, but you do not want to specify a prefix, you still must include an empty `Prefix` element in the `Logging` element.
- * **PriceClass** (*string*) – A complex type that contains information about price class for this distribution.
- * **Enabled** (*boolean*) – Whether the distribution is enabled to accept end user requests for content.
- * **ViewerCertificate** (*dict*) – A complex type that contains information about viewer certificates for this distribution.
 - **IAMCertificateId** (*string*) – If you want viewers to use HTTPS to request your objects and you’re using an alternate domain name in your object URLs (for example, <https://example.com/logo.jpg>), specify the IAM certificate identifier of the custom viewer certificate for this distribution. Specify either this value or `CloudFrontDefaultCertificate`.
 - **CloudFrontDefaultCertificate** (*boolean*) – If you want viewers to use HTTPS to request your objects and you’re using the CloudFront domain name of your distribution in your object URLs (for example, <https://d1111111abcdef8.cloudfront.net/logo.jpg>), set to `true`. Omit this value if you are setting an `IAMCertificateId`.
 - **SSLSupportMethod** (*string*) – If you specify a value for `IAMCertificateId`, you must also specify how you want CloudFront to serve HTTPS requests. Valid values are `vip` and `sni-only`. If you specify `vip`, CloudFront uses dedicated IP addresses for your content and can respond to HTTPS requests from any viewer. However, you must request permission to use this feature, and you incur additional monthly charges. If you specify `sni-only`, CloudFront can only respond to HTTPS requests from viewers that support Server Name Indication (SNI). All modern browsers support SNI, but some browsers still in use don’t support SNI. Do not specify a value for `SSLSupportMethod` if you specified `true` for `CloudFrontDefaultCertificate`.
 - **MinimumProtocolVersion** (*string*) – Specify the minimum version of the SSL protocol that you want CloudFront to use, `SSLv3` or `TLSv1`, for HTTPS connections. CloudFront will serve your objects only to browsers or devices that support at least the SSL version that you specify. The `TLSv1` protocol is more secure, so we recommend that you specify `SSLv3` only if your users are using browsers or devices that don’t support `TLSv1`. If you’re using a custom certificate (if you specify a value for `IAMCertificateId`) and if you’re using dedicated IP (if you specify `vip` for `SSLSupportMethod`), you can choose `SSLv3` or `TLSv1` as the `MinimumProtocolVersion`. If you’re using a custom certificate (if you specify a value for `IAMCertificateId`) and if you’re using SNI (if you specify `sni-only` for `SSLSupportMethod`), you must specify `TLSv1` for Mini-

mumProtocolVersion.

* **Restrictions** (*dict*) – A complex type that identifies ways in which you want to restrict distribution of your content.

- **GeoRestriction** (*dict*) – A complex type that controls the countries in which your content is distributed. For more information about geo restriction, go to Customizing Error Responses in the Amazon CloudFront Developer Guide. CloudFront determines the location of your users using MaxMind GeoIP databases. For information about the accuracy of these databases, see How accurate are your GeoIP databases? on the MaxMind website.

- **RestrictionType** (*string*) – The method that you want to use to restrict distribution of your content by country: - none: No geo restriction is enabled, meaning access to content is not restricted by client geo location. - blacklist: The Location elements specify the countries in which you do not want CloudFront to distribute your content. - whitelist: The Location elements specify the countries in which you want CloudFront to distribute your content.

- **Quantity** (*integer*) – When geo restriction is enabled, this is the number of countries in your whitelist or blacklist. Otherwise, when it is not enabled, Quantity is 0, and you can omit Items.

- **Items** (*list*) – A complex type that contains a Location element for each country in which you want CloudFront either to distribute your content (whitelist) or not distribute your content (blacklist). The Location element is a two-letter, uppercase country code for a country that you want to include in your blacklist or whitelist. Include one Location element for each country. CloudFront and MaxMind both use ISO 3166 country codes. For the current list of countries and the corresponding codes, see ISO 3166-1-alpha-2 code on the International Organization for Standardization website. You can also refer to the country list in the CloudFront console, which includes both country names and codes.

- (*string*) –

- **ETag** (*string*) – The current version of the configuration. For example: E2QWRUHAPOMQZL.

get_invalidation (***kwargs*)

Get the information about an invalidation.

Request Syntax

```
response = client.get_invalidation(
    DistributionId='string',
    Id='string'
)
```

Parameters

- **DistributionId** (*string*) – [REQUIRED] The distribution's id.
- **Id** (*string*) – [REQUIRED] The invalidation's id.

Return type dict

Returns

Response Syntax

```
{
  'Invalidation': {
    'Id': 'string',
    'Status': 'string',
    'CreateTime': datetime(2015, 1, 1),
    'InvalidationBatch': {
      'Paths': {
        'Quantity': 123,
        'Items': [
          'string',
        ]
      },
      'CallerReference': 'string'
    }
  }
}
```

Response Structure

- *(dict)* – The returned result of the corresponding request.
 - **Invalidation** (*dict*) – The invalidation’s information.
 - * **Id** (*string*) – The identifier for the invalidation request. For example: IDFDVBD632BHDS5.
 - * **Status** (*string*) – The status of the invalidation request. When the invalidation batch is finished, the status is Completed.
 - * **CreateTime** (*datetime*) – The date and time the invalidation request was first made.
 - * **InvalidationBatch** (*dict*) – The current invalidation information for the batch request.
 - **Paths** (*dict*) – The path of the object to invalidate. The path is relative to the distribution and must begin with a slash (/). You must enclose each invalidation object with the Path element tags. If the path includes non-ASCII characters or unsafe characters as defined in RFC 1738 (<http://www.ietf.org/rfc/rfc1738.txt>), URL encode those characters. Do not URL encode any other characters in the path, or CloudFront will not invalidate the old version of the updated object.
 - **Quantity** (*integer*) – The number of objects that you want to invalidate.
 - **Items** (*list*) – A complex type that contains a list of the objects that you want to invalidate.
 - (*string*) –
 - **CallerReference** (*string*) – A unique name that ensures the request can’t be replayed. If the CallerReference is new (no matter the content of the Path object), a new distribution is created. If the CallerReference is a value you already sent in a previous request to create an invalidation batch, and the content of each Path element is identical to the original request, the response includes the same information returned to the original request. If the CallerReference is a value you already sent in a previous request to create a distribution but the content of any Path is different from the original request, CloudFront returns an InvalidationBatchAlreadyExists error.

`get_paginator(operation_name)`

Create a paginator for an operation.

Parameters `operation_name` (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises `OperationNotPageableError` Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_streaming_distribution (***kwargs*)

Get the information about a streaming distribution.

Request Syntax

```
response = client.get_streaming_distribution(
    Id='string'
)
```

Parameters `Id` (*string*) – **[REQUIRED]** The streaming distribution's id.

Return type dict

Returns

Response Syntax

```
{
  'StreamingDistribution': {
    'Id': 'string',
    'Status': 'string',
    'LastModifiedTime': datetime(2015, 1, 1),
    'DomainName': 'string',
    'ActiveTrustedSigners': {
      'Enabled': True|False,
      'Quantity': 123,
      'Items': [
        {
          'AwsAccountNumber': 'string',
          'KeyPairIds': {
            'Quantity': 123,
            'Items': [
              'string',
            ]
          }
        }
      ],
    },
  ],
  'StreamingDistributionConfig': {
    'CallerReference': 'string',
    'S3Origin': {
      'DomainName': 'string',
      'OriginAccessIdentity': 'string'
    },
    'Aliases': {
      'Quantity': 123,
      'Items': [
        'string',
      ]
    }
  }
}
```

```

    },
    'Comment': 'string',
    'Logging': {
        'Enabled': True|False,
        'Bucket': 'string',
        'Prefix': 'string'
    },
    'TrustedSigners': {
        'Enabled': True|False,
        'Quantity': 123,
        'Items': [
            'string',
        ]
    },
    'PriceClass': 'PriceClass_100'|'PriceClass_200'|'PriceClass_All',
    'Enabled': True|False
}
},
'ETag': 'string'
}

```

Response Structure

- (*dict*) – The returned result of the corresponding request.
 - **StreamingDistribution** (*dict*) – The streaming distribution’s information.
 - * **Id** (*string*) – The identifier for the streaming distribution. For example: EGTXBD79H29TRA8.
 - * **Status** (*string*) – The current status of the streaming distribution. When the status is Deployed, the distribution’s information is fully propagated throughout the Amazon CloudFront system.
 - * **LastModifiedTime** (*datetime*) – The date and time the distribution was last modified.
 - * **DomainName** (*string*) – The domain name corresponding to the streaming distribution. For example: s5c39gqb8ow64r.cloudfront.net.
 - * **ActiveTrustedSigners** (*dict*) – CloudFront automatically adds this element to the response only if you’ve set up the distribution to serve private content with signed URLs. The element lists the key pair IDs that CloudFront is aware of for each trusted signer. The Signer child element lists the AWS account number of the trusted signer (or an empty Self element if the signer is you). The Signer element also includes the IDs of any active key pairs associated with the trusted signer’s AWS account. If no KeyPairId element appears for a Signer, that signer can’t create working signed URLs.
 - **Enabled** (*boolean*) – Each active trusted signer.
 - **Quantity** (*integer*) – The number of unique trusted signers included in all cache behaviors. For example, if three cache behaviors all list the same three AWS accounts, the value of Quantity for ActiveTrustedSigners will be 3.
 - **Items** (*list*) – A complex type that contains one Signer complex type for each unique trusted signer that is specified in the TrustedSigners complex type, including trusted signers in the default cache behavior and in all of the other cache behaviors.
 - (*dict*) – A complex type that lists the AWS accounts that were included in the TrustedSigners complex type, as well as their active CloudFront key pair IDs, if any.

- **AwsAccountNumber** (*string*) – Specifies an AWS account that can create signed URLs. Values: self, which indicates that the AWS account that was used to create the distribution can create signed URLs, or an AWS account number. Omit the dashes in the account number.
- **KeyPairIds** (*dict*) – A complex type that lists the active CloudFront key pairs, if any, that are associated with AwsAccountNumber.
- **Quantity** (*integer*) – The number of active CloudFront key pairs for AwsAccountNumber.
- **Items** (*list*) – A complex type that lists the active CloudFront key pairs, if any, that are associated with AwsAccountNumber.
- (*string*) –
- * **StreamingDistributionConfig** (*dict*) – The current configuration information for the streaming distribution.
 - **CallerReference** (*string*) – A unique number that ensures the request can't be replayed. If the CallerReference is new (no matter the content of the StreamingDistributionConfig object), a new streaming distribution is created. If the CallerReference is a value you already sent in a previous request to create a streaming distribution, and the content of the StreamingDistributionConfig is identical to the original request (ignoring white space), the response includes the same information returned to the original request. If the CallerReference is a value you already sent in a previous request to create a streaming distribution but the content of the StreamingDistributionConfig is different from the original request, CloudFront returns a DistributionAlreadyExists error.
 - **S3Origin** (*dict*) – A complex type that contains information about the Amazon S3 bucket from which you want CloudFront to get your media files for distribution.
 - **DomainName** (*string*) – The DNS name of the S3 origin.
 - **OriginAccessIdentity** (*string*) – Your S3 origin's origin access identity.
 - **Aliases** (*dict*) – A complex type that contains information about CNAMEs (alternate domain names), if any, for this streaming distribution.
 - **Quantity** (*integer*) – The number of CNAMEs, if any, for this distribution.
 - **Items** (*list*) – Optional: A complex type that contains CNAME elements, if any, for this distribution. If Quantity is 0, you can omit Items.
 - (*string*) –
 - **Comment** (*string*) – Any comments you want to include about the streaming distribution.
 - **Logging** (*dict*) – A complex type that controls whether access logs are written for the streaming distribution.
 - **Enabled** (*boolean*) – Specifies whether you want CloudFront to save access logs to an Amazon S3 bucket. If you do not want to enable logging when you create a streaming distribution or if you want to disable logging for an existing streaming distribution, specify false for Enabled, and specify empty Bucket and Prefix elements. If you specify false for Enabled

but you specify values for Bucket and Prefix, the values are automatically deleted.

- **Bucket** (*string*) – The Amazon S3 bucket to store the access logs in, for example, myawslogbucket.s3.amazonaws.com.
- **Prefix** (*string*) – An optional string that you want CloudFront to prefix to the access log filenames for this streaming distribution, for example, myprefix/. If you want to enable logging, but you do not want to specify a prefix, you still must include an empty Prefix element in the Logging element.
- **TrustedSigners** (*dict*) – A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the PathPattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don't want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more trusted signers, change Enabled to true (if it's currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.
- **Enabled** (*boolean*) – Specifies whether you want to require end users to use signed URLs to access the files specified by PathPattern and TargetOriginId.
- **Quantity** (*integer*) – The number of trusted signers for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **PriceClass** (*string*) – A complex type that contains information about price class for this streaming distribution.
- **Enabled** (*boolean*) – Whether the streaming distribution is enabled to accept end user requests for content.
- **ETag** (*string*) – The current version of the streaming distribution's information. For example: E2QWRUHAPOMQZL.

get_streaming_distribution_config (**kwargs)

Get the configuration information about a streaming distribution.

Request Syntax

```
response = client.get_streaming_distribution_config(
    Id='string'
)
```

Parameters **Id** (*string*) – [REQUIRED] The streaming distribution's id.

Return type dict

Returns

Response Syntax

```
{
    'StreamingDistributionConfig': {
```

```
'CallerReference': 'string',
'S3Origin': {
    'DomainName': 'string',
    'OriginAccessIdentity': 'string'
},
'Aliases': {
    'Quantity': 123,
    'Items': [
        'string',
    ]
},
'Comment': 'string',
'Logging': {
    'Enabled': True|False,
    'Bucket': 'string',
    'Prefix': 'string'
},
'TrustedSigners': {
    'Enabled': True|False,
    'Quantity': 123,
    'Items': [
        'string',
    ]
},
'PriceClass': 'PriceClass_100'|'PriceClass_200'|'PriceClass_All',
'Enabled': True|False
},
'ETag': 'string'
}
```

Response Structure

- (*dict*) – The returned result of the corresponding request.
 - **StreamingDistributionConfig** (*dict*) – The streaming distribution’s configuration information.
 - * **CallerReference** (*string*) – A unique number that ensures the request can’t be replayed. If the CallerReference is new (no matter the content of the StreamingDistributionConfig object), a new streaming distribution is created. If the CallerReference is a value you already sent in a previous request to create a streaming distribution, and the content of the StreamingDistributionConfig is identical to the original request (ignoring white space), the response includes the same information returned to the original request. If the CallerReference is a value you already sent in a previous request to create a streaming distribution but the content of the StreamingDistributionConfig is different from the original request, CloudFront returns a DistributionAlreadyExists error.
 - * **S3Origin** (*dict*) – A complex type that contains information about the Amazon S3 bucket from which you want CloudFront to get your media files for distribution.
 - **DomainName** (*string*) – The DNS name of the S3 origin.
 - **OriginAccessIdentity** (*string*) – Your S3 origin’s origin access identity.
 - * **Aliases** (*dict*) – A complex type that contains information about CNAMEs (alternate domain names), if any, for this streaming distribution.
 - **Quantity** (*integer*) – The number of CNAMEs, if any, for this

- distribution.
- **Items** (*list*) – Optional: A complex type that contains CNAME elements, if any, for this distribution. If Quantity is 0, you can omit Items.
- (*string*) –
- * **Comment** (*string*) – Any comments you want to include about the streaming distribution.
- * **Logging** (*dict*) – A complex type that controls whether access logs are written for the streaming distribution.
 - **Enabled** (*boolean*) – Specifies whether you want CloudFront to save access logs to an Amazon S3 bucket. If you do not want to enable logging when you create a streaming distribution or if you want to disable logging for an existing streaming distribution, specify false for Enabled, and specify empty Bucket and Prefix elements. If you specify false for Enabled but you specify values for Bucket and Prefix, the values are automatically deleted.
 - **Bucket** (*string*) – The Amazon S3 bucket to store the access logs in, for example, myawslogbucket.s3.amazonaws.com.
 - **Prefix** (*string*) – An optional string that you want CloudFront to prefix to the access log filenames for this streaming distribution, for example, myprefix/. If you want to enable logging, but you do not want to specify a prefix, you still must include an empty Prefix element in the Logging element.
- * **TrustedSigners** (*dict*) – A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the PathPattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don't want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more trusted signers, change Enabled to true (if it's currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.
 - **Enabled** (*boolean*) – Specifies whether you want to require end users to use signed URLs to access the files specified by PathPattern and TargetOriginId.
 - **Quantity** (*integer*) – The number of trusted signers for this cache behavior.
 - **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit Items.
 - (*string*) –
- * **PriceClass** (*string*) – A complex type that contains information about price class for this streaming distribution.
- * **Enabled** (*boolean*) – Whether the streaming distribution is enabled to accept end user requests for content.
- **ETag** (*string*) – The current version of the configuration. For example: E2QWRUHAPOMQZL.

`get_waiter` (*waiter_name*)

`list_cloud_front_origin_access_identities(**kwargs)`

List origin access identities.

Request Syntax

```
response = client.list_cloud_front_origin_access_identities(
    Marker='string',
    MaxItems='string'
)
```

Parameters

- **Marker** (*string*) – Use this when paginating results to indicate where to begin in your list of origin access identities. The results include identities in the list that occur after the marker. To get the next page of results, set the Marker to the value of the NextMarker from the current page’s response (which is also the ID of the last identity on that page).
- **MaxItems** (*string*) – The maximum number of origin access identities you want in the response body.

Return type dict

Returns

Response Syntax

```
{
  'CloudFrontOriginAccessIdentityList': {
    'Marker': 'string',
    'NextMarker': 'string',
    'MaxItems': 123,
    'IsTruncated': True|False,
    'Quantity': 123,
    'Items': [
      {
        'Id': 'string',
        'S3CanonicalUserId': 'string',
        'Comment': 'string'
      },
    ]
  }
}
```

Response Structure

- (*dict*) – The returned result of the corresponding request.
 - **CloudFrontOriginAccessIdentityList** (*dict*) – The CloudFrontOriginAccessIdentityList type.
 - * **Marker** (*string*) – The value you provided for the Marker request parameter.
 - * **NextMarker** (*string*) – If IsTruncated is true, this element is present and contains the value you can use for the Marker request parameter to continue listing your origin access identities where they left off.
 - * **MaxItems** (*integer*) – The value you provided for the MaxItems request parameter.
 - * **IsTruncated** (*boolean*) – A flag that indicates whether more origin access identities remain to be listed. If your results were truncated, you can make a follow-up pagination request using the Marker request parameter to retrieve more items in the list.
 - * **Quantity** (*integer*) – The number of CloudFront origin access identities that were created by the current AWS account.

- * **Items** (*list*) – A complex type that contains one `CloudFrontOriginAccessIdentitySummary` element for each origin access identity that was created by the current AWS account.
 - (*dict*) – Summary of the information about a CloudFront origin access identity.
 - **Id** (*string*) – The ID for the origin access identity. For example: E74FTE3AJFJ256A.
 - **S3CanonicalUserId** (*string*) – The Amazon S3 canonical user ID for the origin access identity, which you use when giving the origin access identity read permission to an object in Amazon S3.
 - **Comment** (*string*) – The comment for this origin access identity, as originally specified when created.

list_distributions (***kwargs*)

List distributions.

Request Syntax

```
response = client.list_distributions(
    Marker='string',
    MaxItems='string'
)
```

Parameters

- **Marker** (*string*) – Use this when paginating results to indicate where to begin in your list of distributions. The results include distributions in the list that occur after the marker. To get the next page of results, set the Marker to the value of the NextMarker from the current page's response (which is also the ID of the last distribution on that page).
- **MaxItems** (*string*) – The maximum number of distributions you want in the response body.

Return type dict

Returns

Response Syntax

```
{
  'DistributionList': {
    'Marker': 'string',
    'NextMarker': 'string',
    'MaxItems': 123,
    'IsTruncated': True|False,
    'Quantity': 123,
    'Items': [
      {
        'Id': 'string',
        'Status': 'string',
        'LastModifiedTime': datetime(2015, 1, 1),
        'DomainName': 'string',
        'Aliases': {
          'Quantity': 123,
          'Items': [
            'string',
          ]
        },
        'Origins': {
```

```
'Quantity': 123,
'Items': [
    {
        'Id': 'string',
        'DomainName': 'string',
        'OriginPath': 'string',
        'S3OriginConfig': {
            'OriginAccessIdentity': 'string'
        },
        'CustomOriginConfig': {
            'HTTPPort': 123,
            'HTTPSPort': 123,
            'OriginProtocolPolicy': 'http-only'|'match-viewer'
        }
    },
],
},
'DefaultCacheBehavior': {
    'TargetOriginId': 'string',
    'ForwardedValues': {
        'QueryString': True|False,
        'Cookies': {
            'Forward': 'none'|'whitelist'|'all',
            'WhitelistedNames': {
                'Quantity': 123,
                'Items': [
                    'string',
                ]
            }
        }
    },
    'Headers': {
        'Quantity': 123,
        'Items': [
            'string',
        ]
    }
},
'TrustedSigners': {
    'Enabled': True|False,
    'Quantity': 123,
    'Items': [
        'string',
    ]
},
'ViewerProtocolPolicy': 'allow-all'|'https-only'|'redirect-to-https',
'MinTTL': 123,
'AllowedMethods': {
    'Quantity': 123,
    'Items': [
        'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
    ],
    'CachedMethods': {
        'Quantity': 123,
        'Items': [
            'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
        ]
    }
},
},
```

```

        'SmoothStreaming': True|False,
        'DefaultTTL': 123,
        'MaxTTL': 123
    },
    'CacheBehaviors': {
        'Quantity': 123,
        'Items': [
            {
                'PathPattern': 'string',
                'TargetOriginId': 'string',
                'ForwardedValues': {
                    'QueryString': True|False,
                    'Cookies': {
                        'Forward': 'none'|'whitelist'|'all',
                        'WhitelistedNames': {
                            'Quantity': 123,
                            'Items': [
                                'string',
                            ]
                        }
                    },
                },
                'Headers': {
                    'Quantity': 123,
                    'Items': [
                        'string',
                    ]
                }
            },
        ],
        'TrustedSigners': {
            'Enabled': True|False,
            'Quantity': 123,
            'Items': [
                'string',
            ]
        },
        'ViewerProtocolPolicy': 'allow-all'|'https-only'|'redirect',
        'MinTTL': 123,
        'AllowedMethods': {
            'Quantity': 123,
            'Items': [
                'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
            ],
            'CachedMethods': {
                'Quantity': 123,
                'Items': [
                    'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
                ]
            }
        },
        'SmoothStreaming': True|False,
        'DefaultTTL': 123,
        'MaxTTL': 123
    },
],
},
'CustomErrorResponses': {
    'Quantity': 123,
    'Items': [

```

```
{
    {
        'ErrorCode': 123,
        'ResponsePagePath': 'string',
        'ResponseCode': 'string',
        'ErrorCachingMinTTL': 123
    },
]
},
'Comment': 'string',
'PriceClass': 'PriceClass_100'|'PriceClass_200'|'PriceClass_All',
'Enabled': True|False,
'ViewerCertificate': {
    'IAMCertificateId': 'string',
    'CloudFrontDefaultCertificate': True|False,
    'SSLSupportMethod': 'sni-only'|'vip',
    'MinimumProtocolVersion': 'SSLv3'|'TLSv1'
},
'Restrictions': {
    'GeoRestriction': {
        'RestrictionType': 'blacklist'|'whitelist'|'none',
        'Quantity': 123,
        'Items': [
            'string',
        ]
    }
}
},
],
}
}
```

Response Structure

- (*dict*) – The returned result of the corresponding request.
 - **DistributionList** (*dict*) – The DistributionList type.
 - * **Marker** (*string*) – The value you provided for the Marker request parameter.
 - * **NextMarker** (*string*) – If IsTruncated is true, this element is present and contains the value you can use for the Marker request parameter to continue listing your distributions where they left off.
 - * **MaxItems** (*integer*) – The value you provided for the MaxItems request parameter.
 - * **IsTruncated** (*boolean*) – A flag that indicates whether more distributions remain to be listed. If your results were truncated, you can make a follow-up pagination request using the Marker request parameter to retrieve more distributions in the list.
 - * **Quantity** (*integer*) – The number of distributions that were created by the current AWS account.
 - * **Items** (*list*) – A complex type that contains one DistributionSummary element for each distribution that was created by the current AWS account.
 - (*dict*) – A summary of the information for an Amazon CloudFront distribution.
 - **Id** (*string*) – The identifier for the distribution. For example: EDFDVBD632BHDS5.
 - **Status** (*string*) – This response element indicates the current status of the distribution. When the status is Deployed, the

distribution's information is fully propagated throughout the Amazon CloudFront system.

- **LastModifiedTime** (*datetime*) – The date and time the distribution was last modified.
- **DomainName** (*string*) – The domain name corresponding to the distribution. For example: d604721fxaaqy9.cloudfront.net.
- **Aliases** (*dict*) – A complex type that contains information about CNAMEs (alternate domain names), if any, for this distribution.
- **Quantity** (*integer*) – The number of CNAMEs, if any, for this distribution.
- **Items** (*list*) – Optional: A complex type that contains CNAME elements, if any, for this distribution. If Quantity is 0, you can omit Items.
- (*string*) –
- **Origins** (*dict*) – A complex type that contains information about origins for this distribution.
- **Quantity** (*integer*) – The number of origins for this distribution.
- **Items** (*list*) – A complex type that contains origins for this distribution.
- (*dict*) – A complex type that describes the Amazon S3 bucket or the HTTP server (for example, a web server) from which CloudFront gets your files. You must create at least one origin.
- **Id** (*string*) – A unique identifier for the origin. The value of Id must be unique within the distribution. You use the value of Id when you create a cache behavior. The Id identifies the origin that CloudFront routes a request to when the request matches the path pattern for that cache behavior.
- **DomainName** (*string*) – Amazon S3 origins: The DNS name of the Amazon S3 bucket from which you want CloudFront to get objects for this origin, for example, myaws-bucket.s3.amazonaws.com. Custom origins: The DNS domain name for the HTTP server from which you want CloudFront to get objects for this origin, for example, www.example.com.
- **OriginPath** (*string*) – An optional element that causes CloudFront to request your content from a directory in your Amazon S3 bucket or your custom origin. When you include the OriginPath element, specify the directory name, beginning with a /. CloudFront appends the directory name to the value of DomainName.
- **S3OriginConfig** (*dict*) – A complex type that contains information about the Amazon S3 origin. If the origin is a custom origin, use the CustomOriginConfig element instead.
- **OriginAccessIdentity** (*string*) – The CloudFront origin access identity to associate with the origin. Use an origin access identity to configure the origin so that end users can only access objects in an Amazon S3 bucket through CloudFront. If you want end users to be able to access objects using either the CloudFront URL or the Amazon S3 URL, specify an empty OriginAccessIdentity element. To delete the origin access identity from an existing distribution, update the distri-

bution configuration and include an empty `OriginAccessIdentity` element. To replace the origin access identity, update the distribution configuration and specify the new origin access identity. Use the format `origin-access-identity/cloudfront/Id` where `Id` is the value that CloudFront returned in the `Id` element when you created the origin access identity.

- **CustomOriginConfig** (*dict*) – A complex type that contains information about a custom origin. If the origin is an Amazon S3 bucket, use the `S3OriginConfig` element instead.
- **HTTPPort** (*integer*) – The HTTP port the custom origin listens on.
- **HTTPSPort** (*integer*) – The HTTPS port the custom origin listens on.
- **OriginProtocolPolicy** (*string*) – The origin protocol policy to apply to your origin.
- **DefaultCacheBehavior** (*dict*) – A complex type that describes the default cache behavior if you do not specify a `CacheBehavior` element or if files don't match any of the values of `PathPattern` in `CacheBehavior` elements. You must create exactly one default cache behavior.
- **TargetOriginId** (*string*) – The value of ID for the origin that you want CloudFront to route requests to when a request matches the path pattern either for a cache behavior or for the default cache behavior.
- **ForwardedValues** (*dict*) – A complex type that specifies how CloudFront handles query strings, cookies and headers.
- **QueryString** (*boolean*) – Indicates whether you want CloudFront to forward query strings to the origin that is associated with this cache behavior. If so, specify `true`; if not, specify `false`.
- **Cookies** (*dict*) – A complex type that specifies how CloudFront handles cookies.
- **Forward** (*string*) – Use this element to specify whether you want CloudFront to forward cookies to the origin that is associated with this cache behavior. You can specify `all`, `none` or `whitelist`. If you choose `All`, CloudFront forwards all cookies regardless of how many your application uses.
- **WhitelistedNames** (*dict*) – A complex type that specifies the whitelisted cookies, if any, that you want CloudFront to forward to your origin that is associated with this cache behavior.
- **Quantity** (*integer*) – The number of whitelisted cookies for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains whitelisted cookies for this cache behavior. If `Quantity` is 0, you can omit `Items`.
- (*string*) –
- **Headers** (*dict*) – A complex type that specifies the Headers, if any, that you want CloudFront to vary upon for this cache behavior.
- **Quantity** (*integer*) – The number of different headers that you want CloudFront to forward to the origin and to vary on for this cache behavior. The maximum number of headers that you can specify by name is 10. If you want CloudFront to forward all headers to the origin and vary on all of them, specify

1 for Quantity and * for Name. If you don't want CloudFront to forward any additional headers to the origin or to vary on any headers, specify 0 for Quantity and omit Items.

- **Items** (*list*) – Optional: A complex type that contains a Name element for each header that you want CloudFront to forward to the origin and to vary on for this cache behavior. If Quantity is 0, omit Items.
- (*string*) –
- **TrustedSigners** (*dict*) – A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the PathPattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don't want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more trusted signers, change Enabled to true (if it's currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.
- **Enabled** (*boolean*) – Specifies whether you want to require end users to use signed URLs to access the files specified by PathPattern and TargetOriginId.
- **Quantity** (*integer*) – The number of trusted signers for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **ViewerProtocolPolicy** (*string*) – Use this element to specify the protocol that users can use to access the files in the origin specified by TargetOriginId when a request matches the path pattern in PathPattern. If you want CloudFront to allow end users to use any available protocol, specify allow-all. If you want CloudFront to require HTTPS, specify https. If you want CloudFront to respond to an HTTP request with an HTTP status code of 301 (Moved Permanently) and the HTTPS URL, specify redirect-to-https. The viewer then resubmits the request using the HTTPS URL.
- **MinTTL** (*integer*) – The minimum amount of time that you want objects to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **AllowedMethods** (*dict*) – A complex type that controls which HTTP methods CloudFront processes and forwards to your Amazon S3 bucket or your custom origin. There are three choices:
 - CloudFront forwards only GET and HEAD requests.
 - CloudFront forwards only GET, HEAD and OPTIONS requests.
 - CloudFront forwards GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests.
 If you pick the third choice, you may need to restrict access to your

Amazon S3 bucket or to your custom origin so users can't perform operations that you don't want them to. For example, you may not want users to have permission to delete objects from your origin.

- **Quantity** (*integer*) – The number of HTTP methods that you want CloudFront to forward to your origin. Valid values are 2 (for GET and HEAD requests), 3 (for GET, HEAD and OPTIONS requests) and 7 (for GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests).
- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to process and forward to your origin.
- (*string*) –
- **CachedMethods** (*dict*) – A complex type that controls whether CloudFront caches the response to requests using the specified HTTP methods. There are two choices: - CloudFront caches responses to GET and HEAD requests. - CloudFront caches responses to GET, HEAD, and OPTIONS requests. If you pick the second choice for your S3 Origin, you may need to forward Access-Control-Request-Method, Access-Control-Request-Headers and Origin headers for the responses to be cached correctly.
- **Quantity** (*integer*) – The number of HTTP methods for which you want CloudFront to cache responses. Valid values are 2 (for caching responses to GET and HEAD requests) and 3 (for caching responses to GET, HEAD, and OPTIONS requests).
- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to cache responses to.
- (*string*) –
- **SmoothStreaming** (*boolean*) – Indicates whether you want to distribute media files in Microsoft Smooth Streaming format using the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
- **DefaultTTL** (*integer*) – If you don't configure your origin to add a Cache-Control max-age directive or an Expires header, DefaultTTL is the default amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin does not add HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **MaxTTL** (*integer*) – The maximum amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin adds HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **CacheBehaviors** (*dict*) – A complex type that contains zero or more CacheBehavior elements.
- **Quantity** (*integer*) – The number of cache behaviors for this

distribution.

- **Items** (*list*) – Optional: A complex type that contains cache behaviors for this distribution. If Quantity is 0, you can omit Items.
- (*dict*) – A complex type that describes how CloudFront processes requests. You can create up to 10 cache behaviors. You must create at least as many cache behaviors (including the default cache behavior) as you have origins if you want CloudFront to distribute objects from all of the origins. Each cache behavior specifies the one origin from which you want CloudFront to get objects. If you have two origins and only the default cache behavior, the default cache behavior will cause CloudFront to get objects from one of the origins, but the other origin will never be used. If you don't want to specify any cache behaviors, include only an empty CacheBehaviors element. Don't include an empty CacheBehavior element, or CloudFront returns a MalformedXML error. To delete all cache behaviors in an existing distribution, update the distribution configuration and include only an empty CacheBehaviors element. To add, change, or remove one or more cache behaviors, update the distribution configuration and specify all of the cache behaviors that you want to include in the updated distribution.
- **PathPattern** (*string*) – The pattern (for example, images/**.jpg*) that specifies which requests you want this cache behavior to apply to. When CloudFront receives an end-user request, the requested path is compared with path patterns in the order in which cache behaviors are listed in the distribution. The path pattern for the default cache behavior is * and cannot be changed. If the request for an object does not match the path pattern for any cache behaviors, CloudFront applies the behavior in the default cache behavior.
- **TargetOriginId** (*string*) – The value of ID for the origin that you want CloudFront to route requests to when a request matches the path pattern either for a cache behavior or for the default cache behavior.
- **ForwardedValues** (*dict*) – A complex type that specifies how CloudFront handles query strings, cookies and headers.
- **QueryString** (*boolean*) – Indicates whether you want CloudFront to forward query strings to the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
- **Cookies** (*dict*) – A complex type that specifies how CloudFront handles cookies.
- **Forward** (*string*) – Use this element to specify whether you want CloudFront to forward cookies to the origin that is associated with this cache behavior. You can specify all, none or whitelist. If you choose All, CloudFront forwards all cookies regardless of how many your application uses.
- **WhitelistedNames** (*dict*) – A complex type that specifies the whitelisted cookies, if any, that you want CloudFront to forward to your origin that is associated with this cache behavior.
- **Quantity** (*integer*) – The number of whitelisted cookies for this cache behavior.

- **Items** (*list*) – Optional: A complex type that contains whitelisted cookies for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **Headers** (*dict*) – A complex type that specifies the Headers, if any, that you want CloudFront to vary upon for this cache behavior.
- **Quantity** (*integer*) – The number of different headers that you want CloudFront to forward to the origin and to vary on for this cache behavior. The maximum number of headers that you can specify by name is 10. If you want CloudFront to forward all headers to the origin and vary on all of them, specify 1 for Quantity and * for Name. If you don't want CloudFront to forward any additional headers to the origin or to vary on any headers, specify 0 for Quantity and omit Items.
- **Items** (*list*) – Optional: A complex type that contains a Name element for each header that you want CloudFront to forward to the origin and to vary on for this cache behavior. If Quantity is 0, omit Items.
- (*string*) –
- **TrustedSigners** (*dict*) – A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the PathPattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don't want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more trusted signers, change Enabled to true (if it's currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.
- **Enabled** (*boolean*) – Specifies whether you want to require end users to use signed URLs to access the files specified by PathPattern and TargetOriginId.
- **Quantity** (*integer*) – The number of trusted signers for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **ViewerProtocolPolicy** (*string*) – Use this element to specify the protocol that users can use to access the files in the origin specified by TargetOriginId when a request matches the path pattern in PathPattern. If you want CloudFront to allow end users to use any available protocol, specify allow-all. If you want CloudFront to require HTTPS, specify https. If you want CloudFront to respond to an HTTP request with an HTTP status code of 301 (Moved Permanently) and the HTTPS URL, specify redirect-to-https. The viewer then resubmits the request using the HTTPS URL.
- **MinTTL** (*integer*) – The minimum amount of time that you

want objects to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 3,153,600,000 seconds (100 years).

- **AllowedMethods** (*dict*) – A complex type that controls which HTTP methods CloudFront processes and forwards to your Amazon S3 bucket or your custom origin. There are three choices:
 - CloudFront forwards only GET and HEAD requests.
 - CloudFront forwards only GET, HEAD and OPTIONS requests.
 - CloudFront forwards GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests. If you pick the third choice, you may need to restrict access to your Amazon S3 bucket or to your custom origin so users can't perform operations that you don't want them to. For example, you may not want users to have permission to delete objects from your origin.
- **Quantity** (*integer*) – The number of HTTP methods that you want CloudFront to forward to your origin. Valid values are 2 (for GET and HEAD requests), 3 (for GET, HEAD and OPTIONS requests) and 7 (for GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests).
- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to process and forward to your origin.
- (*string*) –
- **CachedMethods** (*dict*) – A complex type that controls whether CloudFront caches the response to requests using the specified HTTP methods. There are two choices:
 - CloudFront caches responses to GET and HEAD requests.
 - CloudFront caches responses to GET, HEAD, and OPTIONS requests. If you pick the second choice for your S3 Origin, you may need to forward Access-Control-Request-Method, Access-Control-Request-Headers and Origin headers for the responses to be cached correctly.
- **Quantity** (*integer*) – The number of HTTP methods for which you want CloudFront to cache responses. Valid values are 2 (for caching responses to GET and HEAD requests) and 3 (for caching responses to GET, HEAD, and OPTIONS requests).
- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to cache responses to.
- (*string*) –
- **SmoothStreaming** (*boolean*) – Indicates whether you want to distribute media files in Microsoft Smooth Streaming format using the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
- **DefaultTTL** (*integer*) – If you don't configure your origin to add a Cache-Control max-age directive or an Expires header, DefaultTTL is the default amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin does not add HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000

seconds (100 years).

- **MaxTTL** (*integer*) – The maximum amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin adds HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **CustomErrorResponses** (*dict*) – A complex type that contains zero or more CustomErrorResponses elements.
- **Quantity** (*integer*) – The number of custom error responses for this distribution.
- **Items** (*list*) – Optional: A complex type that contains custom error responses for this distribution. If Quantity is 0, you can omit Items.
- (*dict*) – A complex type that describes how you'd prefer CloudFront to respond to requests that result in either a 4xx or 5xx response. You can control whether a custom error page should be displayed, what the desired response code should be for this error page and how long should the error response be cached by CloudFront. If you don't want to specify any custom error responses, include only an empty CustomErrorResponses element. To delete all custom error responses in an existing distribution, update the distribution configuration and include only an empty CustomErrorResponses element. To add, change, or remove one or more custom error responses, update the distribution configuration and specify all of the custom error responses that you want to include in the updated distribution.
- **ErrorCode** (*integer*) – The 4xx or 5xx HTTP status code that you want to customize. For a list of HTTP status codes that you can customize, see CloudFront documentation.
- **ResponsePagePath** (*string*) – The path of the custom error page (for example, /custom_404.html). The path is relative to the distribution and must begin with a slash (/). If the path includes any non-ASCII characters or unsafe characters as defined in RFC 1783 (<http://www.ietf.org/rfc/rfc1738.txt>), URL encode those characters. Do not URL encode any other characters in the path, or CloudFront will not return the custom error page to the viewer.
- **ResponseCode** (*string*) – The HTTP status code that you want CloudFront to return with the custom error page to the viewer. For a list of HTTP status codes that you can replace, see CloudFront Documentation.
- **ErrorCachingMinTTL** (*integer*) – The minimum amount of time you want HTTP error codes to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 31,536,000.
- **Comment** (*string*) – The comment originally specified when this distribution was created.
- **PriceClass** (*string*) –
- **Enabled** (*boolean*) – Whether the distribution is enabled to

accept end user requests for content.

- **ViewerCertificate** (*dict*) – A complex type that contains information about viewer certificates for this distribution.
- **IAMCertificateId** (*string*) – If you want viewers to use HTTPS to request your objects and you’re using an alternate domain name in your object URLs (for example, <https://example.com/logo.jpg>), specify the IAM certificate identifier of the custom viewer certificate for this distribution. Specify either this value or CloudFrontDefaultCertificate.
- **CloudFrontDefaultCertificate** (*boolean*) – If you want viewers to use HTTPS to request your objects and you’re using the CloudFront domain name of your distribution in your object URLs (for example, <https://d1111111abcdef8.cloudfront.net/logo.jpg>), set to true. Omit this value if you are setting an IAMCertificateId.
- **SSLSupportMethod** (*string*) – If you specify a value for IAMCertificateId, you must also specify how you want CloudFront to serve HTTPS requests. Valid values are vip and sni-only. If you specify vip, CloudFront uses dedicated IP addresses for your content and can respond to HTTPS requests from any viewer. However, you must request permission to use this feature, and you incur additional monthly charges. If you specify sni-only, CloudFront can only respond to HTTPS requests from viewers that support Server Name Indication (SNI). All modern browsers support SNI, but some browsers still in use don’t support SNI. Do not specify a value for SSLSupportMethod if you specified true for CloudFrontDefaultCertificate.
- **MinimumProtocolVersion** (*string*) – Specify the minimum version of the SSL protocol that you want CloudFront to use, SSLv3 or TLSv1, for HTTPS connections. CloudFront will serve your objects only to browsers or devices that support at least the SSL version that you specify. The TLSv1 protocol is more secure, so we recommend that you specify SSLv3 only if your users are using browsers or devices that don’t support TLSv1. If you’re using a custom certificate (if you specify a value for IAMCertificateId) and if you’re using dedicated IP (if you specify vip for SSLSupportMethod), you can choose SSLv3 or TLSv1 as the MinimumProtocolVersion. If you’re using a custom certificate (if you specify a value for IAMCertificateId) and if you’re using SNI (if you specify sni-only for SSLSupportMethod), you must specify TLSv1 for MinimumProtocolVersion.
- **Restrictions** (*dict*) – A complex type that identifies ways in which you want to restrict distribution of your content.
- **GeoRestriction** (*dict*) – A complex type that controls the countries in which your content is distributed. For more information about geo restriction, go to Customizing Error Responses in the Amazon CloudFront Developer Guide. CloudFront determines the location of your users using MaxMind GeoIP databases. For information about the accuracy of these databases, see How accurate are your GeoIP databases? on the MaxMind website.
- **RestrictionType** (*string*) – The method that you want to use

to restrict distribution of your content by country: - none: No geo restriction is enabled, meaning access to content is not restricted by client geo location. - blacklist: The Location elements specify the countries in which you do not want CloudFront to distribute your content. - whitelist: The Location elements specify the countries in which you want CloudFront to distribute your content.

- **Quantity** (*integer*) – When geo restriction is enabled, this is the number of countries in your whitelist or blacklist. Otherwise, when it is not enabled, Quantity is 0, and you can omit Items.
- **Items** (*list*) – A complex type that contains a Location element for each country in which you want CloudFront either to distribute your content (whitelist) or not distribute your content (blacklist). The Location element is a two-letter, uppercase country code for a country that you want to include in your blacklist or whitelist. Include one Location element for each country. CloudFront and MaxMind both use ISO 3166 country codes. For the current list of countries and the corresponding codes, see ISO 3166-1-alpha-2 code on the International Organization for Standardization website. You can also refer to the country list in the CloudFront console, which includes both country names and codes.
- (*string*) –

list_invalidations (**kwargs)

List invalidation batches.

Request Syntax

```
response = client.list_invalidations(
    DistributionId='string',
    Marker='string',
    MaxItems='string'
)
```

Parameters

- **DistributionId** (*string*) – **[REQUIRED]** The distribution's id.
- **Marker** (*string*) – Use this parameter when paginating results to indicate where to begin in your list of invalidation batches. Because the results are returned in decreasing order from most recent to oldest, the most recent results are on the first page, the second page will contain earlier results, and so on. To get the next page of results, set the Marker to the value of the NextMarker from the current page's response. This value is the same as the ID of the last invalidation batch on that page.
- **MaxItems** (*string*) – The maximum number of invalidation batches you want in the response body.

Return type dict

Returns

Response Syntax

```
{
    'InvalidationList': {
        'Marker': 'string',
        'NextMarker': 'string',
```

```

'MaxItems': 123,
'IsTruncated': True|False,
'Quantity': 123,
'Items': [
    {
        'Id': 'string',
        'CreateTime': datetime(2015, 1, 1),
        'Status': 'string'
    },
]
}
}

```

Response Structure

- (*dict*) – The returned result of the corresponding request.
 - **InvalidationList** (*dict*) – Information about invalidation batches.
 - * **Marker** (*string*) – The value you provided for the Marker request parameter.
 - * **NextMarker** (*string*) – If IsTruncated is true, this element is present and contains the value you can use for the Marker request parameter to continue listing your invalidation batches where they left off.
 - * **MaxItems** (*integer*) – The value you provided for the MaxItems request parameter.
 - * **IsTruncated** (*boolean*) – A flag that indicates whether more invalidation batch requests remain to be listed. If your results were truncated, you can make a follow-up pagination request using the Marker request parameter to retrieve more invalidation batches in the list.
 - * **Quantity** (*integer*) – The number of invalidation batches that were created by the current AWS account.
 - * **Items** (*list*) – A complex type that contains one InvalidationSummary element for each invalidation batch that was created by the current AWS account.
 - (*dict*) – Summary of an invalidation request.
 - **Id** (*string*) – The unique ID for an invalidation request.
 - **CreateTime** (*datetime*) –
 - **Status** (*string*) – The status of an invalidation request.

list_streaming_distributions (***kwargs*)

List streaming distributions.

Request Syntax

```

response = client.list_streaming_distributions(
    Marker='string',
    MaxItems='string'
)

```

Parameters

- **Marker** (*string*) – Use this when paginating results to indicate where to begin in your list of streaming distributions. The results include distributions in the list that occur after the marker. To get the next page of results, set the Marker to the value of the NextMarker from the current page's response (which is also the ID of the last distribution on that page).
- **MaxItems** (*string*) – The maximum number of streaming distributions you want in the response body.

Return type dict

Returns

Response Syntax

```
{
  'StreamingDistributionList': {
    'Marker': 'string',
    'NextMarker': 'string',
    'MaxItems': 123,
    'IsTruncated': True|False,
    'Quantity': 123,
    'Items': [
      {
        'Id': 'string',
        'Status': 'string',
        'LastModifiedTime': datetime(2015, 1, 1),
        'DomainName': 'string',
        'S3Origin': {
          'DomainName': 'string',
          'OriginAccessIdentity': 'string'
        },
        'Aliases': {
          'Quantity': 123,
          'Items': [
            'string',
          ]
        },
        'TrustedSigners': {
          'Enabled': True|False,
          'Quantity': 123,
          'Items': [
            'string',
          ]
        },
        'Comment': 'string',
        'PriceClass': 'PriceClass_100'|'PriceClass_200'|'PriceClass_All',
        'Enabled': True|False
      },
    ],
  },
}
```

Response Structure

- (*dict*) – The returned result of the corresponding request.
 - **StreamingDistributionList** (*dict*) – The StreamingDistributionList type.
 - * **Marker** (*string*) – The value you provided for the Marker request parameter.
 - * **NextMarker** (*string*) – If IsTruncated is true, this element is present and contains the value you can use for the Marker request parameter to continue listing your streaming distributions where they left off.
 - * **MaxItems** (*integer*) – The value you provided for the MaxItems request parameter.
 - * **IsTruncated** (*boolean*) – A flag that indicates whether more streaming distributions remain to be listed. If your results were truncated, you can make a follow-up pagination request using the Marker request parameter to retrieve more distributions in the list.

- * **Quantity** (*integer*) – The number of streaming distributions that were created by the current AWS account.
- * **Items** (*list*) – A complex type that contains one StreamingDistributionSummary element for each distribution that was created by the current AWS account.
 - (*dict*) – A summary of the information for an Amazon CloudFront streaming distribution.
 - **Id** (*string*) – The identifier for the distribution. For example: EDFDVBD632BHDS5.
 - **Status** (*string*) – Indicates the current status of the distribution. When the status is Deployed, the distribution’s information is fully propagated throughout the Amazon CloudFront system.
 - **LastModifiedTime** (*datetime*) – The date and time the distribution was last modified.
 - **DomainName** (*string*) – The domain name corresponding to the distribution. For example: d604721fxaaqy9.cloudfront.net.
 - **S3Origin** (*dict*) – A complex type that contains information about the Amazon S3 bucket from which you want CloudFront to get your media files for distribution.
 - **DomainName** (*string*) – The DNS name of the S3 origin.
 - **OriginAccessIdentity** (*string*) – Your S3 origin’s origin access identity.
 - **Aliases** (*dict*) – A complex type that contains information about CNAMEs (alternate domain names), if any, for this streaming distribution.
 - **Quantity** (*integer*) – The number of CNAMEs, if any, for this distribution.
 - **Items** (*list*) – Optional: A complex type that contains CNAME elements, if any, for this distribution. If Quantity is 0, you can omit Items.
 - (*string*) –
 - **TrustedSigners** (*dict*) – A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the PathPattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don’t want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more trusted signers, change Enabled to true (if it’s currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.
 - **Enabled** (*boolean*) – Specifies whether you want to require end users to use signed URLs to access the files specified by PathPattern and TargetOriginId.
 - **Quantity** (*integer*) – The number of trusted signers for this cache behavior.
 - **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit

Items.

- *(string)* –
- **Comment** (*string*) – The comment originally specified when this distribution was created.
- **PriceClass** (*string*) –
- **Enabled** (*boolean*) – Whether the distribution is enabled to accept end user requests for content.

update_cloud_front_origin_access_identity (**kwargs)

Update an origin access identity.

Request Syntax

```
response = client.update_cloud_front_origin_access_identity(
    CloudFrontOriginAccessIdentityConfig={
        'CallerReference': 'string',
        'Comment': 'string'
    },
    Id='string',
    IfMatch='string'
)
```

Parameters

- **CloudFrontOriginAccessIdentityConfig** (*dict*) – **[REQUIRED]** The identity's configuration information.
 - **CallerReference** (*string*) – **[REQUIRED]** A unique number that ensures the request can't be replayed. If the CallerReference is new (no matter the content of the CloudFrontOriginAccessIdentityConfig object), a new origin access identity is created. If the CallerReference is a value you already sent in a previous request to create an identity, and the content of the CloudFrontOriginAccessIdentityConfig is identical to the original request (ignoring white space), the response includes the same information returned to the original request. If the CallerReference is a value you already sent in a previous request to create an identity but the content of the CloudFrontOriginAccessIdentityConfig is different from the original request, CloudFront returns a CloudFrontOriginAccessIdentityAlreadyExists error.
 - **Comment** (*string*) – **[REQUIRED]** Any comments you want to include about the origin access identity.
- **Id** (*string*) – **[REQUIRED]** The identity's id.
- **IfMatch** (*string*) – The value of the ETag header you received when retrieving the identity's configuration. For example: E2QWRUHAPOMQZL.

Return type dict

Returns

Response Syntax

```
{
    'CloudFrontOriginAccessIdentity': {
        'Id': 'string',
        'S3CanonicalUserId': 'string',
        'CloudFrontOriginAccessIdentityConfig': {
            'CallerReference': 'string',
            'Comment': 'string'
        }
    },
}
```

```

    'ETag': 'string'
}

```

Response Structure

- (*dict*) – The returned result of the corresponding request.
 - **CloudFrontOriginAccessIdentity** (*dict*) – The origin access identity’s information.
 - * **Id** (*string*) – The ID for the origin access identity. For example: E74FTE3AJFJ256A.
 - * **S3CanonicalUserId** (*string*) – The Amazon S3 canonical user ID for the origin access identity, which you use when giving the origin access identity read permission to an object in Amazon S3.
 - * **CloudFrontOriginAccessIdentityConfig** (*dict*) – The current configuration information for the identity.
 - **CallerReference** (*string*) – A unique number that ensures the request can’t be replayed. If the CallerReference is new (no matter the content of the CloudFrontOriginAccessIdentityConfig object), a new origin access identity is created. If the CallerReference is a value you already sent in a previous request to create an identity, and the content of the CloudFrontOriginAccessIdentityConfig is identical to the original request (ignoring white space), the response includes the same information returned to the original request. If the CallerReference is a value you already sent in a previous request to create an identity but the content of the CloudFrontOriginAccessIdentityConfig is different from the original request, CloudFront returns a CloudFrontOriginAccessIdentityAlreadyExists error.
 - **Comment** (*string*) – Any comments you want to include about the origin access identity.
 - **ETag** (*string*) – The current version of the configuration. For example: E2QWRUHAPOMQZL.

update_distribution (**kwargs)

Update a distribution.

Request Syntax

```

response = client.update_distribution(
    DistributionConfig={
        'CallerReference': 'string',
        'Aliases': {
            'Quantity': 123,
            'Items': [
                'string',
            ]
        },
        'DefaultRootObject': 'string',
        'Origins': {
            'Quantity': 123,
            'Items': [
                {
                    'Id': 'string',
                    'DomainName': 'string',
                    'OriginPath': 'string',
                    'S3OriginConfig': {

```

```
        'OriginAccessIdentity': 'string'
    },
    'CustomOriginConfig': {
        'HTTPPort': 123,
        'HTTPSPort': 123,
        'OriginProtocolPolicy': 'http-only'|'match-viewer'
    }
},
]
},
'DefaultCacheBehavior': {
    'TargetOriginId': 'string',
    'ForwardedValues': {
        'QueryString': True|False,
        'Cookies': {
            'Forward': 'none'|'whitelist'|'all',
            'WhitelistedNames': {
                'Quantity': 123,
                'Items': [
                    'string',
                ]
            }
        }
    },
    'Headers': {
        'Quantity': 123,
        'Items': [
            'string',
        ]
    }
},
'TrustedSigners': {
    'Enabled': True|False,
    'Quantity': 123,
    'Items': [
        'string',
    ]
},
'ViewerProtocolPolicy': 'allow-all'|'https-only'|'redirect-to-https',
'MinTTL': 123,
'AllowedMethods': {
    'Quantity': 123,
    'Items': [
        'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
    ],
    'CachedMethods': {
        'Quantity': 123,
        'Items': [
            'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
        ]
    }
},
'SmoothStreaming': True|False,
'DefaultTTL': 123,
'MaxTTL': 123
},
'CacheBehaviors': {
    'Quantity': 123,
    'Items': [
```



```

{
    'PathPattern': 'string',
    'TargetOriginId': 'string',
    'ForwardedValues': {
        'QueryString': True|False,
        'Cookies': {
            'Forward': 'none'|'whitelist'|'all',
            'WhitelistedNames': {
                'Quantity': 123,
                'Items': [
                    'string',
                ]
            }
        },
        'Headers': {
            'Quantity': 123,
            'Items': [
                'string',
            ]
        }
    },
    'TrustedSigners': {
        'Enabled': True|False,
        'Quantity': 123,
        'Items': [
            'string',
        ]
    },
    'ViewerProtocolPolicy': 'allow-all'|'https-only'|'redirect-to-https',
    'MinTTL': 123,
    'AllowedMethods': {
        'Quantity': 123,
        'Items': [
            'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
        ],
        'CachedMethods': {
            'Quantity': 123,
            'Items': [
                'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
            ]
        }
    },
    'SmoothStreaming': True|False,
    'DefaultTTL': 123,
    'MaxTTL': 123
},
]
},
'CustomErrorResponses': {
    'Quantity': 123,
    'Items': [
        {
            'ErrorCode': 123,
            'ResponsePagePath': 'string',
            'ResponseCode': 'string',
            'ErrorCachingMinTTL': 123
        },
    ]
}
]

```

```
    },
    'Comment': 'string',
    'Logging': {
        'Enabled': True|False,
        'IncludeCookies': True|False,
        'Bucket': 'string',
        'Prefix': 'string'
    },
    'PriceClass': 'PriceClass_100'|'PriceClass_200'|'PriceClass_All',
    'Enabled': True|False,
    'ViewerCertificate': {
        'IAMCertificateId': 'string',
        'CloudFrontDefaultCertificate': True|False,
        'SSLSupportMethod': 'sni-only'|'vip',
        'MinimumProtocolVersion': 'SSLv3'|'TLSv1'
    },
    'Restrictions': {
        'GeoRestriction': {
            'RestrictionType': 'blacklist'|'whitelist'|'none',
            'Quantity': 123,
            'Items': [
                'string',
            ]
        }
    },
    Id='string',
    IfMatch='string'
)
```

Parameters

- **DistributionConfig** (*dict*) – **[REQUIRED]** The distribution’s configuration information.
 - **CallerReference** (*string*) – **[REQUIRED]** A unique number that ensures the request can’t be replayed. If the CallerReference is new (no matter the content of the DistributionConfig object), a new distribution is created. If the CallerReference is a value you already sent in a previous request to create a distribution, and the content of the DistributionConfig is identical to the original request (ignoring white space), the response includes the same information returned to the original request. If the CallerReference is a value you already sent in a previous request to create a distribution but the content of the DistributionConfig is different from the original request, CloudFront returns a DistributionAlreadyExists error.
 - **Aliases** (*dict*) – A complex type that contains information about CNAMEs (alternate domain names), if any, for this distribution.
 - * **Quantity** (*integer*) – **[REQUIRED]** The number of CNAMEs, if any, for this distribution.
 - * **Items** (*list*) – Optional: A complex type that contains CNAME elements, if any, for this distribution. If Quantity is 0, you can omit Items.
 - (*string*) –
 - **DefaultRootObject** (*string*) – The object that you want CloudFront to return (for example, index.html) when an end user requests the root URL for your distribution (<http://www.example.com>) instead of an object in your distribution (<http://www.example.com/index.html>). Specifying a default root object avoids exposing the contents of your distribution. If you don’t

want to specify a default root object when you create a distribution, include an empty `DefaultRootObject` element. To delete the default root object from an existing distribution, update the distribution configuration and include an empty `DefaultRootObject` element. To replace the default root object, update the distribution configuration and specify the new object.

- **Origins** (*dict*) – **[REQUIRED]** A complex type that contains information about origins for this distribution.
 - * **Quantity** (*integer*) – **[REQUIRED]** The number of origins for this distribution.
 - * **Items** (*list*) – A complex type that contains origins for this distribution.
 - (*dict*) – A complex type that describes the Amazon S3 bucket or the HTTP server (for example, a web server) from which CloudFront gets your files. You must create at least one origin.
 - **Id** (*string*) – **[REQUIRED]** A unique identifier for the origin. The value of `Id` must be unique within the distribution. You use the value of `Id` when you create a cache behavior. The `Id` identifies the origin that CloudFront routes a request to when the request matches the path pattern for that cache behavior.
 - **DomainName** (*string*) – **[REQUIRED]** Amazon S3 origins: The DNS name of the Amazon S3 bucket from which you want CloudFront to get objects for this origin, for example, `myawsbucket.s3.amazonaws.com`. Custom origins: The DNS domain name for the HTTP server from which you want CloudFront to get objects for this origin, for example, `www.example.com`.
 - **OriginPath** (*string*) – An optional element that causes CloudFront to request your content from a directory in your Amazon S3 bucket or your custom origin. When you include the `OriginPath` element, specify the directory name, beginning with a `/`. CloudFront appends the directory name to the value of `DomainName`.
 - **S3OriginConfig** (*dict*) – A complex type that contains information about the Amazon S3 origin. If the origin is a custom origin, use the `CustomOriginConfig` element instead.
 - **OriginAccessIdentity** (*string*) – **[REQUIRED]** The CloudFront origin access identity to associate with the origin. Use an origin access identity to configure the origin so that end users can only access objects in an Amazon S3 bucket through CloudFront. If you want end users to be able to access objects using either the CloudFront URL or the Amazon S3 URL, specify an empty `OriginAccessIdentity` element. To delete the origin access identity from an existing distribution, update the distribution configuration and include an empty `OriginAccessIdentity` element. To replace the origin access identity, update the distribution configuration and specify the new origin access identity. Use the format `origin-access-identity/cloudfront/Id` where `Id` is the value that CloudFront returned in the `Id` element when you created the origin access identity.
 - **CustomOriginConfig** (*dict*) – A complex type that contains information about a custom origin. If the origin is an Amazon S3 bucket, use the `S3OriginConfig` element instead.
 - **HTTPPort** (*integer*) – **[REQUIRED]** The HTTP port the

- custom origin listens on.
 - **HTTPSPort** (*integer*) – **[REQUIRED]** The HTTPS port the custom origin listens on.
 - **OriginProtocolPolicy** (*string*) – **[REQUIRED]** The origin protocol policy to apply to your origin.
- **DefaultCacheBehavior** (*dict*) – **[REQUIRED]** A complex type that describes the default cache behavior if you do not specify a CacheBehavior element or if files don't match any of the values of PathPattern in CacheBehavior elements. You must create exactly one default cache behavior.
 - * **TargetOriginId** (*string*) – **[REQUIRED]** The value of ID for the origin that you want CloudFront to route requests to when a request matches the path pattern either for a cache behavior or for the default cache behavior.
 - * **ForwardedValues** (*dict*) – **[REQUIRED]** A complex type that specifies how CloudFront handles query strings, cookies and headers.
 - **QueryString** (*boolean*) – **[REQUIRED]** Indicates whether you want CloudFront to forward query strings to the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
 - **Cookies** (*dict*) – **[REQUIRED]** A complex type that specifies how CloudFront handles cookies.
 - **Forward** (*string*) – **[REQUIRED]** Use this element to specify whether you want CloudFront to forward cookies to the origin that is associated with this cache behavior. You can specify all, none or whitelist. If you choose All, CloudFront forwards all cookies regardless of how many your application uses.
 - **WhitelistedNames** (*dict*) – A complex type that specifies the whitelisted cookies, if any, that you want CloudFront to forward to your origin that is associated with this cache behavior.
 - **Quantity** (*integer*) – **[REQUIRED]** The number of whitelisted cookies for this cache behavior.
 - **Items** (*list*) – Optional: A complex type that contains whitelisted cookies for this cache behavior. If Quantity is 0, you can omit Items.
 - (*string*) –
 - **Headers** (*dict*) – A complex type that specifies the Headers, if any, that you want CloudFront to vary upon for this cache behavior.
 - **Quantity** (*integer*) – **[REQUIRED]** The number of different headers that you want CloudFront to forward to the origin and to vary on for this cache behavior. The maximum number of headers that you can specify by name is 10. If you want CloudFront to forward all headers to the origin and vary on all of them, specify 1 for Quantity and * for Name. If you don't want CloudFront to forward any additional headers to the origin or to vary on any headers, specify 0 for Quantity and omit Items.
 - **Items** (*list*) – Optional: A complex type that contains a Name element for each header that you want CloudFront to forward to the origin and to vary on for this cache behavior. If Quantity is 0, omit Items.
 - (*string*) –

- * **TrustedSigners** (*dict*) – **[REQUIRED]** A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the PathPattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don't want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more trusted signers, change Enabled to true (if it's currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.
 - **Enabled** (*boolean*) – **[REQUIRED]** Specifies whether you want to require end users to use signed URLs to access the files specified by PathPattern and TargetOriginId.
 - **Quantity** (*integer*) – **[REQUIRED]** The number of trusted signers for this cache behavior.
 - **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit Items.
 - (*string*) –
- * **ViewerProtocolPolicy** (*string*) – **[REQUIRED]** Use this element to specify the protocol that users can use to access the files in the origin specified by TargetOriginId when a request matches the path pattern in PathPattern. If you want CloudFront to allow end users to use any available protocol, specify allow-all. If you want CloudFront to require HTTPS, specify https. If you want CloudFront to respond to an HTTP request with an HTTP status code of 301 (Moved Permanently) and the HTTPS URL, specify redirect-to-https. The viewer then resubmits the request using the HTTPS URL.
- * **MinTTL** (*integer*) – **[REQUIRED]** The minimum amount of time that you want objects to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- * **AllowedMethods** (*dict*) – A complex type that controls which HTTP methods CloudFront processes and forwards to your Amazon S3 bucket or your custom origin. There are three choices: - CloudFront forwards only GET and HEAD requests. - CloudFront forwards only GET, HEAD and OPTIONS requests. - CloudFront forwards GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests. If you pick the third choice, you may need to restrict access to your Amazon S3 bucket or to your custom origin so users can't perform operations that you don't want them to. For example, you may not want users to have permission to delete objects from your origin.
 - **Quantity** (*integer*) – **[REQUIRED]** The number of HTTP methods that you want CloudFront to forward to your origin. Valid values are 2 (for GET and HEAD requests), 3 (for GET, HEAD and OPTIONS requests) and 7 (for GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests).
 - **Items** (*list*) – **[REQUIRED]** A complex type that contains the HTTP methods that you want CloudFront to process and

- forward to your origin.
- *(string)* –
- **CachedMethods** (*dict*) – A complex type that controls whether CloudFront caches the response to requests using the specified HTTP methods. There are two choices: - CloudFront caches responses to GET and HEAD requests. - CloudFront caches responses to GET, HEAD, and OPTIONS requests. If you pick the second choice for your S3 Origin, you may need to forward Access-Control-Request-Method, Access-Control-Request-Headers and Origin headers for the responses to be cached correctly.
- **Quantity** (*integer*) – **[REQUIRED]** The number of HTTP methods for which you want CloudFront to cache responses. Valid values are 2 (for caching responses to GET and HEAD requests) and 3 (for caching responses to GET, HEAD, and OPTIONS requests).
- **Items** (*list*) – **[REQUIRED]** A complex type that contains the HTTP methods that you want CloudFront to cache responses to.
- *(string)* –
- * **SmoothStreaming** (*boolean*) – Indicates whether you want to distribute media files in Microsoft Smooth Streaming format using the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
- * **DefaultTTL** (*integer*) – If you don't configure your origin to add a Cache-Control max-age directive or an Expires header, DefaultTTL is the default amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin does not add HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- * **MaxTTL** (*integer*) – The maximum amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin adds HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **CacheBehaviors** (*dict*) – A complex type that contains zero or more CacheBehavior elements.
 - * **Quantity** (*integer*) – **[REQUIRED]** The number of cache behaviors for this distribution.
 - * **Items** (*list*) – Optional: A complex type that contains cache behaviors for this distribution. If Quantity is 0, you can omit Items.
 - (*dict*) – A complex type that describes how CloudFront processes requests. You can create up to 10 cache behaviors. You must create at least as many cache behaviors (including the default cache behavior) as you have origins if you want CloudFront to distribute objects from all of the origins. Each cache behavior specifies the one origin from which you want CloudFront to get objects. If you have two origins and only the default cache behavior, the default cache behavior

will cause CloudFront to get objects from one of the origins, but the other origin will never be used. If you don't want to specify any cache behaviors, include only an empty `CacheBehaviors` element. Don't include an empty `CacheBehavior` element, or CloudFront returns a `MalformedXML` error. To delete all cache behaviors in an existing distribution, update the distribution configuration and include only an empty `CacheBehaviors` element. To add, change, or remove one or more cache behaviors, update the distribution configuration and specify all of the cache behaviors that you want to include in the updated distribution.

- **PathPattern** (*string*) – **[REQUIRED]** The pattern (for example, `images/*.jpg`) that specifies which requests you want this cache behavior to apply to. When CloudFront receives an end-user request, the requested path is compared with path patterns in the order in which cache behaviors are listed in the distribution. The path pattern for the default cache behavior is `*` and cannot be changed. If the request for an object does not match the path pattern for any cache behaviors, CloudFront applies the behavior in the default cache behavior.
- **TargetOriginId** (*string*) – **[REQUIRED]** The value of ID for the origin that you want CloudFront to route requests to when a request matches the path pattern either for a cache behavior or for the default cache behavior.
- **ForwardedValues** (*dict*) – **[REQUIRED]** A complex type that specifies how CloudFront handles query strings, cookies and headers.
- **QueryString** (*boolean*) – **[REQUIRED]** Indicates whether you want CloudFront to forward query strings to the origin that is associated with this cache behavior. If so, specify `true`; if not, specify `false`.
- **Cookies** (*dict*) – **[REQUIRED]** A complex type that specifies how CloudFront handles cookies.
- **Forward** (*string*) – **[REQUIRED]** Use this element to specify whether you want CloudFront to forward cookies to the origin that is associated with this cache behavior. You can specify `all`, `none` or `whitelist`. If you choose `All`, CloudFront forwards all cookies regardless of how many your application uses.
- **WhitelistedNames** (*dict*) – A complex type that specifies the whitelisted cookies, if any, that you want CloudFront to forward to your origin that is associated with this cache behavior.
- **Quantity** (*integer*) – **[REQUIRED]** The number of whitelisted cookies for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains whitelisted cookies for this cache behavior. If `Quantity` is 0, you can omit `Items`.
- (*string*) –
- **Headers** (*dict*) – A complex type that specifies the Headers, if any, that you want CloudFront to vary upon for this cache behavior.
- **Quantity** (*integer*) – **[REQUIRED]** The number of different headers that you want CloudFront to forward to the origin and to vary on for this cache behavior. The maximum number

of headers that you can specify by name is 10. If you want CloudFront to forward all headers to the origin and vary on all of them, specify 1 for Quantity and * for Name. If you don't want CloudFront to forward any additional headers to the origin or to vary on any headers, specify 0 for Quantity and omit Items.

- **Items** (*list*) – Optional: A complex type that contains a Name element for each header that you want CloudFront to forward to the origin and to vary on for this cache behavior. If Quantity is 0, omit Items.
- (*string*) –
- **TrustedSigners** (*dict*) – **[REQUIRED]** A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the PathPattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don't want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more trusted signers, change Enabled to true (if it's currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.
- **Enabled** (*boolean*) – **[REQUIRED]** Specifies whether you want to require end users to use signed URLs to access the files specified by PathPattern and TargetOriginId.
- **Quantity** (*integer*) – **[REQUIRED]** The number of trusted signers for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **ViewerProtocolPolicy** (*string*) – **[REQUIRED]** Use this element to specify the protocol that users can use to access the files in the origin specified by TargetOriginId when a request matches the path pattern in PathPattern. If you want CloudFront to allow end users to use any available protocol, specify allow-all. If you want CloudFront to require HTTPS, specify https. If you want CloudFront to respond to an HTTP request with an HTTP status code of 301 (Moved Permanently) and the HTTPS URL, specify redirect-to-https. The viewer then resubmits the request using the HTTPS URL.
- **MinTTL** (*integer*) – **[REQUIRED]** The minimum amount of time that you want objects to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **AllowedMethods** (*dict*) – A complex type that controls which HTTP methods CloudFront processes and forwards to your Amazon S3 bucket or your custom origin. There are three choices: - CloudFront forwards only GET and HEAD re-

quests. - CloudFront forwards only GET, HEAD and OPTIONS requests. - CloudFront forwards GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests. If you pick the third choice, you may need to restrict access to your Amazon S3 bucket or to your custom origin so users can't perform operations that you don't want them to. For example, you may not want users to have permission to delete objects from your origin.

- **Quantity** (*integer*) – **[REQUIRED]** The number of HTTP methods that you want CloudFront to forward to your origin. Valid values are 2 (for GET and HEAD requests), 3 (for GET, HEAD and OPTIONS requests) and 7 (for GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests).
- **Items** (*list*) – **[REQUIRED]** A complex type that contains the HTTP methods that you want CloudFront to process and forward to your origin.
- (*string*) –
- **CachedMethods** (*dict*) – A complex type that controls whether CloudFront caches the response to requests using the specified HTTP methods. There are two choices: - CloudFront caches responses to GET and HEAD requests. - CloudFront caches responses to GET, HEAD, and OPTIONS requests. If you pick the second choice for your S3 Origin, you may need to forward Access-Control-Request-Method, Access-Control-Request-Headers and Origin headers for the responses to be cached correctly.
- **Quantity** (*integer*) – **[REQUIRED]** The number of HTTP methods for which you want CloudFront to cache responses. Valid values are 2 (for caching responses to GET and HEAD requests) and 3 (for caching responses to GET, HEAD, and OPTIONS requests).
- **Items** (*list*) – **[REQUIRED]** A complex type that contains the HTTP methods that you want CloudFront to cache responses to.
- (*string*) –
- **SmoothStreaming** (*boolean*) – Indicates whether you want to distribute media files in Microsoft Smooth Streaming format using the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
- **DefaultTTL** (*integer*) – If you don't configure your origin to add a Cache-Control max-age directive or an Expires header, DefaultTTL is the default amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin does not add HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **MaxTTL** (*integer*) – The maximum amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin adds HTTP headers

- such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **CustomErrorResponses** (*dict*) – A complex type that contains zero or more CustomErrorResponse elements.
 - * **Quantity** (*integer*) – **[REQUIRED]** The number of custom error responses for this distribution.
 - * **Items** (*list*) – Optional: A complex type that contains custom error responses for this distribution. If Quantity is 0, you can omit Items.
 - (*dict*) – A complex type that describes how you'd prefer CloudFront to respond to requests that result in either a 4xx or 5xx response. You can control whether a custom error page should be displayed, what the desired response code should be for this error page and how long should the error response be cached by CloudFront. If you don't want to specify any custom error responses, include only an empty CustomErrorResponses element. To delete all custom error responses in an existing distribution, update the distribution configuration and include only an empty CustomErrorResponses element. To add, change, or remove one or more custom error responses, update the distribution configuration and specify all of the custom error responses that you want to include in the updated distribution.
 - **ErrorCode** (*integer*) – **[REQUIRED]** The 4xx or 5xx HTTP status code that you want to customize. For a list of HTTP status codes that you can customize, see CloudFront documentation.
 - **ResponsePagePath** (*string*) – The path of the custom error page (for example, /custom_404.html). The path is relative to the distribution and must begin with a slash (/). If the path includes any non-ASCII characters or unsafe characters as defined in RFC 1783 (<http://www.ietf.org/rfc/rfc1738.txt>), URL encode those characters. Do not URL encode any other characters in the path, or CloudFront will not return the custom error page to the viewer.
 - **ResponseCode** (*string*) – The HTTP status code that you want CloudFront to return with the custom error page to the viewer. For a list of HTTP status codes that you can replace, see CloudFront Documentation.
 - **ErrorCachingMinTTL** (*integer*) – The minimum amount of time you want HTTP error codes to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 31,536,000.
 - **Comment** (*string*) – **[REQUIRED]** Any comments you want to include about the distribution.
 - **Logging** (*dict*) – A complex type that controls whether access logs are written for the distribution.
 - * **Enabled** (*boolean*) – **[REQUIRED]** Specifies whether you want CloudFront to save access logs to an Amazon S3 bucket. If you do not want to enable logging when you create a distribution or if you want to disable logging for an existing distribution, specify false for Enabled, and specify empty Bucket and Prefix elements. If you specify false for Enabled but you specify values for Bucket, prefix

and IncludeCookies, the values are automatically deleted.

- * **IncludeCookies** (*boolean*) – **[REQUIRED]** Specifies whether you want CloudFront to include cookies in access logs, specify true for IncludeCookies. If you choose to include cookies in logs, CloudFront logs all cookies regardless of how you configure the cache behaviors for this distribution. If you do not want to include cookies when you create a distribution or if you want to disable include cookies for an existing distribution, specify false for IncludeCookies.
- * **Bucket** (*string*) – **[REQUIRED]** The Amazon S3 bucket to store the access logs in, for example, myawslog-bucket.s3.amazonaws.com.
- * **Prefix** (*string*) – **[REQUIRED]** An optional string that you want CloudFront to prefix to the access log filenames for this distribution, for example, myprefix/. If you want to enable logging, but you do not want to specify a prefix, you still must include an empty Prefix element in the Logging element.
- **PriceClass** (*string*) – A complex type that contains information about price class for this distribution.
- **Enabled** (*boolean*) – **[REQUIRED]** Whether the distribution is enabled to accept end user requests for content.
- **ViewerCertificate** (*dict*) – A complex type that contains information about viewer certificates for this distribution.
 - * **IAMCertificateId** (*string*) – If you want viewers to use HTTPS to request your objects and you’re using an alternate domain name in your object URLs (for example, <https://example.com/logo.jpg>), specify the IAM certificate identifier of the custom viewer certificate for this distribution. Specify either this value or CloudFrontDefaultCertificate.
 - * **CloudFrontDefaultCertificate** (*boolean*) – If you want viewers to use HTTPS to request your objects and you’re using the CloudFront domain name of your distribution in your object URLs (for example, <https://d1111111abcdef8.cloudfront.net/logo.jpg>), set to true. Omit this value if you are setting an IAMCertificateId.
 - * **SSLSupportMethod** (*string*) – If you specify a value for IAMCertificateId, you must also specify how you want CloudFront to serve HTTPS requests. Valid values are vip and sni-only. If you specify vip, CloudFront uses dedicated IP addresses for your content and can respond to HTTPS requests from any viewer. However, you must request permission to use this feature, and you incur additional monthly charges. If you specify sni-only, CloudFront can only respond to HTTPS requests from viewers that support Server Name Indication (SNI). All modern browsers support SNI, but some browsers still in use don’t support SNI. Do not specify a value for SSLSupportMethod if you specified true for CloudFrontDefaultCertificate.
 - * **MinimumProtocolVersion** (*string*) – Specify the minimum version of the SSL protocol that you want CloudFront to use, SSLv3 or TLSv1, for HTTPS connections. CloudFront will serve your objects only to browsers or devices that support at least the SSL version that you specify. The TLSv1 protocol is more secure, so we recommend that you specify SSLv3 only if your users are using browsers or devices that don’t support TLSv1. If you’re using a custom certificate (if you specify a value for IAMCertificateId) and if you’re

using dedicated IP (if you specify `vip` for `SSLSupportMethod`), you can choose `SSLv3` or `TLSv1` as the `MinimumProtocolVersion`. If you're using a custom certificate (if you specify a value for `IAMCertificateId`) and if you're using SNI (if you specify `sni-only` for `SSLSupportMethod`), you must specify `TLSv1` for `MinimumProtocolVersion`.

– **Restrictions** (*dict*) – A complex type that identifies ways in which you want to restrict distribution of your content.

* **GeoRestriction** (*dict*) – **[REQUIRED]** A complex type that controls the countries in which your content is distributed. For more information about geo restriction, go to Customizing Error Responses in the Amazon CloudFront Developer Guide. CloudFront determines the location of your users using MaxMind GeoIP databases. For information about the accuracy of these databases, see [How accurate are your GeoIP databases?](#) on the MaxMind website.

· **RestrictionType** (*string*) – **[REQUIRED]** The method that you want to use to restrict distribution of your content by country: - `none`: No geo restriction is enabled, meaning access to content is not restricted by client geo location. - `blacklist`: The `Location` elements specify the countries in which you do not want CloudFront to distribute your content. - `whitelist`: The `Location` elements specify the countries in which you want CloudFront to distribute your content.

· **Quantity** (*integer*) – **[REQUIRED]** When geo restriction is enabled, this is the number of countries in your whitelist or blacklist. Otherwise, when it is not enabled, `Quantity` is 0, and you can omit `Items`.

· **Items** (*list*) – A complex type that contains a `Location` element for each country in which you want CloudFront either to distribute your content (whitelist) or not distribute your content (blacklist). The `Location` element is a two-letter, uppercase country code for a country that you want to include in your blacklist or whitelist. Include one `Location` element for each country. CloudFront and MaxMind both use ISO 3166 country codes. For the current list of countries and the corresponding codes, see ISO 3166-1-alpha-2 code on the International Organization for Standardization website. You can also refer to the country list in the CloudFront console, which includes both country names and codes.

· (*string*) –

• **Id** (*string*) – **[REQUIRED]** The distribution's id.

• **IfMatch** (*string*) – The value of the `ETag` header you received when retrieving the distribution's configuration. For example: `E2QWRUHAPOMQZL`.

Return type `dict`

Returns

Response Syntax

```
{
  'Distribution': {
    'Id': 'string',
    'Status': 'string',
    'LastModifiedTime': datetime(2015, 1, 1),
    'InProgressInvalidationBatches': 123,
    'DomainName': 'string',
```

```

'ActiveTrustedSigners': {
    'Enabled': True|False,
    'Quantity': 123,
    'Items': [
        {
            'AwsAccountNumber': 'string',
            'KeyPairIds': {
                'Quantity': 123,
                'Items': [
                    'string',
                ]
            },
        },
    ],
},
'DistributionConfig': {
    'CallerReference': 'string',
    'Aliases': {
        'Quantity': 123,
        'Items': [
            'string',
        ]
    },
    'DefaultRootObject': 'string',
    'Origins': {
        'Quantity': 123,
        'Items': [
            {
                'Id': 'string',
                'DomainName': 'string',
                'OriginPath': 'string',
                'S3OriginConfig': {
                    'OriginAccessIdentity': 'string'
                },
                'CustomOriginConfig': {
                    'HTTPPort': 123,
                    'HTTPSPort': 123,
                    'OriginProtocolPolicy': 'http-only'|'match-viewer'
                }
            },
        ],
    },
    'DefaultCacheBehavior': {
        'TargetOriginId': 'string',
        'ForwardedValues': {
            'QueryString': True|False,
            'Cookies': {
                'Forward': 'none'|'whitelist'|'all',
                'WhitelistedNames': {
                    'Quantity': 123,
                    'Items': [
                        'string',
                    ]
                }
            },
        },
        'Headers': {
            'Quantity': 123,
            'Items': [

```

```
        'string',
    ]
}
},
'TrustedSigners': {
    'Enabled': True|False,
    'Quantity': 123,
    'Items': [
        'string',
    ]
},
'ViewerProtocolPolicy': 'allow-all'|'https-only'|'redirect-to-http',
'MinTTL': 123,
'AllowedMethods': {
    'Quantity': 123,
    'Items': [
        'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
    ],
    'CachedMethods': {
        'Quantity': 123,
        'Items': [
            'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
        ]
    }
},
'SmoothStreaming': True|False,
'DefaultTTL': 123,
'MaxTTL': 123
},
'CacheBehaviors': {
    'Quantity': 123,
    'Items': [
        {
            'PathPattern': 'string',
            'TargetOriginId': 'string',
            'ForwardedValues': {
                'QueryString': True|False,
                'Cookies': {
                    'Forward': 'none'|'whitelist'|'all',
                    'WhitelistedNames': {
                        'Quantity': 123,
                        'Items': [
                            'string',
                        ]
                    }
                }
            },
            'Headers': {
                'Quantity': 123,
                'Items': [
                    'string',
                ]
            }
        },
        'TrustedSigners': {
            'Enabled': True|False,
            'Quantity': 123,
            'Items': [
                'string',
```

```

    ],
    'ViewerProtocolPolicy': 'allow-all'|'https-only'|'redirect',
    'MinTTL': 123,
    'AllowedMethods': {
        'Quantity': 123,
        'Items': [
            'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE'
        ],
        'CachedMethods': {
            'Quantity': 123,
            'Items': [
                'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE'
            ]
        }
    },
    'SmoothStreaming': True|False,
    'DefaultTTL': 123,
    'MaxTTL': 123
},
]
},
'CustomErrorResponses': {
    'Quantity': 123,
    'Items': [
        {
            'ErrorCode': 123,
            'ResponsePagePath': 'string',
            'ResponseCode': 'string',
            'ErrorCachingMinTTL': 123
        }
    ],
},
],
'Comment': 'string',
'Logging': {
    'Enabled': True|False,
    'IncludeCookies': True|False,
    'Bucket': 'string',
    'Prefix': 'string'
},
'PriceClass': 'PriceClass_100'|'PriceClass_200'|'PriceClass_All',
'Enabled': True|False,
'ViewerCertificate': {
    'IAMCertificateId': 'string',
    'CloudFrontDefaultCertificate': True|False,
    'SSLSupportMethod': 'sni-only'|'vip',
    'MinimumProtocolVersion': 'SSLv3'|'TLSv1'
},
'Restrictions': {
    'GeoRestriction': {
        'RestrictionType': 'blacklist'|'whitelist'|'none',
        'Quantity': 123,
        'Items': [
            'string',
        ]
    }
}
}

```

```
    },  
    'ETag': 'string'  
}
```

Response Structure

- (*dict*) – The returned result of the corresponding request.
 - **Distribution** (*dict*) – The distribution’s information.
 - * **Id** (*string*) – The identifier for the distribution. For example: EDFD-VBD632BHDS5.
 - * **Status** (*string*) – This response element indicates the current status of the distribution. When the status is Deployed, the distribution’s information is fully propagated throughout the Amazon CloudFront system.
 - * **LastModifiedTime** (*datetime*) – The date and time the distribution was last modified.
 - * **InProgressInvalidationBatches** (*integer*) – The number of invalidation batches currently in progress.
 - * **DomainName** (*string*) – The domain name corresponding to the distribution. For example: d604721fxaaqy9.cloudfront.net.
 - * **ActiveTrustedSigners** (*dict*) – CloudFront automatically adds this element to the response only if you’ve set up the distribution to serve private content with signed URLs. The element lists the key pair IDs that CloudFront is aware of for each trusted signer. The Signer child element lists the AWS account number of the trusted signer (or an empty Self element if the signer is you). The Signer element also includes the IDs of any active key pairs associated with the trusted signer’s AWS account. If no KeyPairId element appears for a Signer, that signer can’t create working signed URLs.
 - **Enabled** (*boolean*) – Each active trusted signer.
 - **Quantity** (*integer*) – The number of unique trusted signers included in all cache behaviors. For example, if three cache behaviors all list the same three AWS accounts, the value of Quantity for ActiveTrustedSigners will be 3.
 - **Items** (*list*) – A complex type that contains one Signer complex type for each unique trusted signer that is specified in the TrustedSigners complex type, including trusted signers in the default cache behavior and in all of the other cache behaviors.
 - (*dict*) – A complex type that lists the AWS accounts that were included in the TrustedSigners complex type, as well as their active CloudFront key pair IDs, if any.
 - **AwsAccountNumber** (*string*) – Specifies an AWS account that can create signed URLs. Values: self, which indicates that the AWS account that was used to create the distribution can create signed URLs, or an AWS account number. Omit the dashes in the account number.
 - **KeyPairIds** (*dict*) – A complex type that lists the active CloudFront key pairs, if any, that are associated with AwsAccountNumber.
 - **Quantity** (*integer*) – The number of active CloudFront key pairs for AwsAccountNumber.
 - **Items** (*list*) – A complex type that lists the active CloudFront key pairs, if any, that are associated with AwsAccountNumber.
 - (*string*) –

- * **DistributionConfig** (*dict*) – The current configuration information for the distribution.
 - **CallerReference** (*string*) – A unique number that ensures the request can't be replayed. If the CallerReference is new (no matter the content of the DistributionConfig object), a new distribution is created. If the CallerReference is a value you already sent in a previous request to create a distribution, and the content of the DistributionConfig is identical to the original request (ignoring white space), the response includes the same information returned to the original request. If the CallerReference is a value you already sent in a previous request to create a distribution but the content of the DistributionConfig is different from the original request, CloudFront returns a DistributionAlreadyExists error.
 - **Aliases** (*dict*) – A complex type that contains information about CNAMEs (alternate domain names), if any, for this distribution.
 - **Quantity** (*integer*) – The number of CNAMEs, if any, for this distribution.
 - **Items** (*list*) – Optional: A complex type that contains CNAME elements, if any, for this distribution. If Quantity is 0, you can omit Items.
 - (*string*) –
 - **DefaultRootObject** (*string*) – The object that you want CloudFront to return (for example, index.html) when an end user requests the root URL for your distribution (<http://www.example.com>) instead of an object in your distribution (<http://www.example.com/index.html>). Specifying a default root object avoids exposing the contents of your distribution. If you don't want to specify a default root object when you create a distribution, include an empty DefaultRootObject element. To delete the default root object from an existing distribution, update the distribution configuration and include an empty DefaultRootObject element. To replace the default root object, update the distribution configuration and specify the new object.
 - **Origins** (*dict*) – A complex type that contains information about origins for this distribution.
 - **Quantity** (*integer*) – The number of origins for this distribution.
 - **Items** (*list*) – A complex type that contains origins for this distribution.
 - (*dict*) – A complex type that describes the Amazon S3 bucket or the HTTP server (for example, a web server) from which CloudFront gets your files. You must create at least one origin.
 - **Id** (*string*) – A unique identifier for the origin. The value of Id must be unique within the distribution. You use the value of Id when you create a cache behavior. The Id identifies the origin that CloudFront routes a request to when the request matches the path pattern for that cache behavior.
 - **DomainName** (*string*) – Amazon S3 origins: The DNS name of the Amazon S3 bucket from which you want CloudFront to get objects for this origin, for example, myaws-bucket.s3.amazonaws.com. Custom origins: The DNS do-

main name for the HTTP server from which you want CloudFront to get objects for this origin, for example, `www.example.com`.

- **OriginPath** (*string*) – An optional element that causes CloudFront to request your content from a directory in your Amazon S3 bucket or your custom origin. When you include the OriginPath element, specify the directory name, beginning with a `/`. CloudFront appends the directory name to the value of DomainName.
- **S3OriginConfig** (*dict*) – A complex type that contains information about the Amazon S3 origin. If the origin is a custom origin, use the CustomOriginConfig element instead.
- **OriginAccessIdentity** (*string*) – The CloudFront origin access identity to associate with the origin. Use an origin access identity to configure the origin so that end users can only access objects in an Amazon S3 bucket through CloudFront. If you want end users to be able to access objects using either the CloudFront URL or the Amazon S3 URL, specify an empty OriginAccessIdentity element. To delete the origin access identity from an existing distribution, update the distribution configuration and include an empty OriginAccessIdentity element. To replace the origin access identity, update the distribution configuration and specify the new origin access identity. Use the format `origin-access-identity/cloudfront/Id` where Id is the value that CloudFront returned in the Id element when you created the origin access identity.
- **CustomOriginConfig** (*dict*) – A complex type that contains information about a custom origin. If the origin is an Amazon S3 bucket, use the S3OriginConfig element instead.
- **HTTPPort** (*integer*) – The HTTP port the custom origin listens on.
- **HTTPSPort** (*integer*) – The HTTPS port the custom origin listens on.
- **OriginProtocolPolicy** (*string*) – The origin protocol policy to apply to your origin.
- **DefaultCacheBehavior** (*dict*) – A complex type that describes the default cache behavior if you do not specify a CacheBehavior element or if files don't match any of the values of PathPattern in CacheBehavior elements. You must create exactly one default cache behavior.
- **TargetOriginId** (*string*) – The value of ID for the origin that you want CloudFront to route requests to when a request matches the path pattern either for a cache behavior or for the default cache behavior.
- **ForwardedValues** (*dict*) – A complex type that specifies how CloudFront handles query strings, cookies and headers.
- **QueryString** (*boolean*) – Indicates whether you want CloudFront to forward query strings to the origin that is associated with this cache behavior. If so, specify `true`; if not, specify `false`.
- **Cookies** (*dict*) – A complex type that specifies how CloudFront handles cookies.
- **Forward** (*string*) – Use this element to specify whether you want CloudFront to forward cookies to the origin that is asso-

ciated with this cache behavior. You can specify all, none or whitelist. If you choose All, CloudFront forwards all cookies regardless of how many your application uses.

- **WhitelistedNames** (*dict*) – A complex type that specifies the whitelisted cookies, if any, that you want CloudFront to forward to your origin that is associated with this cache behavior.
- **Quantity** (*integer*) – The number of whitelisted cookies for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains whitelisted cookies for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **Headers** (*dict*) – A complex type that specifies the Headers, if any, that you want CloudFront to vary upon for this cache behavior.
- **Quantity** (*integer*) – The number of different headers that you want CloudFront to forward to the origin and to vary on for this cache behavior. The maximum number of headers that you can specify by name is 10. If you want CloudFront to forward all headers to the origin and vary on all of them, specify 1 for Quantity and * for Name. If you don't want CloudFront to forward any additional headers to the origin or to vary on any headers, specify 0 for Quantity and omit Items.
- **Items** (*list*) – Optional: A complex type that contains a Name element for each header that you want CloudFront to forward to the origin and to vary on for this cache behavior. If Quantity is 0, omit Items.
- (*string*) –
- **TrustedSigners** (*dict*) – A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the PathPattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don't want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more trusted signers, change Enabled to true (if it's currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.
- **Enabled** (*boolean*) – Specifies whether you want to require end users to use signed URLs to access the files specified by PathPattern and TargetOriginId.
- **Quantity** (*integer*) – The number of trusted signers for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **ViewerProtocolPolicy** (*string*) – Use this element to specify the protocol that users can use to access the files in the origin specified by TargetOriginId when a request matches the path

pattern in PathPattern. If you want CloudFront to allow end users to use any available protocol, specify allow-all. If you want CloudFront to require HTTPS, specify https. If you want CloudFront to respond to an HTTP request with an HTTP status code of 301 (Moved Permanently) and the HTTPS URL, specify redirect-to-https. The viewer then resubmits the request using the HTTPS URL.

- **MinTTL** (*integer*) – The minimum amount of time that you want objects to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **AllowedMethods** (*dict*) – A complex type that controls which HTTP methods CloudFront processes and forwards to your Amazon S3 bucket or your custom origin. There are three choices:
 - CloudFront forwards only GET and HEAD requests.
 - CloudFront forwards only GET, HEAD and OPTIONS requests.
 - CloudFront forwards GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests. If you pick the third choice, you may need to restrict access to your Amazon S3 bucket or to your custom origin so users can't perform operations that you don't want them to. For example, you may not want users to have permission to delete objects from your origin.
- **Quantity** (*integer*) – The number of HTTP methods that you want CloudFront to forward to your origin. Valid values are 2 (for GET and HEAD requests), 3 (for GET, HEAD and OPTIONS requests) and 7 (for GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests).
- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to process and forward to your origin.
- (*string*) –
- **CachedMethods** (*dict*) – A complex type that controls whether CloudFront caches the response to requests using the specified HTTP methods. There are two choices:
 - CloudFront caches responses to GET and HEAD requests.
 - CloudFront caches responses to GET, HEAD, and OPTIONS requests. If you pick the second choice for your S3 Origin, you may need to forward Access-Control-Request-Method, Access-Control-Request-Headers and Origin headers for the responses to be cached correctly.
- **Quantity** (*integer*) – The number of HTTP methods for which you want CloudFront to cache responses. Valid values are 2 (for caching responses to GET and HEAD requests) and 3 (for caching responses to GET, HEAD, and OPTIONS requests).
- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to cache responses to.
- (*string*) –
- **SmoothStreaming** (*boolean*) – Indicates whether you want to distribute media files in Microsoft Smooth Streaming format using the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
- **DefaultTTL** (*integer*) – If you don't configure your origin to

add a Cache-Control max-age directive or an Expires header, DefaultTTL is the default amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin does not add HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).

- **MaxTTL** (*integer*) – The maximum amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin adds HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **CacheBehaviors** (*dict*) – A complex type that contains zero or more CacheBehavior elements.
- **Quantity** (*integer*) – The number of cache behaviors for this distribution.
- **Items** (*list*) – Optional: A complex type that contains cache behaviors for this distribution. If Quantity is 0, you can omit Items.
- (*dict*) – A complex type that describes how CloudFront processes requests. You can create up to 10 cache behaviors. You must create at least as many cache behaviors (including the default cache behavior) as you have origins if you want CloudFront to distribute objects from all of the origins. Each cache behavior specifies the one origin from which you want CloudFront to get objects. If you have two origins and only the default cache behavior, the default cache behavior will cause CloudFront to get objects from one of the origins, but the other origin will never be used. If you don't want to specify any cache behaviors, include only an empty CacheBehaviors element. Don't include an empty CacheBehavior element, or CloudFront returns a MalformedXML error. To delete all cache behaviors in an existing distribution, update the distribution configuration and include only an empty CacheBehaviors element. To add, change, or remove one or more cache behaviors, update the distribution configuration and specify all of the cache behaviors that you want to include in the updated distribution.
- **PathPattern** (*string*) – The pattern (for example, images/* .jpg) that specifies which requests you want this cache behavior to apply to. When CloudFront receives an end-user request, the requested path is compared with path patterns in the order in which cache behaviors are listed in the distribution. The path pattern for the default cache behavior is * and cannot be changed. If the request for an object does not match the path pattern for any cache behaviors, CloudFront applies the behavior in the default cache behavior.
- **TargetOriginId** (*string*) – The value of ID for the origin that you want CloudFront to route requests to when a request

matches the path pattern either for a cache behavior or for the default cache behavior.

- **ForwardedValues** (*dict*) – A complex type that specifies how CloudFront handles query strings, cookies and headers.
- **QueryString** (*boolean*) – Indicates whether you want CloudFront to forward query strings to the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
- **Cookies** (*dict*) – A complex type that specifies how CloudFront handles cookies.
- **Forward** (*string*) – Use this element to specify whether you want CloudFront to forward cookies to the origin that is associated with this cache behavior. You can specify all, none or whitelist. If you choose All, CloudFront forwards all cookies regardless of how many your application uses.
- **WhitelistedNames** (*dict*) – A complex type that specifies the whitelisted cookies, if any, that you want CloudFront to forward to your origin that is associated with this cache behavior.
- **Quantity** (*integer*) – The number of whitelisted cookies for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains whitelisted cookies for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **Headers** (*dict*) – A complex type that specifies the Headers, if any, that you want CloudFront to vary upon for this cache behavior.
- **Quantity** (*integer*) – The number of different headers that you want CloudFront to forward to the origin and to vary on for this cache behavior. The maximum number of headers that you can specify by name is 10. If you want CloudFront to forward all headers to the origin and vary on all of them, specify 1 for Quantity and * for Name. If you don't want CloudFront to forward any additional headers to the origin or to vary on any headers, specify 0 for Quantity and omit Items.
- **Items** (*list*) – Optional: A complex type that contains a Name element for each header that you want CloudFront to forward to the origin and to vary on for this cache behavior. If Quantity is 0, omit Items.
- (*string*) –
- **TrustedSigners** (*dict*) – A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the PathPattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don't want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more trusted signers, change Enabled to true (if it's currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.

- **Enabled** (*boolean*) – Specifies whether you want to require end users to use signed URLs to access the files specified by PathPattern and TargetOriginId.
- **Quantity** (*integer*) – The number of trusted signers for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **ViewerProtocolPolicy** (*string*) – Use this element to specify the protocol that users can use to access the files in the origin specified by TargetOriginId when a request matches the path pattern in PathPattern. If you want CloudFront to allow end users to use any available protocol, specify allow-all. If you want CloudFront to require HTTPS, specify https. If you want CloudFront to respond to an HTTP request with an HTTP status code of 301 (Moved Permanently) and the HTTPS URL, specify redirect-to-https. The viewer then resubmits the request using the HTTPS URL.
- **MinTTL** (*integer*) – The minimum amount of time that you want objects to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **AllowedMethods** (*dict*) – A complex type that controls which HTTP methods CloudFront processes and forwards to your Amazon S3 bucket or your custom origin. There are three choices:
 - CloudFront forwards only GET and HEAD requests.
 - CloudFront forwards only GET, HEAD and OPTIONS requests.
 - CloudFront forwards GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests.
 If you pick the third choice, you may need to restrict access to your Amazon S3 bucket or to your custom origin so users can't perform operations that you don't want them to. For example, you may not want users to have permission to delete objects from your origin.
- **Quantity** (*integer*) – The number of HTTP methods that you want CloudFront to forward to your origin. Valid values are 2 (for GET and HEAD requests), 3 (for GET, HEAD and OPTIONS requests) and 7 (for GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests).
- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to process and forward to your origin.
- (*string*) –
- **CachedMethods** (*dict*) – A complex type that controls whether CloudFront caches the response to requests using the specified HTTP methods. There are two choices:
 - CloudFront caches responses to GET and HEAD requests.
 - CloudFront caches responses to GET, HEAD, and OPTIONS requests.
 If you pick the second choice for your S3 Origin, you may need to forward Access-Control-Request-Method, Access-Control-Request-Headers and Origin headers for the responses to be cached correctly.

- **Quantity** (*integer*) – The number of HTTP methods for which you want CloudFront to cache responses. Valid values are 2 (for caching responses to GET and HEAD requests) and 3 (for caching responses to GET, HEAD, and OPTIONS requests).
- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to cache responses to.
- (*string*) –
- **SmoothStreaming** (*boolean*) – Indicates whether you want to distribute media files in Microsoft Smooth Streaming format using the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
- **DefaultTTL** (*integer*) – If you don't configure your origin to add a Cache-Control max-age directive or an Expires header, DefaultTTL is the default amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin does not add HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **MaxTTL** (*integer*) – The maximum amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin adds HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **CustomErrorResponses** (*dict*) – A complex type that contains zero or more CustomErrorResponse elements.
- **Quantity** (*integer*) – The number of custom error responses for this distribution.
- **Items** (*list*) – Optional: A complex type that contains custom error responses for this distribution. If Quantity is 0, you can omit Items.
- (*dict*) – A complex type that describes how you'd prefer CloudFront to respond to requests that result in either a 4xx or 5xx response. You can control whether a custom error page should be displayed, what the desired response code should be for this error page and how long should the error response be cached by CloudFront. If you don't want to specify any custom error responses, include only an empty CustomErrorResponses element. To delete all custom error responses in an existing distribution, update the distribution configuration and include only an empty CustomErrorResponses element. To add, change, or remove one or more custom error responses, update the distribution configuration and specify all of the custom error responses that you want to include in the updated distribution.
- **ErrorCode** (*integer*) – The 4xx or 5xx HTTP status code that you want to customize. For a list of HTTP status codes that you can customize, see CloudFront documentation.
- **ResponsePagePath** (*string*) – The path of the custom error

page (for example, /custom_404.html). The path is relative to the distribution and must begin with a slash (/). If the path includes any non-ASCII characters or unsafe characters as defined in RFC 1783 (<http://www.ietf.org/rfc/rfc1738.txt>), URL encode those characters. Do not URL encode any other characters in the path, or CloudFront will not return the custom error page to the viewer.

- **ResponseCode** (*string*) – The HTTP status code that you want CloudFront to return with the custom error page to the viewer. For a list of HTTP status codes that you can replace, see CloudFront Documentation.
- **ErrorCachingMinTTL** (*integer*) – The minimum amount of time you want HTTP error codes to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 31,536,000.
- **Comment** (*string*) – Any comments you want to include about the distribution.
- **Logging** (*dict*) – A complex type that controls whether access logs are written for the distribution.
- **Enabled** (*boolean*) – Specifies whether you want CloudFront to save access logs to an Amazon S3 bucket. If you do not want to enable logging when you create a distribution or if you want to disable logging for an existing distribution, specify false for Enabled, and specify empty Bucket and Prefix elements. If you specify false for Enabled but you specify values for Bucket, prefix and IncludeCookies, the values are automatically deleted.
- **IncludeCookies** (*boolean*) – Specifies whether you want CloudFront to include cookies in access logs, specify true for IncludeCookies. If you choose to include cookies in logs, CloudFront logs all cookies regardless of how you configure the cache behaviors for this distribution. If you do not want to include cookies when you create a distribution or if you want to disable include cookies for an existing distribution, specify false for IncludeCookies.
- **Bucket** (*string*) – The Amazon S3 bucket to store the access logs in, for example, myawslogbucket.s3.amazonaws.com.
- **Prefix** (*string*) – An optional string that you want CloudFront to prefix to the access log filenames for this distribution, for example, myprefix/. If you want to enable logging, but you do not want to specify a prefix, you still must include an empty Prefix element in the Logging element.
- **PriceClass** (*string*) – A complex type that contains information about price class for this distribution.
- **Enabled** (*boolean*) – Whether the distribution is enabled to accept end user requests for content.
- **ViewerCertificate** (*dict*) – A complex type that contains information about viewer certificates for this distribution.
- **IAMCertificateId** (*string*) – If you want viewers to use HTTPS to request your objects and you're using an alternate domain name in your object URLs (for example, <https://example.com/logo.jpg>), specify the IAM certificate identifier of the custom viewer certificate for this distribution.

Specify either this value or `CloudFrontDefaultCertificate`.

- **CloudFrontDefaultCertificate** (*boolean*) – If you want viewers to use HTTPS to request your objects and you’re using the CloudFront domain name of your distribution in your object URLs (for example, <https://d1111111abcdef8.cloudfront.net/logo.jpg>), set to `true`. Omit this value if you are setting an `IAMCertificateId`.
- **SSLSupportMethod** (*string*) – If you specify a value for `IAMCertificateId`, you must also specify how you want CloudFront to serve HTTPS requests. Valid values are `vip` and `sni-only`. If you specify `vip`, CloudFront uses dedicated IP addresses for your content and can respond to HTTPS requests from any viewer. However, you must request permission to use this feature, and you incur additional monthly charges. If you specify `sni-only`, CloudFront can only respond to HTTPS requests from viewers that support Server Name Indication (SNI). All modern browsers support SNI, but some browsers still in use don’t support SNI. Do not specify a value for `SSLSupportMethod` if you specified `true` for `CloudFrontDefaultCertificate`.
- **MinimumProtocolVersion** (*string*) – Specify the minimum version of the SSL protocol that you want CloudFront to use, `SSLv3` or `TLSv1`, for HTTPS connections. CloudFront will serve your objects only to browsers or devices that support at least the SSL version that you specify. The `TLSv1` protocol is more secure, so we recommend that you specify `SSLv3` only if your users are using browsers or devices that don’t support `TLSv1`. If you’re using a custom certificate (if you specify a value for `IAMCertificateId`) and if you’re using dedicated IP (if you specify `vip` for `SSLSupportMethod`), you can choose `SSLv3` or `TLSv1` as the `MinimumProtocolVersion`. If you’re using a custom certificate (if you specify a value for `IAMCertificateId`) and if you’re using SNI (if you specify `sni-only` for `SSLSupportMethod`), you must specify `TLSv1` for `MinimumProtocolVersion`.
- **Restrictions** (*dict*) – A complex type that identifies ways in which you want to restrict distribution of your content.
- **GeoRestriction** (*dict*) – A complex type that controls the countries in which your content is distributed. For more information about geo restriction, go to Customizing Error Responses in the Amazon CloudFront Developer Guide. CloudFront determines the location of your users using MaxMind GeoIP databases. For information about the accuracy of these databases, see *How accurate are your GeoIP databases?* on the MaxMind website.
- **RestrictionType** (*string*) – The method that you want to use to restrict distribution of your content by country: - `none`: No geo restriction is enabled, meaning access to content is not restricted by client geo location. - `blacklist`: The `Location` elements specify the countries in which you do not want CloudFront to distribute your content. - `whitelist`: The `Location` elements specify the countries in which you want CloudFront to distribute your content.
- **Quantity** (*integer*) – When geo restriction is enabled, this is

the number of countries in your whitelist or blacklist. Otherwise, when it is not enabled, Quantity is 0, and you can omit Items.

- **Items** (*list*) – A complex type that contains a Location element for each country in which you want CloudFront either to distribute your content (whitelist) or not distribute your content (blacklist). The Location element is a two-letter, uppercase country code for a country that you want to include in your blacklist or whitelist. Include one Location element for each country. CloudFront and MaxMind both use ISO 3166 country codes. For the current list of countries and the corresponding codes, see ISO 3166-1-alpha-2 code on the International Organization for Standardization website. You can also refer to the country list in the CloudFront console, which includes both country names and codes.
- (*string*) –
- **ETag** (*string*) – The current version of the configuration. For example: E2QWRUHAPOMQZL.

update_streaming_distribution (**kwargs)

Update a streaming distribution.

Request Syntax

```
response = client.update_streaming_distribution(
    StreamingDistributionConfig={
        'CallerReference': 'string',
        'S3Origin': {
            'DomainName': 'string',
            'OriginAccessIdentity': 'string'
        },
        'Aliases': {
            'Quantity': 123,
            'Items': [
                'string',
            ]
        },
        'Comment': 'string',
        'Logging': {
            'Enabled': True|False,
            'Bucket': 'string',
            'Prefix': 'string'
        },
        'TrustedSigners': {
            'Enabled': True|False,
            'Quantity': 123,
            'Items': [
                'string',
            ]
        },
        'PriceClass': 'PriceClass_100'|'PriceClass_200'|'PriceClass_All',
        'Enabled': True|False
    },
    Id='string',
    IfMatch='string'
)
```

Parameters

- **StreamingDistributionConfig** (*dict*) – **[REQUIRED]** The streaming distribution's configuration information.
 - **CallerReference** (*string*) – **[REQUIRED]** A unique number that ensures the request can't be replayed. If the CallerReference is new (no matter the content of the StreamingDistributionConfig object), a new streaming distribution is created. If the CallerReference is a value you already sent in a previous request to create a streaming distribution, and the content of the StreamingDistributionConfig is identical to the original request (ignoring white space), the response includes the same information returned to the original request. If the CallerReference is a value you already sent in a previous request to create a streaming distribution but the content of the StreamingDistributionConfig is different from the original request, CloudFront returns a DistributionAlreadyExists error.
 - **S3Origin** (*dict*) – **[REQUIRED]** A complex type that contains information about the Amazon S3 bucket from which you want CloudFront to get your media files for distribution.
 - * **DomainName** (*string*) – **[REQUIRED]** The DNS name of the S3 origin.
 - * **OriginAccessIdentity** (*string*) – **[REQUIRED]** Your S3 origin's origin access identity.
 - **Aliases** (*dict*) – A complex type that contains information about CNAMEs (alternate domain names), if any, for this streaming distribution.
 - * **Quantity** (*integer*) – **[REQUIRED]** The number of CNAMEs, if any, for this distribution.
 - * **Items** (*list*) – Optional: A complex type that contains CNAME elements, if any, for this distribution. If Quantity is 0, you can omit Items.
 - (*string*) –
 - **Comment** (*string*) – **[REQUIRED]** Any comments you want to include about the streaming distribution.
 - **Logging** (*dict*) – A complex type that controls whether access logs are written for the streaming distribution.
 - * **Enabled** (*boolean*) – **[REQUIRED]** Specifies whether you want CloudFront to save access logs to an Amazon S3 bucket. If you do not want to enable logging when you create a streaming distribution or if you want to disable logging for an existing streaming distribution, specify false for Enabled, and specify empty Bucket and Prefix elements. If you specify false for Enabled but you specify values for Bucket and Prefix, the values are automatically deleted.
 - * **Bucket** (*string*) – **[REQUIRED]** The Amazon S3 bucket to store the access logs in, for example, myawslog-bucket.s3.amazonaws.com.
 - * **Prefix** (*string*) – **[REQUIRED]** An optional string that you want CloudFront to prefix to the access log filenames for this streaming distribution, for example, myprefix/. If you want to enable logging, but you do not want to specify a prefix, you still must include an empty Prefix element in the Logging element.
 - **TrustedSigners** (*dict*) – **[REQUIRED]** A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the PathPattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don't

want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more trusted signers, change Enabled to true (if it's currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.

- * **Enabled** (*boolean*) – **[REQUIRED]** Specifies whether you want to require end users to use signed URLs to access the files specified by PathPattern and TargetOriginId.
- * **Quantity** (*integer*) – **[REQUIRED]** The number of trusted signers for this cache behavior.
- * **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit Items.
 - (*string*) –
- **PriceClass** (*string*) – A complex type that contains information about price class for this streaming distribution.
- **Enabled** (*boolean*) – **[REQUIRED]** Whether the streaming distribution is enabled to accept end user requests for content.
- **Id** (*string*) – **[REQUIRED]** The streaming distribution's id.
- **IfMatch** (*string*) – The value of the ETag header you received when retrieving the streaming distribution's configuration. For example: E2QWRUHAPOMQZL.

Return type dict

Returns

Response Syntax

```
{
  'StreamingDistribution': {
    'Id': 'string',
    'Status': 'string',
    'LastModifiedTime': datetime(2015, 1, 1),
    'DomainName': 'string',
    'ActiveTrustedSigners': {
      'Enabled': True|False,
      'Quantity': 123,
      'Items': [
        {
          'AwsAccountNumber': 'string',
          'KeyPairIds': {
            'Quantity': 123,
            'Items': [
              'string',
            ]
          }
        }
      ]
    },
  },
  'StreamingDistributionConfig': {
    'CallerReference': 'string',
    'S3Origin': {
      'DomainName': 'string',
      'OriginAccessIdentity': 'string'
    },
    'Aliases': {
      'Quantity': 123,
      'Items': [
        'string',
      ]
    }
  }
}
```

```
    ],
    },
    'Comment': 'string',
    'Logging': {
        'Enabled': True|False,
        'Bucket': 'string',
        'Prefix': 'string'
    },
    'TrustedSigners': {
        'Enabled': True|False,
        'Quantity': 123,
        'Items': [
            'string',
        ]
    },
    },
    'PriceClass': 'PriceClass_100'|'PriceClass_200'|'PriceClass_All',
    'Enabled': True|False
}
},
'ETag': 'string'
}
```

Response Structure

- (*dict*) – The returned result of the corresponding request.
 - **StreamingDistribution** (*dict*) – The streaming distribution’s information.
 - * **Id** (*string*) – The identifier for the streaming distribution. For example: EGTxBD79H29TRA8.
 - * **Status** (*string*) – The current status of the streaming distribution. When the status is Deployed, the distribution’s information is fully propagated throughout the Amazon CloudFront system.
 - * **LastModifiedTime** (*datetime*) – The date and time the distribution was last modified.
 - * **DomainName** (*string*) – The domain name corresponding to the streaming distribution. For example: s5c39gqb8ow64r.cloudfront.net.
 - * **ActiveTrustedSigners** (*dict*) – CloudFront automatically adds this element to the response only if you’ve set up the distribution to serve private content with signed URLs. The element lists the key pair IDs that CloudFront is aware of for each trusted signer. The Signer child element lists the AWS account number of the trusted signer (or an empty Self element if the signer is you). The Signer element also includes the IDs of any active key pairs associated with the trusted signer’s AWS account. If no KeyPairId element appears for a Signer, that signer can’t create working signed URLs.
 - **Enabled** (*boolean*) – Each active trusted signer.
 - **Quantity** (*integer*) – The number of unique trusted signers included in all cache behaviors. For example, if three cache behaviors all list the same three AWS accounts, the value of Quantity for ActiveTrustedSigners will be 3.
 - **Items** (*list*) – A complex type that contains one Signer complex type for each unique trusted signer that is specified in the TrustedSigners complex type, including trusted signers in the default cache behavior and in all of the other cache behaviors.
 - (*dict*) – A complex type that lists the AWS accounts that were included in the TrustedSigners complex type, as well as their

- active CloudFront key pair IDs, if any.
- **AwsAccountNumber** (*string*) – Specifies an AWS account that can create signed URLs. Values: self, which indicates that the AWS account that was used to create the distribution can create signed URLs, or an AWS account number. Omit the dashes in the account number.
- **KeyPairIds** (*dict*) – A complex type that lists the active CloudFront key pairs, if any, that are associated with AwsAccountNumber.
- **Quantity** (*integer*) – The number of active CloudFront key pairs for AwsAccountNumber.
- **Items** (*list*) – A complex type that lists the active CloudFront key pairs, if any, that are associated with AwsAccountNumber.
- (*string*) –
- * **StreamingDistributionConfig** (*dict*) – The current configuration information for the streaming distribution.
 - **CallerReference** (*string*) – A unique number that ensures the request can't be replayed. If the CallerReference is new (no matter the content of the StreamingDistributionConfig object), a new streaming distribution is created. If the CallerReference is a value you already sent in a previous request to create a streaming distribution, and the content of the StreamingDistributionConfig is identical to the original request (ignoring white space), the response includes the same information returned to the original request. If the CallerReference is a value you already sent in a previous request to create a streaming distribution but the content of the StreamingDistributionConfig is different from the original request, CloudFront returns a DistributionAlreadyExists error.
 - **S3Origin** (*dict*) – A complex type that contains information about the Amazon S3 bucket from which you want CloudFront to get your media files for distribution.
 - **DomainName** (*string*) – The DNS name of the S3 origin.
 - **OriginAccessIdentity** (*string*) – Your S3 origin's origin access identity.
 - **Aliases** (*dict*) – A complex type that contains information about CNAMEs (alternate domain names), if any, for this streaming distribution.
 - **Quantity** (*integer*) – The number of CNAMEs, if any, for this distribution.
 - **Items** (*list*) – Optional: A complex type that contains CNAME elements, if any, for this distribution. If Quantity is 0, you can omit Items.
 - (*string*) –
 - **Comment** (*string*) – Any comments you want to include about the streaming distribution.
 - **Logging** (*dict*) – A complex type that controls whether access logs are written for the streaming distribution.
 - **Enabled** (*boolean*) – Specifies whether you want CloudFront to save access logs to an Amazon S3 bucket. If you do not want to enable logging when you create a streaming distribution or if you want to disable logging for an existing streaming distribution, specify false for Enabled, and specify empty

Bucket and Prefix elements. If you specify false for Enabled but you specify values for Bucket and Prefix, the values are automatically deleted.

- **Bucket** (*string*) – The Amazon S3 bucket to store the access logs in, for example, myawslogbucket.s3.amazonaws.com.
 - **Prefix** (*string*) – An optional string that you want CloudFront to prefix to the access log filenames for this streaming distribution, for example, myprefix/. If you want to enable logging, but you do not want to specify a prefix, you still must include an empty Prefix element in the Logging element.
 - **TrustedSigners** (*dict*) – A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the PathPattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don't want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more trusted signers, change Enabled to true (if it's currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.
 - **Enabled** (*boolean*) – Specifies whether you want to require end users to use signed URLs to access the files specified by PathPattern and TargetOriginId.
 - **Quantity** (*integer*) – The number of trusted signers for this cache behavior.
 - **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit Items.
 - (*string*) –
 - **PriceClass** (*string*) – A complex type that contains information about price class for this streaming distribution.
 - **Enabled** (*boolean*) – Whether the streaming distribution is enabled to accept end user requests for content.
- **ETag** (*string*) – The current version of the configuration. For example: E2QWRUHAPOMQZL.

Paginators

The available paginators are:

- `CloudFront.Paginator.ListCloudFrontOriginAccessIdentities`
- `CloudFront.Paginator.ListDistributions`
- `CloudFront.Paginator.ListInvalidations`
- `CloudFront.Paginator.ListStreamingDistributions`

`class CloudFront.Paginator.ListCloudFrontOriginAccessIdentities`


```
paginator = client.get_paginator('list_cloud_front_origin_access_identities')
```

paginate (**kwargs)

Creates an iterator that will paginate through responses from `CloudFront.Client.list_cloud_front_origin_access_identities()`.

Request Syntax

```
response_iterator = paginator.paginate(
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
    'CloudFrontOriginAccessIdentityList': {
        'Marker': 'string',
        'NextMarker': 'string',
        'MaxItems': 123,
        'IsTruncated': True|False,
        'Quantity': 123,
        'Items': [
            {
                'Id': 'string',
                'S3CanonicalUserId': 'string',
                'Comment': 'string'
            },
        ],
    },
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) – The returned result of the corresponding request.
 - **CloudFrontOriginAccessIdentityList** (*dict*) – The `CloudFrontOriginAccessIdentityList` type.

- * **Marker** (*string*) – The value you provided for the Marker request parameter.
 - * **NextMarker** (*string*) – If IsTruncated is true, this element is present and contains the value you can use for the Marker request parameter to continue listing your origin access identities where they left off.
 - * **MaxItems** (*integer*) – The value you provided for the MaxItems request parameter.
 - * **IsTruncated** (*boolean*) – A flag that indicates whether more origin access identities remain to be listed. If your results were truncated, you can make a follow-up pagination request using the Marker request parameter to retrieve more items in the list.
 - * **Quantity** (*integer*) – The number of CloudFront origin access identities that were created by the current AWS account.
 - * **Items** (*list*) – A complex type that contains one CloudFrontOriginAccessIdentitySummary element for each origin access identity that was created by the current AWS account.
 - (*dict*) – Summary of the information about a CloudFront origin access identity.
 - **Id** (*string*) – The ID for the origin access identity. For example: E74FTE3AJFJ256A.
 - **S3CanonicalUserId** (*string*) – The Amazon S3 canonical user ID for the origin access identity, which you use when giving the origin access identity read permission to an object in Amazon S3.
 - **Comment** (*string*) – The comment for this origin access identity, as originally specified when created.
- **NextToken** (*string*) –
- A token to resume pagination.

class CloudFront.Paginator.**ListDistributions**

```
paginator = client.get_paginator('list_distributions')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `CloudFront.Client.list_distributions()`.

Request Syntax

```
response_iterator = paginator.paginate(
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the NextToken from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'DistributionList': {
    'Marker': 'string',
    'NextMarker': 'string',
    'MaxItems': 123,
    'IsTruncated': True|False,
    'Quantity': 123,
    'Items': [
      {
        'Id': 'string',
        'Status': 'string',
        'LastModifiedTime': datetime(2015, 1, 1),
        'DomainName': 'string',
        'Aliases': {
          'Quantity': 123,
          'Items': [
            'string',
          ]
        },
      },
      {
        'Origins': {
          'Quantity': 123,
          'Items': [
            {
              'Id': 'string',
              'DomainName': 'string',
              'OriginPath': 'string',
              'S3OriginConfig': {
                'OriginAccessIdentity': 'string'
              },
              'CustomOriginConfig': {
                'HTTPPort': 123,
                'HTTPSPort': 123,
                'OriginProtocolPolicy': 'http-only'|'match-viewer'
              }
            },
          ]
        },
      },
    ],
  },
  'DefaultCacheBehavior': {
    'TargetOriginId': 'string',
    'ForwardedValues': {
      'QueryString': True|False,
      'Cookies': {
        'Forward': 'none'|'whitelist'|'all',
        'WhitelistedNames': {
          'Quantity': 123,
          'Items': [
            'string',
          ]
        }
      }
    }
  }
}
```

```
        ]
    },
    'Headers': {
        'Quantity': 123,
        'Items': [
            'string',
        ]
    }
},
'TrustedSigners': {
    'Enabled': True|False,
    'Quantity': 123,
    'Items': [
        'string',
    ]
},
'ViewerProtocolPolicy': 'allow-all'|'https-only'|'redirect-to-https',
'MinTTL': 123,
'AllowedMethods': {
    'Quantity': 123,
    'Items': [
        'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
    ],
    'CachedMethods': {
        'Quantity': 123,
        'Items': [
            'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
        ]
    }
},
'SmoothStreaming': True|False,
'DefaultTTL': 123,
'MaxTTL': 123
},
'CacheBehaviors': {
    'Quantity': 123,
    'Items': [
        {
            'PathPattern': 'string',
            'TargetOriginId': 'string',
            'ForwardedValues': {
                'QueryString': True|False,
                'Cookies': {
                    'Forward': 'none'|'whitelist'|'all',
                    'WhitelistedNames': {
                        'Quantity': 123,
                        'Items': [
                            'string',
                        ]
                    }
                }
            },
            'Headers': {
                'Quantity': 123,
                'Items': [
                    'string',
                ]
            }
        }
    ]
}
```

```

    },
    'TrustedSigners': {
        'Enabled': True|False,
        'Quantity': 123,
        'Items': [
            'string',
        ]
    },
    'ViewerProtocolPolicy': 'allow-all'|'https-only'|'redirect',
    'MinTTL': 123,
    'AllowedMethods': {
        'Quantity': 123,
        'Items': [
            'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
        ],
        'CachedMethods': {
            'Quantity': 123,
            'Items': [
                'GET'|'HEAD'|'POST'|'PUT'|'PATCH'|'OPTIONS'|'DELETE',
            ]
        }
    },
    'SmoothStreaming': True|False,
    'DefaultTTL': 123,
    'MaxTTL': 123
},
]
},
'CustomErrorResponses': {
    'Quantity': 123,
    'Items': [
        {
            'ErrorCode': 123,
            'ResponsePagePath': 'string',
            'ResponseCode': 'string',
            'ErrorCachingMinTTL': 123
        },
    ],
},
],
'Comment': 'string',
'PriceClass': 'PriceClass_100'|'PriceClass_200'|'PriceClass_All',
'Enabled': True|False,
'ViewerCertificate': {
    'IAMCertificateId': 'string',
    'CloudFrontDefaultCertificate': True|False,
    'SSLSupportMethod': 'sni-only'|'vip',
    'MinimumProtocolVersion': 'SSLv3'|'TLSv1'
},
'Restrictions': {
    'GeoRestriction': {
        'RestrictionType': 'blacklist'|'whitelist'|'none',
        'Quantity': 123,
        'Items': [
            'string',
        ]
    },
},
},
},
},

```

```
    ]
    },
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) – The returned result of the corresponding request.
 - **DistributionList** (*dict*) – The DistributionList type.
 - * **Marker** (*string*) – The value you provided for the Marker request parameter.
 - * **NextMarker** (*string*) – If IsTruncated is true, this element is present and contains the value you can use for the Marker request parameter to continue listing your distributions where they left off.
 - * **MaxItems** (*integer*) – The value you provided for the MaxItems request parameter.
 - * **IsTruncated** (*boolean*) – A flag that indicates whether more distributions remain to be listed. If your results were truncated, you can make a follow-up pagination request using the Marker request parameter to retrieve more distributions in the list.
 - * **Quantity** (*integer*) – The number of distributions that were created by the current AWS account.
 - * **Items** (*list*) – A complex type that contains one DistributionSummary element for each distribution that was created by the current AWS account.
 - (*dict*) – A summary of the information for an Amazon CloudFront distribution.
 - **Id** (*string*) – The identifier for the distribution. For example: EDFDVBD632BHDS5.
 - **Status** (*string*) – This response element indicates the current status of the distribution. When the status is Deployed, the distribution's information is fully propagated throughout the Amazon CloudFront system.
 - **LastModifiedTime** (*datetime*) – The date and time the distribution was last modified.
 - **DomainName** (*string*) – The domain name corresponding to the distribution. For example: d604721fxaaqy9.cloudfront.net.
 - **Aliases** (*dict*) – A complex type that contains information about CNAMEs (alternate domain names), if any, for this distribution.
 - **Quantity** (*integer*) – The number of CNAMEs, if any, for this distribution.
 - **Items** (*list*) – Optional: A complex type that contains CNAME elements, if any, for this distribution. If Quantity is 0, you can omit Items.
 - (*string*) –
 - **Origins** (*dict*) – A complex type that contains information about origins for this distribution.
 - **Quantity** (*integer*) – The number of origins for this distribution.
 - **Items** (*list*) – A complex type that contains origins for this distribution.
 - (*dict*) – A complex type that describes the Amazon S3 bucket or the HTTP server (for example, a web server) from which

CloudFront gets your files. You must create at least one origin.

- **Id** (*string*) – A unique identifier for the origin. The value of Id must be unique within the distribution. You use the value of Id when you create a cache behavior. The Id identifies the origin that CloudFront routes a request to when the request matches the path pattern for that cache behavior.
- **DomainName** (*string*) – Amazon S3 origins: The DNS name of the Amazon S3 bucket from which you want CloudFront to get objects for this origin, for example, myaws-bucket.s3.amazonaws.com. Custom origins: The DNS domain name for the HTTP server from which you want CloudFront to get objects for this origin, for example, www.example.com.
- **OriginPath** (*string*) – An optional element that causes CloudFront to request your content from a directory in your Amazon S3 bucket or your custom origin. When you include the OriginPath element, specify the directory name, beginning with a /. CloudFront appends the directory name to the value of DomainName.
- **S3OriginConfig** (*dict*) – A complex type that contains information about the Amazon S3 origin. If the origin is a custom origin, use the CustomOriginConfig element instead.
- **OriginAccessIdentity** (*string*) – The CloudFront origin access identity to associate with the origin. Use an origin access identity to configure the origin so that end users can only access objects in an Amazon S3 bucket through CloudFront. If you want end users to be able to access objects using either the CloudFront URL or the Amazon S3 URL, specify an empty OriginAccessIdentity element. To delete the origin access identity from an existing distribution, update the distribution configuration and include an empty OriginAccessIdentity element. To replace the origin access identity, update the distribution configuration and specify the new origin access identity. Use the format origin-access-identity/cloudfront/Id where Id is the value that CloudFront returned in the Id element when you created the origin access identity.
- **CustomOriginConfig** (*dict*) – A complex type that contains information about a custom origin. If the origin is an Amazon S3 bucket, use the S3OriginConfig element instead.
- **HTTPPort** (*integer*) – The HTTP port the custom origin listens on.
- **HTTPSPort** (*integer*) – The HTTPS port the custom origin listens on.
- **OriginProtocolPolicy** (*string*) – The origin protocol policy to apply to your origin.
- **DefaultCacheBehavior** (*dict*) – A complex type that describes the default cache behavior if you do not specify a CacheBehavior element or if files don't match any of the values of PathPattern in CacheBehavior elements. You must create exactly one default cache behavior.
- **TargetOriginId** (*string*) – The value of ID for the origin that you want CloudFront to route requests to when a request matches the path pattern either for a cache behavior or for the default cache behavior.

- **ForwardedValues** (*dict*) – A complex type that specifies how CloudFront handles query strings, cookies and headers.
- **QueryString** (*boolean*) – Indicates whether you want CloudFront to forward query strings to the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
- **Cookies** (*dict*) – A complex type that specifies how CloudFront handles cookies.
- **Forward** (*string*) – Use this element to specify whether you want CloudFront to forward cookies to the origin that is associated with this cache behavior. You can specify all, none or whitelist. If you choose All, CloudFront forwards all cookies regardless of how many your application uses.
- **WhitelistedNames** (*dict*) – A complex type that specifies the whitelisted cookies, if any, that you want CloudFront to forward to your origin that is associated with this cache behavior.
- **Quantity** (*integer*) – The number of whitelisted cookies for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains whitelisted cookies for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **Headers** (*dict*) – A complex type that specifies the Headers, if any, that you want CloudFront to vary upon for this cache behavior.
- **Quantity** (*integer*) – The number of different headers that you want CloudFront to forward to the origin and to vary on for this cache behavior. The maximum number of headers that you can specify by name is 10. If you want CloudFront to forward all headers to the origin and vary on all of them, specify 1 for Quantity and * for Name. If you don't want CloudFront to forward any additional headers to the origin or to vary on any headers, specify 0 for Quantity and omit Items.
- **Items** (*list*) – Optional: A complex type that contains a Name element for each header that you want CloudFront to forward to the origin and to vary on for this cache behavior. If Quantity is 0, omit Items.
- (*string*) –
- **TrustedSigners** (*dict*) – A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the PathPattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don't want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more trusted signers, change Enabled to true (if it's currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.
- **Enabled** (*boolean*) – Specifies whether you want to require end users to use signed URLs to access the files specified by

PathPattern and TargetOriginId.

- **Quantity** (*integer*) – The number of trusted signers for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **ViewerProtocolPolicy** (*string*) – Use this element to specify the protocol that users can use to access the files in the origin specified by TargetOriginId when a request matches the path pattern in PathPattern. If you want CloudFront to allow end users to use any available protocol, specify allow-all. If you want CloudFront to require HTTPS, specify https. If you want CloudFront to respond to an HTTP request with an HTTP status code of 301 (Moved Permanently) and the HTTPS URL, specify redirect-to-https. The viewer then resubmits the request using the HTTPS URL.
- **MinTTL** (*integer*) – The minimum amount of time that you want objects to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **AllowedMethods** (*dict*) – A complex type that controls which HTTP methods CloudFront processes and forwards to your Amazon S3 bucket or your custom origin. There are three choices:
 - CloudFront forwards only GET and HEAD requests.
 - CloudFront forwards only GET, HEAD and OPTIONS requests.
 - CloudFront forwards GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests.
 If you pick the third choice, you may need to restrict access to your Amazon S3 bucket or to your custom origin so users can't perform operations that you don't want them to. For example, you may not want users to have permission to delete objects from your origin.
- **Quantity** (*integer*) – The number of HTTP methods that you want CloudFront to forward to your origin. Valid values are 2 (for GET and HEAD requests), 3 (for GET, HEAD and OPTIONS requests) and 7 (for GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests).
- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to process and forward to your origin.
- (*string*) –
- **CachedMethods** (*dict*) – A complex type that controls whether CloudFront caches the response to requests using the specified HTTP methods. There are two choices:
 - CloudFront caches responses to GET and HEAD requests.
 - CloudFront caches responses to GET, HEAD, and OPTIONS requests.
 If you pick the second choice for your S3 Origin, you may need to forward Access-Control-Request-Method, Access-Control-Request-Headers and Origin headers for the responses to be cached correctly.
- **Quantity** (*integer*) – The number of HTTP methods for which you want CloudFront to cache responses. Valid values are 2

- (for caching responses to GET and HEAD requests) and 3 (for caching responses to GET, HEAD, and OPTIONS requests).
- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to cache responses to.
 - (*string*) –
 - **SmoothStreaming** (*boolean*) – Indicates whether you want to distribute media files in Microsoft Smooth Streaming format using the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
 - **DefaultTTL** (*integer*) – If you don't configure your origin to add a Cache-Control max-age directive or an Expires header, DefaultTTL is the default amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin does not add HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
 - **MaxTTL** (*integer*) – The maximum amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin adds HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
 - **CacheBehaviors** (*dict*) – A complex type that contains zero or more CacheBehavior elements.
 - **Quantity** (*integer*) – The number of cache behaviors for this distribution.
 - **Items** (*list*) – Optional: A complex type that contains cache behaviors for this distribution. If Quantity is 0, you can omit Items.
 - (*dict*) – A complex type that describes how CloudFront processes requests. You can create up to 10 cache behaviors. You must create at least as many cache behaviors (including the default cache behavior) as you have origins if you want CloudFront to distribute objects from all of the origins. Each cache behavior specifies the one origin from which you want CloudFront to get objects. If you have two origins and only the default cache behavior, the default cache behavior will cause CloudFront to get objects from one of the origins, but the other origin will never be used. If you don't want to specify any cache behaviors, include only an empty CacheBehaviors element. Don't include an empty CacheBehavior element, or CloudFront returns a MalformedXML error. To delete all cache behaviors in an existing distribution, update the distribution configuration and include only an empty CacheBehaviors element. To add, change, or remove one or more cache behaviors, update the distribution configuration and specify all of the cache behaviors that you want to include in the updated distribution.
 - **PathPattern** (*string*) – The pattern (for example, im-

ages/**.jpg*) that specifies which requests you want this cache behavior to apply to. When CloudFront receives an end-user request, the requested path is compared with path patterns in the order in which cache behaviors are listed in the distribution. The path pattern for the default cache behavior is *** and cannot be changed. If the request for an object does not match the path pattern for any cache behaviors, CloudFront applies the behavior in the default cache behavior.

- **TargetOriginId** (*string*) – The value of ID for the origin that you want CloudFront to route requests to when a request matches the path pattern either for a cache behavior or for the default cache behavior.
- **ForwardedValues** (*dict*) – A complex type that specifies how CloudFront handles query strings, cookies and headers.
- **QueryString** (*boolean*) – Indicates whether you want CloudFront to forward query strings to the origin that is associated with this cache behavior. If so, specify *true*; if not, specify *false*.
- **Cookies** (*dict*) – A complex type that specifies how CloudFront handles cookies.
- **Forward** (*string*) – Use this element to specify whether you want CloudFront to forward cookies to the origin that is associated with this cache behavior. You can specify *all*, *none* or *whitelist*. If you choose *All*, CloudFront forwards all cookies regardless of how many your application uses.
- **WhitelistedNames** (*dict*) – A complex type that specifies the whitelisted cookies, if any, that you want CloudFront to forward to your origin that is associated with this cache behavior.
- **Quantity** (*integer*) – The number of whitelisted cookies for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains whitelisted cookies for this cache behavior. If *Quantity* is 0, you can omit *Items*.
- (*string*) –
- **Headers** (*dict*) – A complex type that specifies the Headers, if any, that you want CloudFront to vary upon for this cache behavior.
- **Quantity** (*integer*) – The number of different headers that you want CloudFront to forward to the origin and to vary on for this cache behavior. The maximum number of headers that you can specify by name is 10. If you want CloudFront to forward all headers to the origin and vary on all of them, specify 1 for *Quantity* and *** for *Name*. If you don't want CloudFront to forward any additional headers to the origin or to vary on any headers, specify 0 for *Quantity* and omit *Items*.
- **Items** (*list*) – Optional: A complex type that contains a *Name* element for each header that you want CloudFront to forward to the origin and to vary on for this cache behavior. If *Quantity* is 0, omit *Items*.
- (*string*) –
- **TrustedSigners** (*dict*) – A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the Path-

Pattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don't want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more trusted signers, change Enabled to true (if it's currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.

- **Enabled** (*boolean*) – Specifies whether you want to require end users to use signed URLs to access the files specified by PathPattern and TargetOriginId.
- **Quantity** (*integer*) – The number of trusted signers for this cache behavior.
- **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit Items.
- (*string*) –
- **ViewerProtocolPolicy** (*string*) – Use this element to specify the protocol that users can use to access the files in the origin specified by TargetOriginId when a request matches the path pattern in PathPattern. If you want CloudFront to allow end users to use any available protocol, specify allow-all. If you want CloudFront to require HTTPS, specify https. If you want CloudFront to respond to an HTTP request with an HTTP status code of 301 (Moved Permanently) and the HTTPS URL, specify redirect-to-https. The viewer then resubmits the request using the HTTPS URL.
- **MinTTL** (*integer*) – The minimum amount of time that you want objects to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **AllowedMethods** (*dict*) – A complex type that controls which HTTP methods CloudFront processes and forwards to your Amazon S3 bucket or your custom origin. There are three choices: - CloudFront forwards only GET and HEAD requests. - CloudFront forwards only GET, HEAD and OPTIONS requests. - CloudFront forwards GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests. If you pick the third choice, you may need to restrict access to your Amazon S3 bucket or to your custom origin so users can't perform operations that you don't want them to. For example, you may not want users to have permission to delete objects from your origin.
- **Quantity** (*integer*) – The number of HTTP methods that you want CloudFront to forward to your origin. Valid values are 2 (for GET and HEAD requests), 3 (for GET, HEAD and OPTIONS requests) and 7 (for GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE requests).
- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to process and forward to your origin.

- *(string)* –
- **CachedMethods** (*dict*) – A complex type that controls whether CloudFront caches the response to requests using the specified HTTP methods. There are two choices: - CloudFront caches responses to GET and HEAD requests. - CloudFront caches responses to GET, HEAD, and OPTIONS requests. If you pick the second choice for your S3 Origin, you may need to forward Access-Control-Request-Method, Access-Control-Request-Headers and Origin headers for the responses to be cached correctly.
- **Quantity** (*integer*) – The number of HTTP methods for which you want CloudFront to cache responses. Valid values are 2 (for caching responses to GET and HEAD requests) and 3 (for caching responses to GET, HEAD, and OPTIONS requests).
- **Items** (*list*) – A complex type that contains the HTTP methods that you want CloudFront to cache responses to.
- *(string)* –
- **SmoothStreaming** (*boolean*) – Indicates whether you want to distribute media files in Microsoft Smooth Streaming format using the origin that is associated with this cache behavior. If so, specify true; if not, specify false.
- **DefaultTTL** (*integer*) – If you don't configure your origin to add a Cache-Control max-age directive or an Expires header, DefaultTTL is the default amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin does not add HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **MaxTTL** (*integer*) – The maximum amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. The value that you specify applies only when your origin adds HTTP headers such as Cache-Control max-age, Cache-Control s-maxage, and Expires to objects. You can specify a value from 0 to 3,153,600,000 seconds (100 years).
- **CustomErrorResponses** (*dict*) – A complex type that contains zero or more CustomErrorResponses elements.
- **Quantity** (*integer*) – The number of custom error responses for this distribution.
- **Items** (*list*) – Optional: A complex type that contains custom error responses for this distribution. If Quantity is 0, you can omit Items.
- (*dict*) – A complex type that describes how you'd prefer CloudFront to respond to requests that result in either a 4xx or 5xx response. You can control whether a custom error page should be displayed, what the desired response code should be for this error page and how long should the error response be cached by CloudFront. If you don't want to specify any custom error responses, include only an empty CustomErrorResponses element. To delete all custom error responses

in an existing distribution, update the distribution configuration and include only an empty `CustomErrorResponses` element. To add, change, or remove one or more custom error responses, update the distribution configuration and specify all of the custom error responses that you want to include in the updated distribution.

- **ErrorCode** (*integer*) – The 4xx or 5xx HTTP status code that you want to customize. For a list of HTTP status codes that you can customize, see CloudFront documentation.
- **ResponsePagePath** (*string*) – The path of the custom error page (for example, `/custom_404.html`). The path is relative to the distribution and must begin with a slash (/). If the path includes any non-ASCII characters or unsafe characters as defined in RFC 1783 (<http://www.ietf.org/rfc/rfc1738.txt>), URL encode those characters. Do not URL encode any other characters in the path, or CloudFront will not return the custom error page to the viewer.
- **ResponseCode** (*string*) – The HTTP status code that you want CloudFront to return with the custom error page to the viewer. For a list of HTTP status codes that you can replace, see CloudFront Documentation.
- **ErrorCachingMinTTL** (*integer*) – The minimum amount of time you want HTTP error codes to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. You can specify a value from 0 to 31,536,000.
- **Comment** (*string*) – The comment originally specified when this distribution was created.
- **PriceClass** (*string*) –
- **Enabled** (*boolean*) – Whether the distribution is enabled to accept end user requests for content.
- **ViewerCertificate** (*dict*) – A complex type that contains information about viewer certificates for this distribution.
- **IAMCertificateId** (*string*) – If you want viewers to use HTTPS to request your objects and you're using an alternate domain name in your object URLs (for example, <https://example.com/logo.jpg>), specify the IAM certificate identifier of the custom viewer certificate for this distribution. Specify either this value or `CloudFrontDefaultCertificate`.
- **CloudFrontDefaultCertificate** (*boolean*) – If you want viewers to use HTTPS to request your objects and you're using the CloudFront domain name of your distribution in your object URLs (for example, <https://d1111111abcdef8.cloudfront.net/logo.jpg>), set to `true`. Omit this value if you are setting an `IAMCertificateId`.
- **SSLSupportMethod** (*string*) – If you specify a value for `IAMCertificateId`, you must also specify how you want CloudFront to serve HTTPS requests. Valid values are `vip` and `sni-only`. If you specify `vip`, CloudFront uses dedicated IP addresses for your content and can respond to HTTPS requests from any viewer. However, you must request permission to use this feature, and you incur additional monthly charges. If you specify `sni-only`, CloudFront can only respond to HTTPS requests from viewers that support Server Name Indication

(SNI). All modern browsers support SNI, but some browsers still in use don't support SNI. Do not specify a value for `SSLSupportMethod` if you specified `true` for `CloudFrontDefaultCertificate`.

- **MinimumProtocolVersion** (*string*) – Specify the minimum version of the SSL protocol that you want CloudFront to use, `SSLv3` or `TLSv1`, for HTTPS connections. CloudFront will serve your objects only to browsers or devices that support at least the SSL version that you specify. The `TLSv1` protocol is more secure, so we recommend that you specify `SSLv3` only if your users are using browsers or devices that don't support `TLSv1`. If you're using a custom certificate (if you specify a value for `IAMCertificateId`) and if you're using dedicated IP (if you specify `vip` for `SSLSupportMethod`), you can choose `SSLv3` or `TLSv1` as the `MinimumProtocolVersion`. If you're using a custom certificate (if you specify a value for `IAMCertificateId`) and if you're using SNI (if you specify `sni-only` for `SSLSupportMethod`), you must specify `TLSv1` for `MinimumProtocolVersion`.
- **Restrictions** (*dict*) – A complex type that identifies ways in which you want to restrict distribution of your content.
- **GeoRestriction** (*dict*) – A complex type that controls the countries in which your content is distributed. For more information about geo restriction, go to Customizing Error Responses in the Amazon CloudFront Developer Guide. CloudFront determines the location of your users using MaxMind GeoIP databases. For information about the accuracy of these databases, see *How accurate are your GeoIP databases?* on the MaxMind website.
- **RestrictionType** (*string*) – The method that you want to use to restrict distribution of your content by country: - `none`: No geo restriction is enabled, meaning access to content is not restricted by client geo location. - `blacklist`: The `Location` elements specify the countries in which you do not want CloudFront to distribute your content. - `whitelist`: The `Location` elements specify the countries in which you want CloudFront to distribute your content.
- **Quantity** (*integer*) – When geo restriction is enabled, this is the number of countries in your whitelist or blacklist. Otherwise, when it is not enabled, `Quantity` is 0, and you can omit `Items`.
- **Items** (*list*) – A complex type that contains a `Location` element for each country in which you want CloudFront either to distribute your content (whitelist) or not distribute your content (blacklist). The `Location` element is a two-letter, upper-case country code for a country that you want to include in your blacklist or whitelist. Include one `Location` element for each country. CloudFront and MaxMind both use ISO 3166 country codes. For the current list of countries and the corresponding codes, see ISO 3166-1-alpha-2 code on the International Organization for Standardization website. You can also refer to the country list in the CloudFront console, which includes both country names and codes.
- (*string*) –

– **NextToken** (*string*) –

A token to resume pagination.

class CloudFront.Paginator.**ListInvalidations**

```
paginator = client.get_paginator('list_invalidations')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `CloudFront.Client.list_invalidations()`.

Request Syntax

```
response_iterator = paginator.paginate(
    DistributionId='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **DistributionId** (*string*) – **[REQUIRED]** The distribution’s id.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
    'InvalidationList': {
        'Marker': 'string',
        'NextMarker': 'string',
        'MaxItems': 123,
        'IsTruncated': True|False,
        'Quantity': 123,
        'Items': [
            {
                'Id': 'string',
                'CreateTime': datetime(2015, 1, 1),
                'Status': 'string'
            },
        ]
    }
}
```



```

    },
    'NextToken': 'string'
}

```

Response Structure

- (*dict*) – The returned result of the corresponding request.
 - **InvalidationList** (*dict*) – Information about invalidation batches.
 - * **Marker** (*string*) – The value you provided for the Marker request parameter.
 - * **NextMarker** (*string*) – If IsTruncated is true, this element is present and contains the value you can use for the Marker request parameter to continue listing your invalidation batches where they left off.
 - * **MaxItems** (*integer*) – The value you provided for the MaxItems request parameter.
 - * **IsTruncated** (*boolean*) – A flag that indicates whether more invalidation batch requests remain to be listed. If your results were truncated, you can make a follow-up pagination request using the Marker request parameter to retrieve more invalidation batches in the list.
 - * **Quantity** (*integer*) – The number of invalidation batches that were created by the current AWS account.
 - * **Items** (*list*) – A complex type that contains one InvalidationSummary element for each invalidation batch that was created by the current AWS account.
 - (*dict*) – Summary of an invalidation request.
 - **Id** (*string*) – The unique ID for an invalidation request.
 - **CreateTime** (*datetime*) –
 - **Status** (*string*) – The status of an invalidation request.
 - **NextToken** (*string*) –

A token to resume pagination.

class CloudFront.Paginator.**ListStreamingDistributions**

```
paginator = client.get_paginator('list_streaming_distributions')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `CloudFront.Client.list_streaming_distributions()`.

Request Syntax

```

response_iterator = paginator.paginate(
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)

```

Parameters **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided

in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the NextToken from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'StreamingDistributionList': {
    'Marker': 'string',
    'NextMarker': 'string',
    'MaxItems': 123,
    'IsTruncated': True|False,
    'Quantity': 123,
    'Items': [
      {
        'Id': 'string',
        'Status': 'string',
        'LastModifiedTime': datetime(2015, 1, 1),
        'DomainName': 'string',
        'S3Origin': {
          'DomainName': 'string',
          'OriginAccessIdentity': 'string'
        },
        'Aliases': {
          'Quantity': 123,
          'Items': [
            'string',
          ]
        },
        'TrustedSigners': {
          'Enabled': True|False,
          'Quantity': 123,
          'Items': [
            'string',
          ]
        },
        'Comment': 'string',
        'PriceClass': 'PriceClass_100'|'PriceClass_200'|'PriceClass_All',
        'Enabled': True|False
      },
    ],
  },
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) – The returned result of the corresponding request.
 - **StreamingDistributionList** (*dict*) – The StreamingDistributionList type.
 - * **Marker** (*string*) – The value you provided for the Marker request parameter.
 - * **NextMarker** (*string*) – If IsTruncated is true, this element is present and contains the value you can use for the Marker request parameter

to continue listing your streaming distributions where they left off.

- * **MaxItems** (*integer*) – The value you provided for the MaxItems request parameter.
- * **IsTruncated** (*boolean*) – A flag that indicates whether more streaming distributions remain to be listed. If your results were truncated, you can make a follow-up pagination request using the Marker request parameter to retrieve more distributions in the list.
- * **Quantity** (*integer*) – The number of streaming distributions that were created by the current AWS account.
- * **Items** (*list*) – A complex type that contains one StreamingDistributionSummary element for each distribution that was created by the current AWS account.
 - (*dict*) – A summary of the information for an Amazon CloudFront streaming distribution.
 - **Id** (*string*) – The identifier for the distribution. For example: EDFDVBD632BHDS5.
 - **Status** (*string*) – Indicates the current status of the distribution. When the status is Deployed, the distribution’s information is fully propagated throughout the Amazon CloudFront system.
 - **LastModifiedTime** (*datetime*) – The date and time the distribution was last modified.
 - **DomainName** (*string*) – The domain name corresponding to the distribution. For example: d604721fxaaqy9.cloudfront.net.
 - **S3Origin** (*dict*) – A complex type that contains information about the Amazon S3 bucket from which you want CloudFront to get your media files for distribution.
 - **DomainName** (*string*) – The DNS name of the S3 origin.
 - **OriginAccessIdentity** (*string*) – Your S3 origin’s origin access identity.
 - **Aliases** (*dict*) – A complex type that contains information about CNAMEs (alternate domain names), if any, for this streaming distribution.
 - **Quantity** (*integer*) – The number of CNAMEs, if any, for this distribution.
 - **Items** (*list*) – Optional: A complex type that contains CNAME elements, if any, for this distribution. If Quantity is 0, you can omit Items.
 - (*string*) –
 - **TrustedSigners** (*dict*) – A complex type that specifies the AWS accounts, if any, that you want to allow to create signed URLs for private content. If you want to require signed URLs in requests for objects in the target origin that match the PathPattern for this cache behavior, specify true for Enabled, and specify the applicable values for Quantity and Items. For more information, go to Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide. If you don’t want to require signed URLs in requests for objects that match PathPattern, specify false for Enabled and 0 for Quantity. Omit Items. To add, change, or remove one or more trusted signers, change Enabled to true (if it’s currently false), change Quantity as applicable, and specify all of the trusted signers that you want to include in the updated distribution.

- **Enabled** (*boolean*) – Specifies whether you want to require end users to use signed URLs to access the files specified by PathPattern and TargetOriginId.
 - **Quantity** (*integer*) – The number of trusted signers for this cache behavior.
 - **Items** (*list*) – Optional: A complex type that contains trusted signers for this cache behavior. If Quantity is 0, you can omit Items.
 - (*string*) –
 - **Comment** (*string*) – The comment originally specified when this distribution was created.
 - **PriceClass** (*string*) –
 - **Enabled** (*boolean*) – Whether the distribution is enabled to accept end user requests for content.
- **NextToken** (*string*) –
- A token to resume pagination.

Waiters

The available waiters are:

- `CloudFront.Waiter.DistributionDeployed`
- `CloudFront.Waiter.InvalidatioCompleted`
- `CloudFront.Waiter.StreamingDistributionDeployed`

class `CloudFront.Waiter.DistributionDeployed`

```
waiter = client.get_waiter('distribution_deployed')
```

wait (***kwargs*)

Polls `CloudFront.Client.get_distribution()` every 60 seconds until a successful state is reached. An error is returned after 25 failed checks.

Request Syntax

```
waiter.wait(  
    Id='string'  
)
```

Parameters *Id* (*string*) – [REQUIRED] The distribution’s id.

Returns None

class `CloudFront.Waiter.InvalidatioCompleted`

```
waiter = client.get_waiter('invalidation_completed')
```

wait (***kwargs*)

Polls `CloudFront.Client.get_invalidation()` every 20 seconds until a successful state is reached. An error is returned after 60 failed checks.

Request Syntax

```
waiter.wait(
    DistributionId='string',
    Id='string'
)
```

Parameters

- **DistributionId** (*string*) – [REQUIRED] The distribution’s id.
- **Id** (*string*) – [REQUIRED] The invalidation’s id.

Returns None

class CloudFront.Waiter.**StreamingDistributionDeployed**

```
waiter = client.get_waiter('streaming_distribution_deployed')
```

wait (***kwargs*)

Polls `CloudFront.Client.get_streaming_distribution()` every 60 seconds until a successful state is reached. An error is returned after 25 failed checks.

Request Syntax

```
waiter.wait(
    Id='string'
)
```

Parameters **Id** (*string*) – [REQUIRED] The streaming distribution’s id.

Returns None**CloudHSM****Table of Contents**

- CloudHSM
 - Client

Client

class CloudHSM.**Client**

A low-level client representing Amazon CloudHSM:

```
import boto3
```

```
client = boto3.client('cloudhsm')
```

These are the available methods:

- `can_paginate()`
- `create_hapg()`
- `create_hsm()`
- `create_luna_client()`
- `delete_hapg()`
- `delete_hsm()`
- `delete_luna_client()`

- `describe_hapg()`
- `describe_hsm()`
- `describe_luna_client()`
- `generate_presigned_url()`
- `get_config()`
- `get_paginator()`
- `get_waiter()`
- `list_available_zones()`
- `list_hapgs()`
- `list_hsms()`
- `list_luna_clients()`
- `modify_hapg()`
- `modify_hsm()`
- `modify_luna_client()`

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

create_hapg (***kwargs*)

Creates a high-availability partition group. A high-availability partition group is a group of partitions that spans multiple physical HSMs.

Request Syntax

```
response = client.create_hapg(
    Label='string'
)
```

Parameters *Label* (*string*) – [REQUIRED]

The label of the new high-availability partition group.

Return type dict

Returns

Response Syntax

```
{
    'HapgArn': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output of the CreateHAPartitionGroup action.

- **HapgArn** (*string*) –

The ARN of the high-availability partition group.

create_hsm (***kwargs*)

Creates an uninitialized HSM instance. Running this command provisions an HSM appliance and will result in charges to your AWS account for the HSM.

Request Syntax

```
response = client.create_hsm(
    SubnetId='string',
    SshKey='string',
    EniIp='string',
    IamRoleArn='string',
    ExternalId='string',
    SubscriptionType='PRODUCTION',
    ClientToken='string',
    SyslogIp='string'
)
```

Parameters

- **SubnetId** (*string*) – [REQUIRED]
The identifier of the subnet in your VPC in which to place the HSM.
- **SshKey** (*string*) – [REQUIRED]
The SSH public key to install on the HSM.
- **EniIp** (*string*) – The IP address to assign to the HSM’s ENI.
- **IamRoleArn** (*string*) – [REQUIRED]
The ARN of an IAM role to enable the AWS CloudHSM service to allocate an ENI on your behalf.
- **ExternalId** (*string*) – The external ID from **IamRoleArn** , if present.
- **SubscriptionType** (*string*) – [REQUIRED]
The subscription type.
- **ClientToken** (*string*) – A user-defined token to ensure idempotence. Subsequent calls to this action with the same token will be ignored.
- **SyslogIp** (*string*) – The IP address for the syslog monitoring server.

Return type dict

Returns

Response Syntax

```
{
    'HsmArn': 'string'
}
```

Response Structure

- (*dict*) –
Contains the output of the CreateHsm action.
 - **HsmArn** (*string*) –
The ARN of the HSM.

create_luna_client (***kwargs*)
Creates an HSM client.

Request Syntax

```
response = client.create_luna_client(
    Label='string',
    Certificate='string'
)
```

Parameters

- **Label** (*string*) – The label for the client.

- **Certificate** (*string*) – [REQUIRED]

The contents of a Base64-Encoded X.509 v3 certificate to be installed on the HSMs used by this client.

Return type dict

Returns

Response Syntax

```
{
    'ClientArn': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output of the CreateLunaClient action.

- **ClientArn** (*string*) –

The ARN of the client.

delete_hapg (***kwargs*)

Deletes a high-availability partition group.

Request Syntax

```
response = client.delete_hapg(
    HapgArn='string'
)
```

Parameters **HapgArn** (*string*) – [REQUIRED]

The ARN of the high-availability partition group to delete.

Return type dict

Returns

Response Syntax

```
{
    'Status': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output of the DeleteHapg action.

- **Status** (*string*) –

The status of the action.

delete_hsm (***kwargs*)

Deletes an HSM. Once complete, this operation cannot be undone and your key material cannot be recovered.

Request Syntax

```
response = client.delete_hsm(
    HsmArn='string'
)
```


Parameters `HsmArn` (*string*) – [REQUIRED]

The ARN of the HSM to delete.

Return type dict

Returns**Response Syntax**

```
{
    'Status': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output of the DeleteHsm action.

- **Status** (*string*) –

The status of the action.

delete_luna_client (***kwargs*)

Deletes a client.

Request Syntax

```
response = client.delete_luna_client(
    ClientArn='string'
)
```

Parameters `ClientArn` (*string*) – [REQUIRED]

The ARN of the client to delete.

Return type dict

Returns**Response Syntax**

```
{
    'Status': 'string'
}
```

Response Structure

- (*dict*) –

- **Status** (*string*) –

The status of the action.

describe_hapg (***kwargs*)

Retrieves information about a high-availability partition group.

Request Syntax

```
response = client.describe_hapg(
    HapgArn='string'
)
```

Parameters `HapgArn` (*string*) – [REQUIRED]

The ARN of the high-availability partition group to describe.

Return type dict

Returns

Response Syntax

```
{
  'HapgArn': 'string',
  'HapgSerial': 'string',
  'HsmsLastActionFailed': [
    'string',
  ],
  'HsmsPendingDeletion': [
    'string',
  ],
  'HsmsPendingRegistration': [
    'string',
  ],
  'Label': 'string',
  'LastModifiedTimestamp': 'string',
  'PartitionSerialList': [
    'string',
  ],
  'State': 'READY' | 'UPDATING' | 'DEGRADED'
}
```

Response Structure

- (*dict*) –

Contains the output of the DescribeHapg action.

- **HapgArn** (*string*) –

The ARN of the high-availability partition group.

- **HapgSerial** (*string*) –

The serial number of the high-availability partition group.

- **HsmsLastActionFailed** (*list*) –

Contains a list of ARNs that identify the HSMs.

* (*string*) –

An ARN that identifies an HSM.

- **HsmsPendingDeletion** (*list*) –

Contains a list of ARNs that identify the HSMs.

* (*string*) –

An ARN that identifies an HSM.

- **HsmsPendingRegistration** (*list*) –

Contains a list of ARNs that identify the HSMs.

* (*string*) –

An ARN that identifies an HSM.

- **Label** (*string*) –

The label for the high-availability partition group.

- **LastModifiedTimestamp** (*string*) –

The date and time the high-availability partition group was last modified.

- **PartitionSerialList** (*list*) –

The list of partition serial numbers that belong to the high-availability partition group.

- * (*string*) –
- **State** (*string*) –

The state of the high-availability partition group.

describe_hsm (**kwargs)

Retrieves information about an HSM. You can identify the HSM by its ARN or its serial number.

Request Syntax

```
response = client.describe_hsm(
    HsmArn='string',
    HsmSerialNumber='string'
)
```

Parameters

- **HsmArn** (*string*) – The ARN of the HSM. Either the *HsmArn* or the *SerialNumber* parameter must be specified.
- **HsmSerialNumber** (*string*) – The serial number of the HSM. Either the *HsmArn* or the *HsmSerialNumber* parameter must be specified.

Return type dict

Returns

Response Syntax

```
{
    'HsmArn': 'string',
    'Status': 'PENDING' | 'RUNNING' | 'UPDATING' | 'SUSPENDED' | 'TERMINATING' | 'TERMINATED',
    'StatusDetails': 'string',
    'AvailabilityZone': 'string',
    'EniId': 'string',
    'EniIp': 'string',
    'SubscriptionType': 'PRODUCTION',
    'SubscriptionStartDate': 'string',
    'SubscriptionEndDate': 'string',
    'VpcId': 'string',
    'SubnetId': 'string',
    'IamRoleArn': 'string',
    'SerialNumber': 'string',
    'VendorName': 'string',
    'HsmType': 'string',
    'SoftwareVersion': 'string',
    'SshPublicKey': 'string',
    'SshKeyLastUpdated': 'string',
    'ServerCertUri': 'string',
    'ServerCertLastUpdated': 'string',
    'Partitions': [
        'string',
    ]
}
```

Response Structure

- (*dict*) –
 - Contains the output of the DescribeHsm action.
 - **HsmArn** (*string*) –
 - The ARN of the HSM.

- **Status** (*string*) –
The status of the HSM.
- **StatusDetails** (*string*) –
Contains additional information about the status of the HSM.
- **AvailabilityZone** (*string*) –
The Availability Zone that the HSM is in.
- **EniId** (*string*) –
The identifier of the elastic network interface (ENI) attached to the HSM.
- **EniIp** (*string*) –
The IP address assigned to the HSM’s ENI.
- **SubscriptionType** (*string*) –
The subscription type.
- **SubscriptionStartDate** (*string*) –
The subscription start date.
- **SubscriptionEndDate** (*string*) –
The subscription end date.
- **VpcId** (*string*) –
The identifier of the VPC that the HSM is in.
- **SubnetId** (*string*) –
The identifier of the subnet the HSM is in.
- **IamRoleArn** (*string*) –
The ARN of the IAM role assigned to the HSM.
- **SerialNumber** (*string*) –
The serial number of the HSM.
- **VendorName** (*string*) –
The name of the HSM vendor.
- **HsmType** (*string*) –
The HSM model type.
- **SoftwareVersion** (*string*) –
The HSM software version.
- **SshPublicKey** (*string*) –
The public SSH key.
- **SshKeyLastUpdated** (*string*) –
The date and time the SSH key was last updated.
- **ServerCertUri** (*string*) –
The URI of the certificate server.
- **ServerCertLastUpdated** (*string*) –
The date and time the server certificate was last updated.
- **Partitions** (*list*) –
The list of partitions on the HSM.
 - * (*string*) –

describe_luna_client (***kwargs*)
Retrieves information about an HSM client.

Request Syntax

```
response = client.describe_luna_client(
    ClientArn='string',
    CertificateFingerprint='string'
)
```

Parameters

- **ClientArn** (*string*) – The ARN of the client.
- **CertificateFingerprint** (*string*) – The certificate fingerprint.

Return type dict

Returns

Response Syntax

```
{
    'ClientArn': 'string',
    'Certificate': 'string',
    'CertificateFingerprint': 'string',
    'LastModifiedTimestamp': 'string',
    'Label': 'string'
}
```

Response Structure

- (*dict*) –
 - **ClientArn** (*string*) –
The ARN of the client.
 - **Certificate** (*string*) –
The certificate installed on the HSMs used by this client.
 - **CertificateFingerprint** (*string*) –
The certificate fingerprint.
 - **LastModifiedTimestamp** (*string*) –
The date and time the client was last modified.
 - **Label** (*string*) –
The label of the client.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_config (***kwargs*)

Gets the configuration files necessary to connect to all high availability partition groups the client is associated with.

Request Syntax

```
response = client.get_config(  
    ClientArn='string',  
    ClientVersion='5.1'|'5.3',  
    HapgList=[  
        'string',  
    ]  
)
```

Parameters

- **ClientArn** (*string*) – [REQUIRED]

The ARN of the client.

- **ClientVersion** (*string*) – [REQUIRED]

The client version.

- **HapgList** (*list*) – [REQUIRED]

A list of ARNs that identify the high-availability partition groups that are associated with the client.

– (*string*) –

Return type dict

Returns**Response Syntax**

```
{  
    'ConfigType': 'string',  
    'ConfigFile': 'string',  
    'ConfigCred': 'string'  
}
```

Response Structure

- (*dict*) –
 - **ConfigType** (*string*) –
The type of credentials.
 - **ConfigFile** (*string*) –
The chrystoki.conf configuration file.
 - **ConfigCred** (*string*) –
The certificate file containing the server.pem files of the HSMs.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type L{botocore.paginate.Paginator}

Returns A paginator object.

get_waiter (*waiter_name*)

list_available_zones ()

Lists the Availability Zones that have available AWS CloudHSM capacity.

Request Syntax

```
response = client.list_available_zones()
```

Return type dict**Returns****Response Syntax**

```
{
    'AZList': [
        'string',
    ]
}
```

Response Structure

- (*dict*) –
 - **AZList** (*list*) –

The list of Availability Zones that have available AWS CloudHSM capacity.

* (*string*) –

list_hapgs (***kwargs*)

Lists the high-availability partition groups for the account.

This operation supports pagination with the use of the *NextToken* member. If more results are available, the *NextToken* member of the response contains a token that you pass in the next call to ListHapgs to retrieve the next set of items.

Request Syntax

```
response = client.list_hapgs(
    NextToken='string'
)
```

Parameters **NextToken** (*string*) – The *NextToken* value from a previous call to ListHapgs .
Pass null if this is the first call.

Return type dict**Returns****Response Syntax**

```
{
    'HapgList': [
        'string',
    ],
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **HapgList** (*list*) –

The list of high-availability partition groups.

* (*string*) –

– **NextToken** (*string*) –

If not null, more results are available. Pass this value to ListHaps to retrieve the next set of items.

list_hsms (***kwargs*)

Retrieves the identifiers of all of the HSMs provisioned for the current customer.

This operation supports pagination with the use of the *NextToken* member. If more results are available, the *NextToken* member of the response contains a token that you pass in the next call to ListHsms to retrieve the next set of items.

Request Syntax

```
response = client.list_hsms(
    NextToken='string'
)
```

Parameters **NextToken** (*string*) – The *NextToken* value from a previous call to ListHsms .

Pass null if this is the first call.

Return type dict

Returns

Response Syntax

```
{
    'HsmList': [
        'string',
    ],
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output of the ListHsms action.

– **HsmList** (*list*) –

The list of ARNs that identify the HSMs.

* (*string*) –

An ARN that identifies an HSM.

– **NextToken** (*string*) –

If not null, more results are available. Pass this value to ListHsms to retrieve the next set of items.

list_luna_clients (***kwargs*)

Lists all of the clients.

This operation supports pagination with the use of the *NextToken* member. If more results are available, the *NextToken* member of the response contains a token that you pass in the next call to ListLunaClients to retrieve the next set of items.

Request Syntax

```
response = client.list_luna_clients(
    NextToken='string'
)
```


Parameters **NextToken** (*string*) – The *NextToken* value from a previous call to `ListLunaClients`. Pass null if this is the first call.

Return type dict

Returns

Response Syntax

```
{
    'ClientList': [
        'string',
    ],
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **ClientList** (*list*) –

The list of clients.

 - * (*string*) –
 - **NextToken** (*string*) –

If not null, more results are available. Pass this to `ListLunaClients` to retrieve the next set of items.

modify_hapg (**kwargs)

Modifies an existing high-availability partition group.

Request Syntax

```
response = client.modify_hapg(
    HapgArn='string',
    Label='string',
    PartitionSerialList=[
        'string',
    ]
)
```

Parameters

- **HapgArn** (*string*) – [REQUIRED]

The ARN of the high-availability partition group to modify.
- **Label** (*string*) – The new label for the high-availability partition group.
- **PartitionSerialList** (*list*) – The list of partition serial numbers to make members of the high-availability partition group.
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'HapgArn': 'string'
}
```

Response Structure

- (*dict*) –

– **HapgArn** (*string*) –

The ARN of the high-availability partition group.

modify_hsm (***kwargs*)

Modifies an HSM.

Request Syntax

```
response = client.modify_hsm(  
    HsmArn='string',  
    SubnetId='string',  
    EniIp='string',  
    IamRoleArn='string',  
    ExternalId='string',  
    SyslogIp='string'  
)
```

Parameters

- **HsmArn** (*string*) – **[REQUIRED]**

The ARN of the HSM to modify.

- **SubnetId** (*string*) – The new identifier of the subnet that the HSM is in.
- **EniIp** (*string*) – The new IP address for the elastic network interface attached to the HSM.
- **IamRoleArn** (*string*) – The new IAM role ARN.
- **ExternalId** (*string*) – The new external ID.
- **SyslogIp** (*string*) – The new IP address for the syslog monitoring server.

Return type dict

Returns

Response Syntax

```
{  
    'HsmArn': 'string'  
}
```

Response Structure

- (*dict*) –

Contains the output of the ModifyHsm action.

– **HsmArn** (*string*) –

The ARN of the HSM.

modify_luna_client (***kwargs*)

Modifies the certificate used by the client.

This action can potentially start a workflow to install the new certificate on the client’s HSMs.

Request Syntax

```
response = client.modify_luna_client(  
    ClientArn='string',  
    Certificate='string'  
)
```

Parameters

- **ClientArn** (*string*) – **[REQUIRED]**

The ARN of the client.

- **Certificate** (*string*) – [REQUIRED]

The new certificate for the client.

Return type dict

Returns

Response Syntax

```
{
    'ClientArn': 'string'
}
```

Response Structure

- (*dict*) –
 - **ClientArn** (*string*) –

The ARN of the client.

CloudSearch

Table of Contents

- CloudSearch
 - Client

Client

class CloudSearch.**Client**

A low-level client representing Amazon CloudSearch:

```
import boto3

client = boto3.client('cloudsearch')
```

These are the available methods:

- `build_suggesters()`
- `can_paginate()`
- `create_domain()`
- `define_analysis_scheme()`
- `define_expression()`
- `define_index_field()`
- `define_suggester()`
- `delete_analysis_scheme()`
- `delete_domain()`
- `delete_expression()`
- `delete_index_field()`
- `delete_suggester()`
- `describe_analysis_schemes()`
- `describe_availability_options()`
- `describe_domains()`
- `describe_expressions()`
- `describe_index_fields()`
- `describe_scaling_parameters()`

- `describe_service_access_policies()`
- `describe_suggesters()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_waiter()`
- `index_documents()`
- `list_domain_names()`
- `update_availability_options()`
- `update_scaling_parameters()`
- `update_service_access_policies()`

build_suggesters (***kwargs*)

Indexes the search suggestions. For more information, see [Configuring Suggesters](#) in the *Amazon CloudSearch Developer Guide*.

Request Syntax

```
response = client.build_suggesters(
    DomainName='string'
)
```

Parameters **DomainName** (*string*) – [REQUIRED]

A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

Return type dict

Returns**Response Syntax**

```
{
    'FieldNames': [
        'string',
    ]
}
```

Response Structure

- (*dict*) –

The result of a `BuildSuggester` request. Contains a list of the fields used for suggestions.

- **FieldNames** (*list*) –

A list of field names.

- * (*string*) –

A string that represents the name of an index field. CloudSearch supports regular index fields as well as dynamic fields. A dynamic field's name defines a pattern that begins or ends with a wildcard. Any document fields that don't map to a regular index field but do match a dynamic field's pattern are configured with the dynamic field's indexing options.

Regular field names begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and _ (underscore). Dynamic field names must begin or end with a wildcard (*). The wildcard can also

be the only character in a dynamic field name. Multiple wildcards, and wildcards embedded within a string are not supported.

The name `score` is reserved and cannot be used as a field name. To reference a document's ID, you can use the name `_id`.

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

create_domain (***kwargs*)

Creates a new search domain. For more information, see [Creating a Search Domain](#) in the *Amazon CloudSearch Developer Guide*.

Request Syntax

```
response = client.create_domain(
    DomainName='string'
)
```

Parameters *DomainName* (*string*) – [REQUIRED]

A name for the domain you are creating. Allowed characters are a-z (lower-case letters), 0-9, and hyphen (-). Domain names must start with a letter or number and be at least 3 and no more than 28 characters long.

Return type dict

Returns

Response Syntax

```
{
    'DomainStatus': {
        'DomainId': 'string',
        'DomainName': 'string',
        'ARN': 'string',
        'Created': True|False,
        'Deleted': True|False,
        'DocService': {
            'Endpoint': 'string'
        },
        'SearchService': {
            'Endpoint': 'string'
        },
        'RequiresIndexDocuments': True|False,
        'Processing': True|False,
        'SearchInstanceType': 'string',
        'SearchPartitionCount': 123,
        'SearchInstanceCount': 123,
        'Limits': {
            'MaximumReplicationCount': 123,
            'MaximumPartitionCount': 123
        }
    }
}
```

Response Structure

- *(dict)* –

The result of a `CreateDomainRequest` . Contains the status of a newly created domain.

- **DomainStatus** *(dict)* –

The current status of the search domain.

- * **DomainId** *(string)* –

An internally generated unique identifier for a domain.

- * **DomainName** *(string)* –

A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

- * **ARN** *(string)* –

The Amazon Resource Name (ARN) of the search domain. See [Identifiers for IAM Entities](#) in *Using AWS Identity and Access Management* for more information.

- * **Created** *(boolean)* –

True if the search domain is created. It can take several minutes to initialize a domain when `CreateDomain` is called. Newly created search domains are returned from `DescribeDomains` with a false value for `Created` until domain creation is complete.

- * **Deleted** *(boolean)* –

True if the search domain has been deleted. The system must clean up resources dedicated to the search domain when `DeleteDomain` is called. Newly deleted search domains are returned from `DescribeDomains` with a true value for `IsDeleted` for several minutes until resource cleanup is complete.

- * **DocService** *(dict)* –

The service endpoint for updating documents in a search domain.

- **Endpoint** *(string)* –

The endpoint to which service requests can be submitted. For example, `search-imdb-movies-oopcnjfn6ugofer3zx5iadxxca.eu-west-1.amazonaws.com` or `doc-imdb-movies-oopcnjfn6ugofer3zx5iadxxca.eu-west-1.amazonaws.com`.

- * **SearchService** *(dict)* –

The service endpoint for requesting search results from a search domain.

- **Endpoint** *(string)* –

The endpoint to which service requests can be submitted. For example, `search-imdb-movies-oopcnjfn6ugofer3zx5iadxxca.eu-west-1.amazonaws.com` or `doc-imdb-movies-oopcnjfn6ugofer3zx5iadxxca.eu-west-1.amazonaws.com`.

- * **RequiresIndexDocuments** *(boolean)* –

True if `IndexDocuments` needs to be called to activate the current domain configuration.

- * **Processing** (*boolean*) –
True if processing is being done to activate the current domain configuration.
- * **SearchInstanceType** (*string*) –
The instance type that is being used to process search requests.
- * **SearchPartitionCount** (*integer*) –
The number of partitions across which the search index is spread.
- * **SearchInstanceCount** (*integer*) –
The number of search instances that are available to process search requests.
- * **Limits** (*dict*) –
 - **MaximumReplicationCount** (*integer*) –
 - **MaximumPartitionCount** (*integer*) –

define_analysis_scheme (***kwargs*)

Configures an analysis scheme that can be applied to a `text` or `text-array` field to define language-specific text processing options. For more information, see [Configuring Analysis Schemes](#) in the *Amazon CloudSearch Developer Guide*.

Request Syntax

```
response = client.define_analysis_scheme(
    DomainName='string',
    AnalysisScheme={
        'AnalysisSchemeName': 'string',
        'AnalysisSchemeLanguage': 'ar'|'bg'|'ca'|'cs'|'da'|'de'|'el'|'en'|'es'|'eu'|'fa'|'f
        'AnalysisOptions': {
            'Synonyms': 'string',
            'Stopwords': 'string',
            'StemmingDictionary': 'string',
            'JapaneseTokenizationDictionary': 'string',
            'AlgorithmicStemming': 'none'|'minimal'|'light'|'full'
        }
    }
)
```

Parameters

- **DomainName** (*string*) – **[REQUIRED]**

A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

- **AnalysisScheme** (*dict*) – **[REQUIRED]**

Configuration information for an analysis scheme. Each analysis scheme has a unique name and specifies the language of the text to be processed. The following options can be configured for an analysis scheme: `Synonyms`, `Stopwords`, `StemmingDictionary`, `JapaneseTokenizationDictionary` and `AlgorithmicStemming`.

- **AnalysisSchemeName** (*string*) – **[REQUIRED]**

Names must begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and _ (underscore).

- **AnalysisSchemeLanguage** (*string*) – **[REQUIRED]**

An [IETF RFC 4646](#) language code or `mul` for multiple languages.

– **AnalysisOptions** (*dict*) –

Synonyms, stopwords, and stemming options for an analysis scheme. Includes tokenization dictionary for Japanese.

* **Synonyms** (*string*) –

A JSON object that defines synonym groups and aliases. A synonym group is an array of arrays, where each sub-array is a group of terms where each term in the group is considered a synonym of every other term in the group. The `aliases` value is an object that contains a collection of `string:value` pairs where the `string` specifies a term and the array of values specifies each of the aliases for that term. An alias is considered a synonym of the specified term, but the term is not considered a synonym of the alias. For more information about specifying synonyms, see [Synonyms](#) in the *Amazon CloudSearch Developer Guide*.

* **Stopwords** (*string*) –

A JSON array of terms to ignore during indexing and searching. For example, `["a", "an", "the", "of"]`. The stopwords dictionary must explicitly list each word you want to ignore. Wildcards and regular expressions are not supported.

* **StemmingDictionary** (*string*) –

A JSON object that contains a collection of `string:value` pairs that each map a term to its stem. For example, `{"term1": "stem1", "term2": "stem2", "term3": "stem3"}`. The stemming dictionary is applied in addition to any algorithmic stemming. This enables you to override the results of the algorithmic stemming to correct specific cases of overstemming or understemming. The maximum size of a stemming dictionary is 500 KB.

* **JapaneseTokenizationDictionary** (*string*) –

A JSON array that contains a collection of terms, tokens, readings and part of speech for Japanese Tokenization. The Japanese tokenization dictionary enables you to override the default tokenization for selected terms. This is only valid for Japanese language fields.

* **AlgorithmicStemming** (*string*) –

The level of algorithmic stemming to perform: `none`, `minimal`, `light`, or `full`. The available levels vary depending on the language. For more information, see [Language Specific Text Processing Settings](#) in the *Amazon CloudSearch Developer Guide*.

Return type `dict`

Returns

Response Syntax

```
{
  'AnalysisScheme': {
    'Options': {
      'AnalysisSchemeName': 'string',
      'AnalysisSchemeLanguage': 'ar'|'bg'|'ca'|'cs'|'da'|'de'|'el'|'en'|'es'|
      'AnalysisOptions': {
        'Synonyms': 'string',
        'Stopwords': 'string',
```



```

        'StemmingDictionary': 'string',
        'JapaneseTokenizationDictionary': 'string',
        'AlgorithmicStemming': 'none'|'minimal'|'light'|'full'
    },
    'Status': {
        'CreationDate': datetime(2015, 1, 1),
        'UpdateDate': datetime(2015, 1, 1),
        'UpdateVersion': 123,
        'State': 'RequiresIndexDocuments'|'Processing'|'Active'|'FailedToValid
        'PendingDeletion': True|False
    }
}

```

Response Structure

- (*dict*) –

The result of a “DefineAnalysisScheme” request. Contains the status of the newly-configured analysis scheme.

- **AnalysisScheme** (*dict*) –

The status and configuration of an AnalysisScheme.

- * **Options** (*dict*) –

Configuration information for an analysis scheme. Each analysis scheme has a unique name and specifies the language of the text to be processed. The following options can be configured for an analysis scheme: Synonyms, Stopwords, StemmingDictionary, JapaneseTokenizationDictionary and AlgorithmicStemming.

- **AnalysisSchemeName** (*string*) –

Names must begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and _ (underscore).

- **AnalysisSchemeLanguage** (*string*) –

An [IETF RFC 4646](#) language code or mul for multiple languages.

- **AnalysisOptions** (*dict*) –

Synonyms, stopwords, and stemming options for an analysis scheme. Includes tokenization dictionary for Japanese.

- **Synonyms** (*string*) –

A JSON object that defines synonym groups and aliases. A synonym group is an array of arrays, where each sub-array is a group of terms where each term in the group is considered a synonym of every other term in the group. The aliases value is an object that contains a collection of string:value pairs where the string specifies a term and the array of values specifies each of the aliases for that term. An alias is considered a synonym of the specified term, but the term is not considered a synonym of the alias. For more information about specifying synonyms, see [Synonyms](#) in the *Amazon CloudSearch Developer Guide*.

- **Stopwords** (*string*) –

A JSON array of terms to ignore during indexing and searching. For example, ["a", "an", "the", "of"] . The stopwords dictionary must explicitly list each word you want to ignore. Wildcards and regular expressions are not supported.

- **StemmingDictionary** (*string*) –

A JSON object that contains a collection of string:value pairs that each map a term to its stem. For example, {"term1": "stem1", "term2": "stem2", "term3": "stem3"} . The stemming dictionary is applied in addition to any algorithmic stemming. This enables you to override the results of the algorithmic stemming to correct specific cases of overstemming or understemming. The maximum size of a stemming dictionary is 500 KB.

- **JapaneseTokenizationDictionary** (*string*) –

A JSON array that contains a collection of terms, tokens, readings and part of speech for Japanese Tokenization. The Japanese tokenization dictionary enables you to override the default tokenization for selected terms. This is only valid for Japanese language fields.

- **AlgorithmicStemming** (*string*) –

The level of algorithmic stemming to perform: `none` , `minimal` , `light` , or `full` . The available levels vary depending on the language. For more information, see [Language Specific Text Processing Settings](#) in the *Amazon CloudSearch Developer Guide*

- * **Status** (*dict*) –

The status of domain configuration option.

- **CreationDate** (*datetime*) –

A timestamp for when this option was created.

- **UpdateDate** (*datetime*) –

A timestamp for when this option was last updated.

- **UpdateVersion** (*integer*) –

A unique integer that indicates when this option was last updated.

- **State** (*string*) –

The state of processing a change to an option. Possible values:

- **RequiresIndexDocuments** : the option's latest value will not be deployed until IndexDocuments has been called and indexing is complete.
- **Processing** : the option's latest value is in the process of being activated.
- **Active** : the option's latest value is completely deployed.
- **FailedToValidate** : the option value is not compatible with the domain's data and cannot be used to index the data. You must either modify the option value or update or remove the incompatible documents.
- **PendingDeletion** (*boolean*) –

Indicates that the option will be deleted once processing is

complete.

define_expression (**kwargs)

Configures an “Expression” for the search domain. Used to create new expressions and modify existing ones. If the expression exists, the new configuration replaces the old one. For more information, see [Configuring Expressions](#) in the *Amazon CloudSearch Developer Guide*.

Request Syntax

```
response = client.define_expression(
    DomainName='string',
    Expression={
        'ExpressionName': 'string',
        'ExpressionValue': 'string'
    }
)
```

Parameters

- **DomainName** (*string*) – **[REQUIRED]**

A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

- **Expression** (*dict*) – **[REQUIRED]**

A named expression that can be evaluated at search time. Can be used to sort the search results, define other expressions, or return computed information in the search results.

- **ExpressionName** (*string*) – **[REQUIRED]**

Names must begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and _ (underscore).

- **ExpressionValue** (*string*) – **[REQUIRED]**

The expression to evaluate for sorting while processing a search request. The Expression syntax is based on JavaScript expressions. For more information, see [Configuring Expressions](#) in the *Amazon CloudSearch Developer Guide*.

Return type dict

Returns

Response Syntax

```
{
    'Expression': {
        'Options': {
            'ExpressionName': 'string',
            'ExpressionValue': 'string'
        },
        'Status': {
            'CreationDate': datetime(2015, 1, 1),
            'UpdateDate': datetime(2015, 1, 1),
            'UpdateVersion': 123,
            'State': 'RequiresIndexDocuments'|'Processing'|'Active'|'FailedToValid
            'PendingDeletion': True|False
        }
    }
}
```

Response Structure

- (*dict*) –

The result of a `DefineExpression` request. Contains the status of the newly-configured expression.

- **Expression** (*dict*) –

The value of an `Expression` and its current status.

- * **Options** (*dict*) –

The expression that is evaluated for sorting while processing a search request.

- **ExpressionName** (*string*) –

Names must begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and _ (underscore).

- **ExpressionValue** (*string*) –

The expression to evaluate for sorting while processing a search request. The `Expression` syntax is based on JavaScript expressions. For more information, see [Configuring Expressions](#) in the *Amazon CloudSearch Developer Guide*.

- * **Status** (*dict*) –

The status of domain configuration option.

- **CreationDate** (*datetime*) –

A timestamp for when this option was created.

- **UpdateDate** (*datetime*) –

A timestamp for when this option was last updated.

- **UpdateVersion** (*integer*) –

A unique integer that indicates when this option was last updated.

- **State** (*string*) –

The state of processing a change to an option. Possible values:

- `RequiresIndexDocuments` : the option's latest value will not be deployed until `IndexDocuments` has been called and indexing is complete.
- `Processing` : the option's latest value is in the process of being activated.
- `Active` : the option's latest value is completely deployed.
- `FailedToValidate` : the option value is not compatible with the domain's data and cannot be used to index the data. You must either modify the option value or update or remove the incompatible documents.
- **PendingDeletion** (*boolean*) –

Indicates that the option will be deleted once processing is complete.

define_index_field (***kwargs*)

Configures an “`IndexField`” for the search domain. Used to create new fields and modify existing ones. You must specify the name of the domain you are configuring and an index field configuration. The index field configuration specifies a unique name, the index field type, and the options you want to configure for the field. The options you can specify depend on the “`IndexFieldType`”. If the field exists, the new

configuration replaces the old one. For more information, see [Configuring Index Fields](#) in the *Amazon CloudSearch Developer Guide*.

Request Syntax

```
response = client.define_index_field(
    DomainName='string',
    IndexField={
        'IndexFieldName': 'string',
        'IndexFieldType': 'int'|'double'|'literal'|'text'|'date'|'latlon'|'int-array'|'doub
        'IntOptions': {
            'DefaultValue': 123,
            'SourceField': 'string',
            'FacetEnabled': True|False,
            'SearchEnabled': True|False,
            'ReturnEnabled': True|False,
            'SortEnabled': True|False
        },
        'DoubleOptions': {
            'DefaultValue': 123.0,
            'SourceField': 'string',
            'FacetEnabled': True|False,
            'SearchEnabled': True|False,
            'ReturnEnabled': True|False,
            'SortEnabled': True|False
        },
        'LiteralOptions': {
            'DefaultValue': 'string',
            'SourceField': 'string',
            'FacetEnabled': True|False,
            'SearchEnabled': True|False,
            'ReturnEnabled': True|False,
            'SortEnabled': True|False
        },
        'TextOptions': {
            'DefaultValue': 'string',
            'SourceField': 'string',
            'ReturnEnabled': True|False,
            'SortEnabled': True|False,
            'HighlightEnabled': True|False,
            'AnalysisScheme': 'string'
        },
        'DateOptions': {
            'DefaultValue': 'string',
            'SourceField': 'string',
            'FacetEnabled': True|False,
            'SearchEnabled': True|False,
            'ReturnEnabled': True|False,
            'SortEnabled': True|False
        },
        'LatLonOptions': {
            'DefaultValue': 'string',
            'SourceField': 'string',
            'FacetEnabled': True|False,
            'SearchEnabled': True|False,
            'ReturnEnabled': True|False,
            'SortEnabled': True|False
        },
        'IntArrayOptions': {
```

```
        'DefaultValue': 123,
        'SourceFields': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False
    },
    'DoubleArrayOptions': {
        'DefaultValue': 123.0,
        'SourceFields': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False
    },
    'LiteralArrayOptions': {
        'DefaultValue': 'string',
        'SourceFields': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False
    },
    'TextArrayOptions': {
        'DefaultValue': 'string',
        'SourceFields': 'string',
        'ReturnEnabled': True|False,
        'HighlightEnabled': True|False,
        'AnalysisScheme': 'string'
    },
    'DateArrayOptions': {
        'DefaultValue': 'string',
        'SourceFields': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False
    }
}
```

Parameters

- **DomainName** (*string*) – [REQUIRED]

A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

- **IndexField** (*dict*) – [REQUIRED]

The index field and field options you want to configure.

- **IndexFieldName** (*string*) – [REQUIRED]

A string that represents the name of an index field. CloudSearch supports regular index fields as well as dynamic fields. A dynamic field's name defines a pattern that begins or ends with a wildcard. Any document fields that don't map to a regular index field but do match a dynamic field's pattern are configured with the dynamic field's indexing options.

Regular field names begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and _ (underscore). Dynamic field names must begin or end with a wildcard (*). The wildcard can also be the only character in a dynamic field name. Multiple wildcards, and wildcards em-

bedded within a string are not supported.

The name `score` is reserved and cannot be used as a field name. To reference a document's ID, you can use the name `_id`.

– **IndexFieldType** (*string*) – [REQUIRED]

The type of field. The valid options for a field depend on the field type. For more information about the supported field types, see [Configuring Index Fields](#) in the *Amazon CloudSearch Developer Guide*.

– **IntOptions** (*dict*) –

Options for a 64-bit signed integer field. Present if `IndexFieldType` specifies the field is of type `int`. All options are enabled by default.

- * **DefaultValue** (*integer*) – A value to use for the field if the field isn't specified for a document. This can be important if you are using the field in an expression and that field is not present in every document.
- * **SourceField** (*string*) –

The name of the source field to map to the field.

- * **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- * **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- * **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- * **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

– **DoubleOptions** (*dict*) –

Options for a double-precision 64-bit floating point field. Present if `IndexFieldType` specifies the field is of type `double`. All options are enabled by default.

- * **DefaultValue** (*float*) –

A value to use for the field if the field isn't specified for a document. This can be important if you are using the field in an expression and that field is not present in every document.

- * **SourceField** (*string*) –

The name of the source field to map to the field.

- * **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- * **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- * **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- * **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

– **LiteralOptions** (*dict*) –

Options for literal field. Present if `IndexFieldType` specifies the field is of type `literal`. All options are enabled by default.

- * **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.

- * **SourceField** (*string*) –

A string that represents the name of an index field. CloudSearch supports regular index fields as well as dynamic fields. A dynamic field's name defines a pattern that begins or ends with a wildcard. Any document fields that don't map to a regular index field but do match a dynamic field's pattern are configured with the dynamic field's indexing options.

Regular field names begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and _ (underscore). Dynamic field names must begin or end with a wildcard (*). The wildcard can also be the only character in a dynamic field name. Multiple wildcards, and wildcards embedded within a string are not supported.

The name `score` is reserved and cannot be used as a field name. To reference a document's ID, you can use the name `_id`.

- * **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- * **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- * **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- * **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

- **TextOptions** (*dict*) –

Options for text field. Present if `IndexFieldType` specifies the field is of type `text`. A `text` field is always searchable. All options are enabled by default.

- * **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.

- * **SourceField** (*string*) –

A string that represents the name of an index field. CloudSearch supports regular index fields as well as dynamic fields. A dynamic field's name defines a pattern that begins or ends with a wildcard. Any document fields that don't map to a regular index field but do match a dynamic field's pattern are configured with the dynamic field's indexing options.

Regular field names begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and _ (underscore). Dynamic field names must begin or end with a wildcard (*). The wildcard can also be the only character in a dynamic field name. Multiple wildcards, and wildcards embedded within a string are not supported.

The name `score` is reserved and cannot be used as a field name. To reference a document's ID, you can use the name `_id`.

- * **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- * **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

- * **HighlightEnabled** (*boolean*) –

Whether highlights can be returned for the field.

- * **AnalysisScheme** (*string*) –

The name of an analysis scheme for a `text` field.

- **DateOptions** (*dict*) –

Options for a date field. Dates and times are specified in UTC (Coordinated Universal Time) according to IETF RFC3339: `yyyy-mm-ddT00:00:00Z`. Present if `IndexFieldType` specifies the field is of type `date`. All options are enabled by default.

- * **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.

- * **SourceField** (*string*) –

A string that represents the name of an index field. CloudSearch supports regular index fields as well as dynamic fields. A dynamic field's name defines a pattern that begins or ends with a wildcard. Any document fields that don't map to a regular index field but do match a dynamic field's pattern are configured with the dynamic field's indexing options.

Regular field names begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and `_` (underscore). Dynamic field names must begin or end with a wildcard (`*`). The wildcard can also be the only character in a dynamic field name. Multiple wildcards, and wildcards embedded within a string are not supported.

The name `score` is reserved and cannot be used as a field name. To reference a document's ID, you can use the name `_id`.

- * **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- * **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- * **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- * **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

- **LatLonOptions** (*dict*) –

Options for a `latlon` field. A `latlon` field contains a location stored as a latitude and longitude value pair. Present if `IndexFieldType` specifies the field is of type `latlon`. All options are enabled by default.

- * **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.

- * **SourceField** (*string*) –

A string that represents the name of an index field. CloudSearch supports regular index fields as well as dynamic fields. A dynamic field's name defines a pattern that begins or ends with a wildcard. Any document fields that don't map to a regular index field but do

match a dynamic field's pattern are configured with the dynamic field's indexing options.

Regular field names begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and `_` (underscore). Dynamic field names must begin or end with a wildcard (`*`). The wildcard can also be the only character in a dynamic field name. Multiple wildcards, and wildcards embedded within a string are not supported.

The name `score` is reserved and cannot be used as a field name. To reference a document's ID, you can use the name `_id`.

- * **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- * **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- * **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- * **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

- **IntArrayOptions** (*dict*) –

Options for a field that contains an array of 64-bit signed integers. Present if `IndexFieldType` specifies the field is of type `int-array`. All options are enabled by default.

- * **DefaultValue** (*integer*) – A value to use for the field if the field isn't specified for a document.

- * **SourceFields** (*string*) –

A list of source fields to map to the field.

- * **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- * **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- * **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **DoubleArrayOptions** (*dict*) –

Options for a field that contains an array of double-precision 64-bit floating point values. Present if `IndexFieldType` specifies the field is of type `double-array`. All options are enabled by default.

- * **DefaultValue** (*float*) – A value to use for the field if the field isn't specified for a document.

- * **SourceFields** (*string*) –

A list of source fields to map to the field.

- * **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- * **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- * **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **LiteralArrayOptions** (*dict*) –

Options for a field that contains an array of literal strings. Present if `IndexFieldType` specifies the field is of type `literal-array`. All options are enabled by default.

- * **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.

- * **SourceFields** (*string*) –

A list of source fields to map to the field.

- * **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- * **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- * **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **TextArrayOptions** (*dict*) –

Options for a field that contains an array of text strings. Present if `IndexFieldType` specifies the field is of type `text-array`. A `text-array` field is always searchable. All options are enabled by default.

- * **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.

- * **SourceFields** (*string*) –

A list of source fields to map to the field.

- * **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- * **HighlightEnabled** (*boolean*) –

Whether highlights can be returned for the field.

- * **AnalysisScheme** (*string*) –

The name of an analysis scheme for a `text-array` field.

- **DateArrayOptions** (*dict*) –

Options for a field that contains an array of dates. Present if `IndexFieldType` specifies the field is of type `date-array`. All options are enabled by default.

- * **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.

- * **SourceFields** (*string*) –

A list of source fields to map to the field.

- * **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- * **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

* **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

Return type dict

Returns

Response Syntax

```
{
  'IndexField': {
    'Options': {
      'IndexFieldName': 'string',
      'IndexFieldType': 'int'|'double'|'literal'|'text'|'date'|'latlon'|'int',
      'IntOptions': {
        'DefaultValue': 123,
        'SourceField': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False,
        'SortEnabled': True|False
      },
      'DoubleOptions': {
        'DefaultValue': 123.0,
        'SourceField': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False,
        'SortEnabled': True|False
      },
      'LiteralOptions': {
        'DefaultValue': 'string',
        'SourceField': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False,
        'SortEnabled': True|False
      },
      'TextOptions': {
        'DefaultValue': 'string',
        'SourceField': 'string',
        'ReturnEnabled': True|False,
        'SortEnabled': True|False,
        'HighlightEnabled': True|False,
        'AnalysisScheme': 'string'
      },
      'DateOptions': {
        'DefaultValue': 'string',
        'SourceField': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False,
        'SortEnabled': True|False
      },
      'LatLonOptions': {
        'DefaultValue': 'string',
        'SourceField': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
```

```

        'ReturnEnabled': True|False,
        'SortEnabled': True|False
    },
    'IntArrayOptions': {
        'DefaultValue': 123,
        'SourceFields': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False
    },
    'DoubleArrayOptions': {
        'DefaultValue': 123.0,
        'SourceFields': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False
    },
    'LiteralArrayOptions': {
        'DefaultValue': 'string',
        'SourceFields': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False
    },
    'TextArrayOptions': {
        'DefaultValue': 'string',
        'SourceFields': 'string',
        'ReturnEnabled': True|False,
        'HighlightEnabled': True|False,
        'AnalysisScheme': 'string'
    },
    'DateArrayOptions': {
        'DefaultValue': 'string',
        'SourceFields': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False
    }
},
'Status': {
    'CreationDate': datetime(2015, 1, 1),
    'UpdateDate': datetime(2015, 1, 1),
    'UpdateVersion': 123,
    'State': 'RequiresIndexDocuments'|'Processing'|'Active'|'FailedToValidate',
    'PendingDeletion': True|False
}
}
}

```

Response Structure

- (*dict*) –

The result of a “DefineIndexField” request. Contains the status of the newly-configured index field.

- **IndexField** (*dict*) –

The value of an IndexField and its current status.

- * **Options** (*dict*) –

Configuration information for a field in the index, including its name, type, and options. The supported options depend on the “`IndexFieldType`”.

- **IndexFieldName** (*string*) –

A string that represents the name of an index field. CloudSearch supports regular index fields as well as dynamic fields. A dynamic field’s name defines a pattern that begins or ends with a wildcard. Any document fields that don’t map to a regular index field but do match a dynamic field’s pattern are configured with the dynamic field’s indexing options.

Regular field names begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and `_` (underscore). Dynamic field names must begin or end with a wildcard (`*`). The wildcard can also be the only character in a dynamic field name. Multiple wildcards, and wildcards embedded within a string are not supported.

The name `score` is reserved and cannot be used as a field name. To reference a document’s ID, you can use the name `_id`.

- **IndexFieldType** (*string*) –

The type of field. The valid options for a field depend on the field type. For more information about the supported field types, see [Configuring Index Fields](#) in the *Amazon CloudSearch Developer Guide*.

- **IntOptions** (*dict*) –

Options for a 64-bit signed integer field. Present if `IndexFieldType` specifies the field is of type `int`. All options are enabled by default.

- **DefaultValue** (*integer*) – A value to use for the field if the field isn’t specified for a document. This can be important if you are using the field in an expression and that field is not present in every document.

- **SourceField** (*string*) –

The name of the source field to map to the field.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

- **DoubleOptions** (*dict*) –

Options for a double-precision 64-bit floating point field. Present if `IndexFieldType` specifies the field is of type `double`. All options are enabled by default.

- **DefaultValue** (*float*) –

A value to use for the field if the field isn't specified for a document. This can be important if you are using the field in an expression and that field is not present in every document.

- **SourceField** (*string*) –

The name of the source field to map to the field.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

- **LiteralOptions** (*dict*) –

Options for literal field. Present if `IndexFieldType` specifies the field is of type `literal`. All options are enabled by default.

- **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.

- **SourceField** (*string*) –

A string that represents the name of an index field. CloudSearch supports regular index fields as well as dynamic fields. A dynamic field's name defines a pattern that begins or ends with a wildcard. Any document fields that don't map to a regular index field but do match a dynamic field's pattern are configured with the dynamic field's indexing options.

Regular field names begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and `_` (underscore). Dynamic field names must begin or end with a wildcard (`*`). The wildcard can also be the only character in a dynamic field name. Multiple wildcards, and wildcards embedded within a string are not supported.

The name `score` is reserved and cannot be used as a field name. To reference a document's ID, you can use the name `_id`.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

- **TextOptions** (*dict*) –

Options for text field. Present if `IndexFieldType` specifies the field is of type `text`. A `text` field is always searchable. All options are enabled by default.

- **DefaultValue** (*string*) – A value to use for the field if the field isn’t specified for a document.
- **SourceField** (*string*) –

A string that represents the name of an index field. CloudSearch supports regular index fields as well as dynamic fields. A dynamic field’s name defines a pattern that begins or ends with a wildcard. Any document fields that don’t map to a regular index field but do match a dynamic field’s pattern are configured with the dynamic field’s indexing options.

Regular field names begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and `_` (underscore). Dynamic field names must begin or end with a wildcard (`*`). The wildcard can also be the only character in a dynamic field name. Multiple wildcards, and wildcards embedded within a string are not supported.

The name `score` is reserved and cannot be used as a field name. To reference a document’s ID, you can use the name `_id`.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

- **HighlightEnabled** (*boolean*) –

Whether highlights can be returned for the field.

- **AnalysisScheme** (*string*) –

The name of an analysis scheme for a `text` field.

- **DateOptions** (*dict*) –

Options for a date field. Dates and times are specified in UTC (Coordinated Universal Time) according to IETF RFC3339: `yyyy-mm-ddT00:00:00Z`. Present if `IndexFieldType` specifies the field is of type `date`. All options are enabled by default.

- **DefaultValue** (*string*) – A value to use for the field if the field isn’t specified for a document.
- **SourceField** (*string*) –

A string that represents the name of an index field. CloudSearch supports regular index fields as well as dynamic fields. A dynamic field’s name defines a pattern that begins or ends with a wildcard. Any document fields that don’t map to a regular index field but do match a dynamic field’s pattern are configured with the dynamic field’s indexing options.

Regular field names begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and `_` (underscore). Dynamic field names must begin or end with a wildcard (`*`). The wildcard can also be the only character in a

dynamic field name. Multiple wildcards, and wildcards embedded within a string are not supported.

The name `score` is reserved and cannot be used as a field name. To reference a document's ID, you can use the name `_id`.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

- **LatLonOptions** (*dict*) –

Options for a latlon field. A latlon field contains a location stored as a latitude and longitude value pair. Present if `IndexFieldType` specifies the field is of type `latlon`. All options are enabled by default.

- **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.

- **SourceField** (*string*) –

A string that represents the name of an index field. CloudSearch supports regular index fields as well as dynamic fields. A dynamic field's name defines a pattern that begins or ends with a wildcard. Any document fields that don't map to a regular index field but do match a dynamic field's pattern are configured with the dynamic field's indexing options.

Regular field names begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and `_` (underscore). Dynamic field names must begin or end with a wildcard (`*`). The wildcard can also be the only character in a dynamic field name. Multiple wildcards, and wildcards embedded within a string are not supported.

The name `score` is reserved and cannot be used as a field name. To reference a document's ID, you can use the name `_id`.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

- **IntArrayOptions** (*dict*) –

Options for a field that contains an array of 64-bit signed integers. Present if `IndexFieldType` specifies the field is of type `int-array`. All options are enabled by default.

- **DefaultValue** (*integer*) – A value to use for the field if the field isn't specified for a document.
- **SourceFields** (*string*) –

A list of source fields to map to the field.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **DoubleArrayOptions** (*dict*) –

Options for a field that contains an array of double-precision 64-bit floating point values. Present if `IndexFieldType` specifies the field is of type `double-array`. All options are enabled by default.

- **DefaultValue** (*float*) – A value to use for the field if the field isn't specified for a document.
- **SourceFields** (*string*) –

A list of source fields to map to the field.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **LiteralArrayOptions** (*dict*) –

Options for a field that contains an array of literal strings. Present if `IndexFieldType` specifies the field is of type `literal-array`. All options are enabled by default.

- **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.
- **SourceFields** (*string*) –

A list of source fields to map to the field.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **TextArrayOptions** (*dict*) –

Options for a field that contains an array of text strings. Present if `IndexFieldType` specifies the field is of type `text-array`. A `text-array` field is always searchable. All options are enabled by default.

- **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.

- **SourceFields** (*string*) –

A list of source fields to map to the field.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **HighlightEnabled** (*boolean*) –

Whether highlights can be returned for the field.

- **AnalysisScheme** (*string*) –

The name of an analysis scheme for a `text-array` field.

- **DateArrayOptions** (*dict*) –

Options for a field that contains an array of dates. Present if `IndexFieldType` specifies the field is of type `date-array`. All options are enabled by default.

- **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.

- **SourceFields** (*string*) –

A list of source fields to map to the field.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- * **Status** (*dict*) –

The status of domain configuration option.

- **CreationDate** (*datetime*) –

A timestamp for when this option was created.

- **UpdateDate** (*datetime*) –

A timestamp for when this option was last updated.

- **UpdateVersion** (*integer*) –

A unique integer that indicates when this option was last updated.

- **State** (*string*) –

The state of processing a change to an option. Possible values:

- `RequiresIndexDocuments`: the option's latest value will not be deployed until `IndexDocuments` has been called and indexing is complete.
- `Processing`: the option's latest value is in the process of being activated.
- `Active`: the option's latest value is completely deployed.

- **FailedToValidate** : the option value is not compatible with the domain's data and cannot be used to index the data. You must either modify the option value or update or remove the incompatible documents.
- **PendingDeletion** (*boolean*) –

Indicates that the option will be deleted once processing is complete.

define_suggester (***kwargs*)

Configures a suggester for a domain. A suggester enables you to display possible matches before users finish typing their queries. When you configure a suggester, you must specify the name of the text field you want to search for possible matches and a unique name for the suggester. For more information, see [Getting Search Suggestions](#) in the *Amazon CloudSearch Developer Guide*.

Request Syntax

```
response = client.define_suggester(
    DomainName='string',
    Suggester={
        'SuggesterName': 'string',
        'DocumentSuggesterOptions': {
            'SourceField': 'string',
            'FuzzyMatching': 'none'|'low'|'high',
            'SortExpression': 'string'
        }
    }
)
```

Parameters

- **DomainName** (*string*) – **[REQUIRED]**

A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

- **Suggester** (*dict*) – **[REQUIRED]**

Configuration information for a search suggester. Each suggester has a unique name and specifies the text field you want to use for suggestions. The following options can be configured for a suggester: `FuzzyMatching`, `SortExpression`.

- **SuggesterName** (*string*) – **[REQUIRED]**

Names must begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and _ (underscore).

- **DocumentSuggesterOptions** (*dict*) – **[REQUIRED]**

Options for a search suggester.

- * **SourceField** (*string*) – **[REQUIRED]**

The name of the index field you want to use for suggestions.

- * **FuzzyMatching** (*string*) –

The level of fuzziness allowed when suggesting matches for a string: `none`, `low`, or `high`. With `none`, the specified string is treated as an exact prefix. With `low`, suggestions must differ from the specified string by no more than one character. With `high`, suggestions can differ by up to two characters. The default is `none`.

* **SortExpression** (*string*) –

An expression that computes a score for each suggestion to control how they are sorted. The scores are rounded to the nearest integer, with a floor of 0 and a ceiling of $2^{31}-1$. A document's relevance score is not computed for suggestions, so sort expressions cannot reference the `_score` value. To sort suggestions using a numeric field or existing expression, simply specify the name of the field or expression. If no expression is configured for the suggester, the suggestions are sorted with the closest matches listed first.

Return type dict

Returns

Response Syntax

```
{
  'Suggester': {
    'Options': {
      'SuggesterName': 'string',
      'DocumentSuggesterOptions': {
        'SourceField': 'string',
        'FuzzyMatching': 'none'|'low'|'high',
        'SortExpression': 'string'
      }
    },
    'Status': {
      'CreationDate': datetime(2015, 1, 1),
      'UpdateDate': datetime(2015, 1, 1),
      'UpdateVersion': 123,
      'State': 'RequiresIndexDocuments'|'Processing'|'Active'|'FailedToValid',
      'PendingDeletion': True|False
    }
  }
}
```

Response Structure

- (*dict*) –

The result of a `DefineSuggester` request. Contains the status of the newly-configured suggester.

- **Suggester** (*dict*) –

The value of a `Suggester` and its current status.

- * **Options** (*dict*) –

Configuration information for a search suggester. Each suggester has a unique name and specifies the text field you want to use for suggestions. The following options can be configured for a suggester: `FuzzyMatching`, `SortExpression`.

- **SuggesterName** (*string*) –

Names must begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and `_` (underscore).

- **DocumentSuggesterOptions** (*dict*) –

Options for a search suggester.

- **SourceField** (*string*) –

The name of the index field you want to use for suggestions.

- **FuzzyMatching** (*string*) –

The level of fuzziness allowed when suggesting matches for a string: `none`, `low`, or `high`. With `none`, the specified string is treated as an exact prefix. With `low`, suggestions must differ from the specified string by no more than one character. With `high`, suggestions can differ by up to two characters. The default is `none`.

- **SortExpression** (*string*) –

An expression that computes a score for each suggestion to control how they are sorted. The scores are rounded to the nearest integer, with a floor of 0 and a ceiling of $2^{31}-1$. A document's relevance score is not computed for suggestions, so sort expressions cannot reference the `_score` value. To sort suggestions using a numeric field or existing expression, simply specify the name of the field or expression. If no expression is configured for the suggester, the suggestions are sorted with the closest matches listed first.

- * **Status** (*dict*) –

The status of domain configuration option.

- **CreationDate** (*datetime*) –

A timestamp for when this option was created.

- **UpdateDate** (*datetime*) –

A timestamp for when this option was last updated.

- **UpdateVersion** (*integer*) –

A unique integer that indicates when this option was last updated.

- **State** (*string*) –

The state of processing a change to an option. Possible values:

- `RequiresIndexDocuments`: the option's latest value will not be deployed until `IndexDocuments` has been called and indexing is complete.
- `Processing`: the option's latest value is in the process of being activated.
- `Active`: the option's latest value is completely deployed.
- `FailedToValidate`: the option value is not compatible with the domain's data and cannot be used to index the data. You must either modify the option value or update or remove the incompatible documents.
- **PendingDeletion** (*boolean*) –

Indicates that the option will be deleted once processing is complete.

delete_analysis_scheme (**kwargs)

Deletes an analysis scheme. For more information, see [Configuring Analysis Schemes](#) in the *Amazon CloudSearch Developer Guide*.

Request Syntax

```
response = client.delete_analysis_scheme(  
    DomainName='string',
```

```

AnalysisSchemeName='string'
)

```

Parameters

- **DomainName** (*string*) – **[REQUIRED]**

A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

- **AnalysisSchemeName** (*string*) – **[REQUIRED]**

The name of the analysis scheme you want to delete.

Return type dict

Returns

Response Syntax

```

{
  'AnalysisScheme': {
    'Options': {
      'AnalysisSchemeName': 'string',
      'AnalysisSchemeLanguage': 'ar'|'bg'|'ca'|'cs'|'da'|'de'|'el'|'en'|'es',
      'AnalysisOptions': {
        'Synonyms': 'string',
        'Stopwords': 'string',
        'StemmingDictionary': 'string',
        'JapaneseTokenizationDictionary': 'string',
        'AlgorithmicStemming': 'none'|'minimal'|'light'|'full'
      }
    },
    'Status': {
      'CreationDate': datetime(2015, 1, 1),
      'UpdateDate': datetime(2015, 1, 1),
      'UpdateVersion': 123,
      'State': 'RequiresIndexDocuments'|'Processing'|'Active'|'FailedToValidate',
      'PendingDeletion': True|False
    }
  }
}

```

Response Structure

- (*dict*) –

The result of a DeleteAnalysisScheme request. Contains the status of the deleted analysis scheme.

- **AnalysisScheme** (*dict*) –

The status of the analysis scheme being deleted.

- * **Options** (*dict*) –

Configuration information for an analysis scheme. Each analysis scheme has a unique name and specifies the language of the text to be processed. The following options can be configured for an analysis scheme: Synonyms , Stopwords , StemmingDictionary , JapaneseTokenizationDictionary and AlgorithmicStemming.

- **AnalysisSchemeName** (*string*) –

Names must begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and _ (underscore).

- **AnalysisSchemeLanguage** (*string*) –

An [IETF RFC 4646](#) language code or `mul` for multiple languages.

- **AnalysisOptions** (*dict*) –

Synonyms, stopwords, and stemming options for an analysis scheme. Includes tokenization dictionary for Japanese.

- **Synonyms** (*string*) –

A JSON object that defines synonym groups and aliases. A synonym group is an array of arrays, where each sub-array is a group of terms where each term in the group is considered a synonym of every other term in the group. The aliases value is an object that contains a collection of string:value pairs where the string specifies a term and the array of values specifies each of the aliases for that term. An alias is considered a synonym of the specified term, but the term is not considered a synonym of the alias. For more information about specifying synonyms, see [Synonyms](#) in the *Amazon CloudSearch Developer Guide*.

- **Stopwords** (*string*) –

A JSON array of terms to ignore during indexing and searching. For example, `["a", "an", "the", "of"]`. The stopwords dictionary must explicitly list each word you want to ignore. Wildcards and regular expressions are not supported.

- **StemmingDictionary** (*string*) –

A JSON object that contains a collection of string:value pairs that each map a term to its stem. For example, `{"term1": "stem1", "term2": "stem2", "term3": "stem3"}`. The stemming dictionary is applied in addition to any algorithmic stemming. This enables you to override the results of the algorithmic stemming to correct specific cases of overstemming or understemming. The maximum size of a stemming dictionary is 500 KB.

- **JapaneseTokenizationDictionary** (*string*) –

A JSON array that contains a collection of terms, tokens, readings and part of speech for Japanese Tokenization. The Japanese tokenization dictionary enables you to override the default tokenization for selected terms. This is only valid for Japanese language fields.

- **AlgorithmicStemming** (*string*) –

The level of algorithmic stemming to perform: `none`, `minimal`, `light`, or `full`. The available levels vary depending on the language. For more information, see [Language Specific Text Processing Settings](#) in the *Amazon CloudSearch Developer Guide*.

- * **Status** (*dict*) –

The status of domain configuration option.

- **CreationDate** (*datetime*) –
A timestamp for when this option was created.
- **UpdateDate** (*datetime*) –
A timestamp for when this option was last updated.
- **UpdateVersion** (*integer*) –
A unique integer that indicates when this option was last updated.
- **State** (*string*) –
The state of processing a change to an option. Possible values:
 - **RequiresIndexDocuments** : the option's latest value will not be deployed until IndexDocuments has been called and indexing is complete.
 - **Processing** : the option's latest value is in the process of being activated.
 - **Active** : the option's latest value is completely deployed.
 - **FailedToValidate** : the option value is not compatible with the domain's data and cannot be used to index the data. You must either modify the option value or update or remove the incompatible documents.
- **PendingDeletion** (*boolean*) –
Indicates that the option will be deleted once processing is complete.

delete_domain (**kwargs)

Permanently deletes a search domain and all of its data. Once a domain has been deleted, it cannot be recovered. For more information, see [Deleting a Search Domain](#) in the *Amazon CloudSearch Developer Guide*.

Request Syntax

```
response = client.delete_domain(
    DomainName='string'
)
```

Parameters **DomainName** (*string*) – [REQUIRED]

The name of the domain you want to permanently delete.

Return type dict

Returns

Response Syntax

```
{
  'DomainStatus': {
    'DomainId': 'string',
    'DomainName': 'string',
    'ARN': 'string',
    'Created': True|False,
    'Deleted': True|False,
    'DocService': {
      'Endpoint': 'string'
    },
    'SearchService': {
```

```
        'Endpoint': 'string'
    },
    'RequiresIndexDocuments': True|False,
    'Processing': True|False,
    'SearchInstanceType': 'string',
    'SearchPartitionCount': 123,
    'SearchInstanceCount': 123,
    'Limits': {
        'MaximumReplicationCount': 123,
        'MaximumPartitionCount': 123
    }
}
```

Response Structure

- (*dict*) –

The result of a `DeleteDomain` request. Contains the status of a newly deleted domain, or no status if the domain has already been completely deleted.

- **DomainStatus** (*dict*) –

The current status of the search domain.

- * **DomainId** (*string*) –

An internally generated unique identifier for a domain.

- * **DomainName** (*string*) –

A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

- * **ARN** (*string*) –

The Amazon Resource Name (ARN) of the search domain. See [Identifiers for IAM Entities](#) in *Using AWS Identity and Access Management* for more information.

- * **Created** (*boolean*) –

True if the search domain is created. It can take several minutes to initialize a domain when `CreateDomain` is called. Newly created search domains are returned from `DescribeDomains` with a false value for `Created` until domain creation is complete.

- * **Deleted** (*boolean*) –

True if the search domain has been deleted. The system must clean up resources dedicated to the search domain when `DeleteDomain` is called. Newly deleted search domains are returned from `DescribeDomains` with a true value for `IsDeleted` for several minutes until resource cleanup is complete.

- * **DocService** (*dict*) –

The service endpoint for updating documents in a search domain.

- **Endpoint** (*string*) –

The endpoint to which service requests can be submitted. For example,

search-imdb-movies-oopcnjfn6ugofer3zx5iadxxca.eu-west-1.cl
or doc-imdb-movies-oopcnjfn6ugofer3zx5iadxxca.eu-west-1.cl

* **SearchService** (*dict*) –

The service endpoint for requesting search results from a search domain.

· **Endpoint** (*string*) –

The endpoint to which service requests can be submitted. For example,

search-imdb-movies-oopcnjfn6ugofer3zx5iadxxca.eu-west-1.amazonaws.com
or doc-imdb-movies-oopcnjfn6ugofer3zx5iadxxca.eu-west-1.amazonaws.com

* **RequiresIndexDocuments** (*boolean*) –

True if IndexDocuments needs to be called to activate the current domain configuration.

* **Processing** (*boolean*) –

True if processing is being done to activate the current domain configuration.

* **SearchInstanceType** (*string*) –

The instance type that is being used to process search requests.

* **SearchPartitionCount** (*integer*) –

The number of partitions across which the search index is spread.

* **SearchInstanceCount** (*integer*) –

The number of search instances that are available to process search requests.

* **Limits** (*dict*) –

· **MaximumReplicationCount** (*integer*) –

· **MaximumPartitionCount** (*integer*) –

delete_expression (***kwargs*)

Removes an “ Expression” from the search domain. For more information, see [Configuring Expressions](#) in the *Amazon CloudSearch Developer Guide*.

Request Syntax

```
response = client.delete_expression(
    DomainName='string',
    ExpressionName='string'
)
```

Parameters

• **DomainName** (*string*) – [REQUIRED]

A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

• **ExpressionName** (*string*) – [REQUIRED]

The name of the “ Expression” to delete.

Return type dict

Returns

Response Syntax

```
{
  'Expression': {
    'Options': {
      'ExpressionName': 'string',
      'ExpressionValue': 'string'
    },
    'Status': {
      'CreationDate': datetime(2015, 1, 1),
      'UpdateDate': datetime(2015, 1, 1),
      'UpdateVersion': 123,
      'State': 'RequiresIndexDocuments'|'Processing'|'Active'|'FailedToValid
      'PendingDeletion': True|False
    }
  }
}
```

Response Structure

- *(dict)* –

The result of a “DeleteExpression” request. Specifies the expression being deleted.

- **Expression** (*dict*) –

The status of the expression being deleted.

- * **Options** (*dict*) –

The expression that is evaluated for sorting while processing a search request.

- **ExpressionName** (*string*) –

Names must begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and _ (underscore).

- **ExpressionValue** (*string*) –

The expression to evaluate for sorting while processing a search request. The Expression syntax is based on JavaScript expressions. For more information, see [Configuring Expressions](#) in the *Amazon CloudSearch Developer Guide*.

- * **Status** (*dict*) –

The status of domain configuration option.

- **CreationDate** (*datetime*) –

A timestamp for when this option was created.

- **UpdateDate** (*datetime*) –

A timestamp for when this option was last updated.

- **UpdateVersion** (*integer*) –

A unique integer that indicates when this option was last updated.

- **State** (*string*) –

The state of processing a change to an option. Possible values:

- `RequiresIndexDocuments`: the option’s latest value will not be deployed until IndexDocuments has been called and indexing is complete.

- **Processing** : the option's latest value is in the process of being activated.
- **Active** : the option's latest value is completely deployed.
- **FailedToValidate** : the option value is not compatible with the domain's data and cannot be used to index the data. You must either modify the option value or update or remove the incompatible documents.
- **PendingDeletion** (*boolean*) –

Indicates that the option will be deleted once processing is complete.

delete_index_field (**kwargs)

Removes an “IndexField” from the search domain. For more information, see [Configuring Index Fields](#) in the *Amazon CloudSearch Developer Guide*.

Request Syntax

```
response = client.delete_index_field(
    DomainName='string',
    IndexFieldName='string'
)
```

Parameters

- **DomainName** (*string*) – **[REQUIRED]**

A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

- **IndexFieldName** (*string*) – **[REQUIRED]**

The name of the index field your want to remove from the domain's indexing options.

Return type dict

Returns

Response Syntax

```
{
  'IndexField': {
    'Options': {
      'IndexFieldName': 'string',
      'IndexFieldType': 'int'|'double'|'literal'|'text'|'date'|'latlon'|'int',
      'IntOptions': {
        'DefaultValue': 123,
        'SourceField': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False,
        'SortEnabled': True|False
      },
      'DoubleOptions': {
        'DefaultValue': 123.0,
        'SourceField': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False,
        'SortEnabled': True|False
      }
    }
  }
}
```

```
    },
    'LiteralOptions': {
        'DefaultValue': 'string',
        'SourceField': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False,
        'SortEnabled': True|False
    },
    'TextOptions': {
        'DefaultValue': 'string',
        'SourceField': 'string',
        'ReturnEnabled': True|False,
        'SortEnabled': True|False,
        'HighlightEnabled': True|False,
        'AnalysisScheme': 'string'
    },
    'DateOptions': {
        'DefaultValue': 'string',
        'SourceField': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False,
        'SortEnabled': True|False
    },
    'LatLonOptions': {
        'DefaultValue': 'string',
        'SourceField': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False,
        'SortEnabled': True|False
    },
    'IntArrayOptions': {
        'DefaultValue': 123,
        'SourceFields': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False
    },
    'DoubleArrayOptions': {
        'DefaultValue': 123.0,
        'SourceFields': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False
    },
    'LiteralArrayOptions': {
        'DefaultValue': 'string',
        'SourceFields': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False
    },
    'TextArrayOptions': {
        'DefaultValue': 'string',
        'SourceFields': 'string',
        'ReturnEnabled': True|False,
```

```

        'HighlightEnabled': True|False,
        'AnalysisScheme': 'string'
    },
    'DateArrayOptions': {
        'DefaultValue': 'string',
        'SourceFields': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False
    }
},
'Status': {
    'CreationDate': datetime(2015, 1, 1),
    'UpdateDate': datetime(2015, 1, 1),
    'UpdateVersion': 123,
    'State': 'RequiresIndexDocuments'|'Processing'|'Active'|'FailedToValid
    'PendingDeletion': True|False
}
}
}

```

Response Structure

- (*dict*) –

The result of a “DeleteIndexField” request.

- **IndexField** (*dict*) –

The status of the index field being deleted.

- * **Options** (*dict*) –

Configuration information for a field in the index, including its name, type, and options. The supported options depend on the “IndexFieldType”.

- **IndexFieldName** (*string*) –

A string that represents the name of an index field. CloudSearch supports regular index fields as well as dynamic fields. A dynamic field’s name defines a pattern that begins or ends with a wildcard. Any document fields that don’t map to a regular index field but do match a dynamic field’s pattern are configured with the dynamic field’s indexing options.

Regular field names begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and _ (underscore). Dynamic field names must begin or end with a wildcard (*). The wildcard can also be the only character in a dynamic field name. Multiple wildcards, and wildcards embedded within a string are not supported.

The name `score` is reserved and cannot be used as a field name. To reference a document’s ID, you can use the name `_id`.

- **IndexFieldType** (*string*) –

The type of field. The valid options for a field depend on the field type. For more information about the supported field types, see [Configuring Index Fields](#) in the *Amazon CloudSearch Developer Guide*.

- **IntOptions** (*dict*) –

Options for a 64-bit signed integer field. Present if `IndexFieldType` specifies the field is of type `int`. All options are enabled by default.

- **DefaultValue** (*integer*) – A value to use for the field if the field isn't specified for a document. This can be important if you are using the field in an expression and that field is not present in every document.

- **SourceField** (*string*) –

The name of the source field to map to the field.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

- **DoubleOptions** (*dict*) –

Options for a double-precision 64-bit floating point field. Present if `IndexFieldType` specifies the field is of type `double`. All options are enabled by default.

- **DefaultValue** (*float*) –

A value to use for the field if the field isn't specified for a document. This can be important if you are using the field in an expression and that field is not present in every document.

- **SourceField** (*string*) –

The name of the source field to map to the field.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

- **LiteralOptions** (*dict*) –

Options for literal field. Present if `IndexFieldType` specifies the field is of type `literal`. All options are enabled by default.

- **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.

- **SourceField** (*string*) –

A string that represents the name of an index field. CloudSearch supports regular index fields as well as dynamic fields. A dynamic field's name defines a pattern that begins or ends

with a wildcard. Any document fields that don't map to a regular index field but do match a dynamic field's pattern are configured with the dynamic field's indexing options.

Regular field names begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and `_` (underscore). Dynamic field names must begin or end with a wildcard (`*`). The wildcard can also be the only character in a dynamic field name. Multiple wildcards, and wildcards embedded within a string are not supported.

The name `score` is reserved and cannot be used as a field name. To reference a document's ID, you can use the name `_id`.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

- **TextOptions** (*dict*) –

Options for text field. Present if `IndexFieldType` specifies the field is of type `text`. A `text` field is always searchable. All options are enabled by default.

- **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.

- **SourceField** (*string*) –

A string that represents the name of an index field. CloudSearch supports regular index fields as well as dynamic fields. A dynamic field's name defines a pattern that begins or ends with a wildcard. Any document fields that don't map to a regular index field but do match a dynamic field's pattern are configured with the dynamic field's indexing options.

Regular field names begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and `_` (underscore). Dynamic field names must begin or end with a wildcard (`*`). The wildcard can also be the only character in a dynamic field name. Multiple wildcards, and wildcards embedded within a string are not supported.

The name `score` is reserved and cannot be used as a field name. To reference a document's ID, you can use the name `_id`.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

- **HighlightEnabled** (*boolean*) –

Whether highlights can be returned for the field.

- **AnalysisScheme** (*string*) –

The name of an analysis scheme for a `text` field.

- **DateOptions** (*dict*) –

Options for a date field. Dates and times are specified in UTC (Coordinated Universal Time) according to IETF RFC3339: `yyyy-mm-ddT00:00:00Z`. Present if `IndexFieldType` specifies the field is of type `date`. All options are enabled by default.

- **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.

- **SourceField** (*string*) –

A string that represents the name of an index field. CloudSearch supports regular index fields as well as dynamic fields. A dynamic field's name defines a pattern that begins or ends with a wildcard. Any document fields that don't map to a regular index field but do match a dynamic field's pattern are configured with the dynamic field's indexing options.

Regular field names begin with a letter and can contain the following characters: `a-z` (lowercase), `0-9`, and `_` (underscore). Dynamic field names must begin or end with a wildcard (`*`). The wildcard can also be the only character in a dynamic field name. Multiple wildcards, and wildcards embedded within a string are not supported.

The name `score` is reserved and cannot be used as a field name. To reference a document's ID, you can use the name `_id`.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

- **LatLonOptions** (*dict*) –

Options for a latlon field. A latlon field contains a location stored as a latitude and longitude value pair. Present if `IndexFieldType` specifies the field is of type `latlon`. All options are enabled by default.

- **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.

- **SourceField** (*string*) –

A string that represents the name of an index field. CloudSearch supports regular index fields as well as dynamic fields. A dynamic field's name defines a pattern that begins or ends

with a wildcard. Any document fields that don't map to a regular index field but do match a dynamic field's pattern are configured with the dynamic field's indexing options.

Regular field names begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and `_` (underscore). Dynamic field names must begin or end with a wildcard (`*`). The wildcard can also be the only character in a dynamic field name. Multiple wildcards, and wildcards embedded within a string are not supported.

The name `score` is reserved and cannot be used as a field name. To reference a document's ID, you can use the name `_id`.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

- **IntArrayOptions** (*dict*) –

Options for a field that contains an array of 64-bit signed integers. Present if `IndexFieldType` specifies the field is of type `int-array`. All options are enabled by default.

- **DefaultValue** (*integer*) – A value to use for the field if the field isn't specified for a document.

- **SourceFields** (*string*) –

A list of source fields to map to the field.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **DoubleArrayOptions** (*dict*) –

Options for a field that contains an array of double-precision 64-bit floating point values. Present if `IndexFieldType` specifies the field is of type `double-array`. All options are enabled by default.

- **DefaultValue** (*float*) – A value to use for the field if the field isn't specified for a document.

- **SourceFields** (*string*) –

A list of source fields to map to the field.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –
Whether the contents of the field are searchable.
- **ReturnEnabled** (*boolean*) –
Whether the contents of the field can be returned in the search results.
- **LiteralArrayOptions** (*dict*) –
Options for a field that contains an array of literal strings. Present if `IndexFieldType` specifies the field is of type `literal-array`. All options are enabled by default.
- **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.
- **SourceFields** (*string*) –
A list of source fields to map to the field.
- **FacetEnabled** (*boolean*) –
Whether facet information can be returned for the field.
- **SearchEnabled** (*boolean*) –
Whether the contents of the field are searchable.
- **ReturnEnabled** (*boolean*) –
Whether the contents of the field can be returned in the search results.
- **TextArrayOptions** (*dict*) –
Options for a field that contains an array of text strings. Present if `IndexFieldType` specifies the field is of type `text-array`. A `text-array` field is always searchable. All options are enabled by default.
- **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.
- **SourceFields** (*string*) –
A list of source fields to map to the field.
- **ReturnEnabled** (*boolean*) –
Whether the contents of the field can be returned in the search results.
- **HighlightEnabled** (*boolean*) –
Whether highlights can be returned for the field.
- **AnalysisScheme** (*string*) –
The name of an analysis scheme for a `text-array` field.
- **DateArrayOptions** (*dict*) –
Options for a field that contains an array of dates. Present if `IndexFieldType` specifies the field is of type `date-array`. All options are enabled by default.
- **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.
- **SourceFields** (*string*) –
A list of source fields to map to the field.
- **FacetEnabled** (*boolean*) –
Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –
Whether the contents of the field are searchable.
- **ReturnEnabled** (*boolean*) –
Whether the contents of the field can be returned in the search results.
- * **Status** (*dict*) –
The status of domain configuration option.
 - **CreationDate** (*datetime*) –
A timestamp for when this option was created.
 - **UpdateDate** (*datetime*) –
A timestamp for when this option was last updated.
 - **UpdateVersion** (*integer*) –
A unique integer that indicates when this option was last updated.
 - **State** (*string*) –
The state of processing a change to an option. Possible values:
 - **RequiresIndexDocuments** : the option's latest value will not be deployed until IndexDocuments has been called and indexing is complete.
 - **Processing** : the option's latest value is in the process of being activated.
 - **Active** : the option's latest value is completely deployed.
 - **FailedToValidate** : the option value is not compatible with the domain's data and cannot be used to index the data. You must either modify the option value or update or remove the incompatible documents.
 - **PendingDeletion** (*boolean*) –
Indicates that the option will be deleted once processing is complete.

delete_suggester (***kwargs*)

Deletes a suggester. For more information, see [Getting Search Suggestions](#) in the *Amazon CloudSearch Developer Guide*.

Request Syntax

```
response = client.delete_suggester(
    DomainName='string',
    SuggesterName='string'
)
```

Parameters

- **DomainName** (*string*) – [REQUIRED]

A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

- **SuggesterName** (*string*) – [REQUIRED]

Specifies the name of the suggester you want to delete.

Return type dict

Returns

Response Syntax

```
{
  'Suggester': {
    'Options': {
      'SuggesterName': 'string',
      'DocumentSuggesterOptions': {
        'SourceField': 'string',
        'FuzzyMatching': 'none'|'low'|'high',
        'SortExpression': 'string'
      }
    },
    'Status': {
      'CreationDate': datetime(2015, 1, 1),
      'UpdateDate': datetime(2015, 1, 1),
      'UpdateVersion': 123,
      'State': 'RequiresIndexDocuments'|'Processing'|'Active'|'FailedToValidate',
      'PendingDeletion': True|False
    }
  }
}
```

Response Structure

- (*dict*) –

The result of a `DeleteSuggester` request. Contains the status of the deleted suggester.

- **Suggester** (*dict*) –

The status of the suggester being deleted.

- * **Options** (*dict*) –

Configuration information for a search suggester. Each suggester has a unique name and specifies the text field you want to use for suggestions. The following options can be configured for a suggester: `FuzzyMatching`, `SortExpression`.

- **SuggesterName** (*string*) –

Names must begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and `_` (underscore).

- **DocumentSuggesterOptions** (*dict*) –

Options for a search suggester.

- **SourceField** (*string*) –

The name of the index field you want to use for suggestions.

- **FuzzyMatching** (*string*) –

The level of fuzziness allowed when suggesting matches for a string: `none`, `low`, or `high`. With `none`, the specified string is treated as an exact prefix. With `low`, suggestions must differ from the specified string by no more than one character. With `high`, suggestions can differ by up to two characters. The default is `none`.

- **SortExpression** (*string*) –

An expression that computes a score for each suggestion to control how they are sorted. The scores are rounded to the

nearest integer, with a floor of 0 and a ceiling of $2^{31}-1$. A document's relevance score is not computed for suggestions, so sort expressions cannot reference the `_score` value. To sort suggestions using a numeric field or existing expression, simply specify the name of the field or expression. If no expression is configured for the suggester, the suggestions are sorted with the closest matches listed first.

* **Status** (*dict*) –

The status of domain configuration option.

• **CreationDate** (*datetime*) –

A timestamp for when this option was created.

• **UpdateDate** (*datetime*) –

A timestamp for when this option was last updated.

• **UpdateVersion** (*integer*) –

A unique integer that indicates when this option was last updated.

• **State** (*string*) –

The state of processing a change to an option. Possible values:

- **RequiresIndexDocuments** : the option's latest value will not be deployed until IndexDocuments has been called and indexing is complete.
- **Processing** : the option's latest value is in the process of being activated.
- **Active** : the option's latest value is completely deployed.
- **FailedToValidate** : the option value is not compatible with the domain's data and cannot be used to index the data. You must either modify the option value or update or remove the incompatible documents.
- **PendingDeletion** (*boolean*) –

Indicates that the option will be deleted once processing is complete.

describe_analysis_schemes (***kwargs*)

Gets the analysis schemes configured for a domain. An analysis scheme defines language-specific text processing options for a `text` field. Can be limited to specific analysis schemes by name. By default, shows all analysis schemes and includes any pending changes to the configuration. Set the `Deployed` option to `true` to show the active configuration and exclude pending changes. For more information, see [Configuring Analysis Schemes](#) in the *Amazon CloudSearch Developer Guide*.

Request Syntax

```
response = client.describe_analysis_schemes(
    DomainName='string',
    AnalysisSchemeNames=[
        'string',
    ],
    Deployed=True|False
)
```

Parameters

- **DomainName** (*string*) – [REQUIRED]

The name of the domain you want to describe.

- **AnalysisSchemeNames** (*list*) – The analysis schemes you want to describe.
 - (*string*) –
Names must begin with a letter and can contain the following characters:
a-z (lowercase), 0-9, and _ (underscore).
- **Deployed** (*boolean*) – Whether to display the deployed configuration (`true`) or include any pending changes (`false`). Defaults to `false`.

Return type dict

Returns

Response Syntax

```
{
  'AnalysisSchemes': [
    {
      'Options': {
        'AnalysisSchemeName': 'string',
        'AnalysisSchemeLanguage': 'ar'|'bg'|'ca'|'cs'|'da'|'de'|'el'|'en'|
        'AnalysisOptions': {
          'Synonyms': 'string',
          'Stopwords': 'string',
          'StemmingDictionary': 'string',
          'JapaneseTokenizationDictionary': 'string',
          'AlgorithmicStemming': 'none'|'minimal'|'light'|'full'
        }
      },
      'Status': {
        'CreationDate': datetime(2015, 1, 1),
        'UpdateDate': datetime(2015, 1, 1),
        'UpdateVersion': 123,
        'State': 'RequiresIndexDocuments'|'Processing'|'Active'|'FailedToV
        'PendingDeletion': True|False
      }
    },
  ]
}
```

Response Structure

- (*dict*) –
The result of a `DescribeAnalysisSchemes` request. Contains the analysis schemes configured for the domain specified in the request.
 - **AnalysisSchemes** (*list*) –
The analysis scheme descriptions.
 - * (*dict*) –
The status and configuration of an `AnalysisScheme`.
 - **Options** (*dict*) –
Configuration information for an analysis scheme. Each analysis scheme has a unique name and specifies the language of the text to be processed. The following options can be configured for an analysis scheme: `Synonyms`, `Stopwords`, `StemmingDictionary`, `JapaneseTokenizationDictionary` and `AlgorithmicStemming`.
 - **AnalysisSchemeName** (*string*) –

Names must begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and _ (underscore).

- **AnalysisSchemeLanguage** (*string*) –

An [IETF RFC 4646](#) language code or `mul` for multiple languages.

- **AnalysisOptions** (*dict*) –

Synonyms, stopwords, and stemming options for an analysis scheme. Includes tokenization dictionary for Japanese.

- **Synonyms** (*string*) –

A JSON object that defines synonym groups and aliases. A synonym group is an array of arrays, where each sub-array is a group of terms where each term in the group is considered a synonym of every other term in the group. The `aliases` value is an object that contains a collection of `string:value` pairs where the `string` specifies a term and the array of values specifies each of the aliases for that term. An alias is considered a synonym of the specified term, but the term is not considered a synonym of the alias. For more information about specifying synonyms, see [Synonyms](#) in the *Amazon CloudSearch Developer Guide*.

- **Stopwords** (*string*) –

A JSON array of terms to ignore during indexing and searching. For example, `["a", "an", "the", "of"]`. The stopwords dictionary must explicitly list each word you want to ignore. Wildcards and regular expressions are not supported.

- **StemmingDictionary** (*string*) –

A JSON object that contains a collection of `string:value` pairs that each map a term to its stem. For example, `{"term1": "stem1", "term2": "stem2", "term3": "stem3"}`. The stemming dictionary is applied in addition to any algorithmic stemming. This enables you to override the results of the algorithmic stemming to correct specific cases of overstemming or understemming. The maximum size of a stemming dictionary is 500 KB.

- **JapaneseTokenizationDictionary** (*string*) –

A JSON array that contains a collection of terms, tokens, readings and part of speech for Japanese Tokenization. The Japanese tokenization dictionary enables you to override the default tokenization for selected terms. This is only valid for Japanese language fields.

- **AlgorithmicStemming** (*string*) –

The level of algorithmic stemming to perform: `none`, `minimal`, `light`, or `full`. The available levels vary depending on the language. For more information, see [Language Specific Text Processing Settings](#) in the *Amazon CloudSearch Developer Guide*.

- **Status** (*dict*) –

The status of domain configuration option.

- **CreationDate** (*datetime*) –

A timestamp for when this option was created.

- **UpdateDate** (*datetime*) –

A timestamp for when this option was last updated.

- **UpdateVersion** (*integer*) –

A unique integer that indicates when this option was last updated.

- **State** (*string*) –

The state of processing a change to an option. Possible values:

- **RequiresIndexDocuments** : the option's latest value will not be deployed until IndexDocuments has been called and indexing is complete.
- **Processing** : the option's latest value is in the process of being activated.
- **Active** : the option's latest value is completely deployed.
- **FailedToValidate** : the option value is not compatible with the domain's data and cannot be used to index the data. You must either modify the option value or update or remove the incompatible documents.
- **PendingDeletion** (*boolean*) –

Indicates that the option will be deleted once processing is complete.

describe_availability_options (**kwargs)

Gets the availability options configured for a domain. By default, shows the configuration with any pending changes. Set the `Deployed` option to `true` to show the active configuration and exclude pending changes. For more information, see [Configuring Availability Options](#) in the *Amazon CloudSearch Developer Guide*.

Request Syntax

```
response = client.describe_availability_options(
    DomainName='string',
    Deployed=True|False
)
```

Parameters

- **DomainName** (*string*) – [REQUIRED]

The name of the domain you want to describe.

- **Deployed** (*boolean*) – Whether to display the deployed configuration (`true`) or include any pending changes (`false`). Defaults to `false`.

Return type dict

Returns

Response Syntax

```
{
  'AvailabilityOptions': {
    'Options': True|False,
    'Status': {
      'CreationDate': datetime(2015, 1, 1),
      'UpdateDate': datetime(2015, 1, 1),
      'UpdateVersion': 123,
      'State': 'RequiresIndexDocuments'|'Processing'|'Active'|'FailedToValid
      'PendingDeletion': True|False
    }
  }
}
```

```
    }
  }
}
```

Response Structure

- (*dict*) –

The result of a `DescribeAvailabilityOptions` request. Indicates whether or not the Multi-AZ option is enabled for the domain specified in the request.

- **AvailabilityOptions** (*dict*) –

The availability options configured for the domain. Indicates whether Multi-AZ is enabled for the domain.

- * **Options** (*boolean*) –

The availability options configured for the domain.

- * **Status** (*dict*) –

The status of domain configuration option.

- **CreationDate** (*datetime*) –

A timestamp for when this option was created.

- **UpdateDate** (*datetime*) –

A timestamp for when this option was last updated.

- **UpdateVersion** (*integer*) –

A unique integer that indicates when this option was last updated.

- **State** (*string*) –

The state of processing a change to an option. Possible values:

- **RequiresIndexDocuments** : the option's latest value will not be deployed until `IndexDocuments` has been called and indexing is complete.
- **Processing** : the option's latest value is in the process of being activated.
- **Active** : the option's latest value is completely deployed.
- **FailedToValidate** : the option value is not compatible with the domain's data and cannot be used to index the data. You must either modify the option value or update or remove the incompatible documents.
- **PendingDeletion** (*boolean*) –

Indicates that the option will be deleted once processing is complete.

describe_domains (***kwargs*)

Gets information about the search domains owned by this account. Can be limited to specific domains. Shows all domains by default. To get the number of searchable documents in a domain, use the `console` or submit a `matchall` request to your domain's search endpoint: `q=matchall&q.parser=structuredamp;size=0`. For more information, see [Getting Information about a Search Domain](#) in the *Amazon CloudSearch Developer Guide*.

Request Syntax

```
response = client.describe_domains(
    DomainNames=[
```

```
        'string',  
    ]  
)
```

Parameters **DomainNames** (*list*) – The names of the domains you want to include in the response.

- (*string*) –

A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

Return type dict

Returns

Response Syntax

```
{  
    'DomainStatusList': [  
        {  
            'DomainId': 'string',  
            'DomainName': 'string',  
            'ARN': 'string',  
            'Created': True|False,  
            'Deleted': True|False,  
            'DocService': {  
                'Endpoint': 'string'  
            },  
            'SearchService': {  
                'Endpoint': 'string'  
            },  
            'RequiresIndexDocuments': True|False,  
            'Processing': True|False,  
            'SearchInstanceType': 'string',  
            'SearchPartitionCount': 123,  
            'SearchInstanceCount': 123,  
            'Limits': {  
                'MaximumReplicationCount': 123,  
                'MaximumPartitionCount': 123  
            }  
        },  
    ],  
}
```

Response Structure

- (*dict*) –

The result of a `DescribeDomains` request. Contains the status of the domains specified in the request or all domains owned by the account.

- **DomainStatusList** (*list*) –

A list that contains the status of each requested domain.

- * (*dict*) –

The current status of the search domain.

- **DomainId** (*string*) –

An internally generated unique identifier for a domain.

- **DomainName** (*string*) –

A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

- **ARN** (*string*) –

The Amazon Resource Name (ARN) of the search domain. See [Identifiers for IAM Entities](#) in *Using AWS Identity and Access Management* for more information.

- **Created** (*boolean*) –

True if the search domain is created. It can take several minutes to initialize a domain when CreateDomain is called. Newly created search domains are returned from DescribeDomains with a false value for Created until domain creation is complete.

- **Deleted** (*boolean*) –

True if the search domain has been deleted. The system must clean up resources dedicated to the search domain when DeleteDomain is called. Newly deleted search domains are returned from DescribeDomains with a true value for IsDeleted for several minutes until resource cleanup is complete.

- **DocService** (*dict*) –

The service endpoint for updating documents in a search domain.

- **Endpoint** (*string*) –

The endpoint to which service requests can be submitted. For example, `search-imdb-movies-oopcnjfn6ugofer3zx5iadxxca.eu-west-1.amazonaws.com` or `doc-imdb-movies-oopcnjfn6ugofer3zx5iadxxca.eu-west-1.amazonaws.com`.

- **SearchService** (*dict*) –

The service endpoint for requesting search results from a search domain.

- **Endpoint** (*string*) –

The endpoint to which service requests can be submitted. For example, `search-imdb-movies-oopcnjfn6ugofer3zx5iadxxca.eu-west-1.amazonaws.com` or `doc-imdb-movies-oopcnjfn6ugofer3zx5iadxxca.eu-west-1.amazonaws.com`.

- **RequiresIndexDocuments** (*boolean*) –

True if IndexDocuments needs to be called to activate the current domain configuration.

- **Processing** (*boolean*) –

True if processing is being done to activate the current domain configuration.

- **SearchInstanceType** (*string*) –

The instance type that is being used to process search requests.

- **SearchPartitionCount** (*integer*) –
The number of partitions across which the search index is spread.
- **SearchInstanceCount** (*integer*) –
The number of search instances that are available to process search requests.
- **Limits** (*dict*) –
- **MaximumReplicationCount** (*integer*) –
- **MaximumPartitionCount** (*integer*) –

describe_expressions (***kwargs*)

Gets the expressions configured for the search domain. Can be limited to specific expressions by name. By default, shows all expressions and includes any pending changes to the configuration. Set the `Deployed` option to `true` to show the active configuration and exclude pending changes. For more information, see [Configuring Expressions](#) in the *Amazon CloudSearch Developer Guide* .

Request Syntax

```
response = client.describe_expressions(  
    DomainName='string',  
    ExpressionNames=[  
        'string',  
    ],  
    Deployed=True|False  
)
```

Parameters

- **DomainName** (*string*) – **[REQUIRED]**
The name of the domain you want to describe.
- **ExpressionNames** (*list*) – Limits the “DescribeExpressions” response to the specified expressions. If not specified, all expressions are shown.
 - (*string*) –
Names must begin with a letter and can contain the following characters:
a-z (lowercase), 0-9, and _ (underscore).
- **Deployed** (*boolean*) – Whether to display the deployed configuration (`true`) or include any pending changes (`false`). Defaults to `false` .

Return type dict

Returns**Response Syntax**

```
{  
    'Expressions': [  
        {  
            'Options': {  
                'ExpressionName': 'string',  
                'ExpressionValue': 'string'  
            },  
            'Status': {  
                'CreationDate': datetime(2015, 1, 1),  
                'UpdateDate': datetime(2015, 1, 1),  
                'UpdateVersion': 123,  
                'State': 'RequiresIndexDocuments'|'Processing'|'Active'|'FailedToV'  
                'PendingDeletion': True|False  
            }  
        }  
    ]  
}
```

```

    },
  ]
}

```

Response Structure

- *(dict)* –

The result of a `DescribeExpressions` request. Contains the expressions configured for the domain specified in the request.

- **Expressions** (*list*) –

The expressions configured for the domain.

- * *(dict)* –

The value of an `Expression` and its current status.

- **Options** (*dict*) –

The expression that is evaluated for sorting while processing a search request.

- **ExpressionName** (*string*) –

Names must begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and _ (underscore).

- **ExpressionValue** (*string*) –

The expression to evaluate for sorting while processing a search request. The `Expression` syntax is based on JavaScript expressions. For more information, see [Configuring Expressions](#) in the *Amazon CloudSearch Developer Guide*.

- **Status** (*dict*) –

The status of domain configuration option.

- **CreationDate** (*datetime*) –

A timestamp for when this option was created.

- **UpdateDate** (*datetime*) –

A timestamp for when this option was last updated.

- **UpdateVersion** (*integer*) –

A unique integer that indicates when this option was last updated.

- **State** (*string*) –

The state of processing a change to an option. Possible values:

- `RequiresIndexDocuments`: the option's latest value will not be deployed until `IndexDocuments` has been called and indexing is complete.
- `Processing`: the option's latest value is in the process of being activated.
- `Active`: the option's latest value is completely deployed.
- `FailedToValidate`: the option value is not compatible with the domain's data and cannot be used to index the data. You must either modify the option value or update or remove the incompatible documents.
- **PendingDeletion** (*boolean*) –

Indicates that the option will be deleted once processing is complete.

describe_index_fields (**kwargs)

Gets information about the index fields configured for the search domain. Can be limited to specific fields by name. By default, shows all fields and includes any pending changes to the configuration. Set the `Deployed` option to `true` to show the active configuration and exclude pending changes. For more information, see [Getting Domain Information](#) in the *Amazon CloudSearch Developer Guide*.

Request Syntax

```
response = client.describe_index_fields(
    DomainName='string',
    FieldNames=[
        'string',
    ],
    Deployed=True|False
)
```

Parameters

- **DomainName** (*string*) – **[REQUIRED]**
The name of the domain you want to describe.
- **FieldNames** (*list*) – A list of the index fields you want to describe. If not specified, information is returned for all configured index fields.
 - (*string*) –
- **Deployed** (*boolean*) – Whether to display the deployed configuration (`true`) or include any pending changes (`false`). Defaults to `false`.

Return type dict

Returns**Response Syntax**

```
{
    'IndexFields': [
        {
            'Options': {
                'IndexFieldName': 'string',
                'IndexFieldType': 'int'|'double'|'literal'|'text'|'date'|'latlon'|
                'IntOptions': {
                    'DefaultValue': 123,
                    'SourceField': 'string',
                    'FacetEnabled': True|False,
                    'SearchEnabled': True|False,
                    'ReturnEnabled': True|False,
                    'SortEnabled': True|False
                },
                'DoubleOptions': {
                    'DefaultValue': 123.0,
                    'SourceField': 'string',
                    'FacetEnabled': True|False,
                    'SearchEnabled': True|False,
                    'ReturnEnabled': True|False,
                    'SortEnabled': True|False
                },
                'LiteralOptions': {
                    'DefaultValue': 'string',
                    'SourceField': 'string',
                    'FacetEnabled': True|False,
                    'SearchEnabled': True|False,
                    'ReturnEnabled': True|False,

```



```

        'SortEnabled': True|False
    },
    'TextOptions': {
        'DefaultValue': 'string',
        'SourceField': 'string',
        'ReturnEnabled': True|False,
        'SortEnabled': True|False,
        'HighlightEnabled': True|False,
        'AnalysisScheme': 'string'
    },
    'DateOptions': {
        'DefaultValue': 'string',
        'SourceField': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False,
        'SortEnabled': True|False
    },
    'LatLonOptions': {
        'DefaultValue': 'string',
        'SourceField': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False,
        'SortEnabled': True|False
    },
    'IntArrayOptions': {
        'DefaultValue': 123,
        'SourceFields': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False
    },
    'DoubleArrayOptions': {
        'DefaultValue': 123.0,
        'SourceFields': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False
    },
    'LiteralArrayOptions': {
        'DefaultValue': 'string',
        'SourceFields': 'string',
        'FacetEnabled': True|False,
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False
    },
    'TextArrayOptions': {
        'DefaultValue': 'string',
        'SourceFields': 'string',
        'ReturnEnabled': True|False,
        'HighlightEnabled': True|False,
        'AnalysisScheme': 'string'
    },
    'DateArrayOptions': {
        'DefaultValue': 'string',
        'SourceFields': 'string',
        'FacetEnabled': True|False,

```

```
        'SearchEnabled': True|False,
        'ReturnEnabled': True|False
    },
    },
    'Status': {
        'CreationDate': datetime(2015, 1, 1),
        'UpdateDate': datetime(2015, 1, 1),
        'UpdateVersion': 123,
        'State': 'RequiresIndexDocuments'|'Processing'|'Active'|'FailedToV
        'PendingDeletion': True|False
    }
    },
    ]
}
```

Response Structure

- (*dict*) –

The result of a `DescribeIndexFields` request. Contains the index fields configured for the domain specified in the request.

- **IndexFields** (*list*) –

The index fields configured for the domain.

- * (*dict*) –

The value of an `IndexField` and its current status.

- **Options** (*dict*) –

Configuration information for a field in the index, including its name, type, and options. The supported options depend on the “`IndexFieldType`”.

- **IndexFieldName** (*string*) –

A string that represents the name of an index field. CloudSearch supports regular index fields as well as dynamic fields. A dynamic field’s name defines a pattern that begins or ends with a wildcard. Any document fields that don’t map to a regular index field but do match a dynamic field’s pattern are configured with the dynamic field’s indexing options.

Regular field names begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and `_` (underscore). Dynamic field names must begin or end with a wildcard (`*`). The wildcard can also be the only character in a dynamic field name. Multiple wildcards, and wildcards embedded within a string are not supported.

The name `score` is reserved and cannot be used as a field name. To reference a document’s ID, you can use the name `_id`.

- **IndexFieldType** (*string*) –

The type of field. The valid options for a field depend on the field type. For more information about the supported field types, see [Configuring Index Fields](#) in the *Amazon CloudSearch Developer Guide*.

- **IntOptions** (*dict*) –

Options for a 64-bit signed integer field. Present if

`IndexFieldType` specifies the field is of type `int`. All options are enabled by default.

- **DefaultValue** (*integer*) – A value to use for the field if the field isn't specified for a document. This can be important if you are using the field in an expression and that field is not present in every document.

- **SourceField** (*string*) –

The name of the source field to map to the field.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

- **DoubleOptions** (*dict*) –

Options for a double-precision 64-bit floating point field. Present if `IndexFieldType` specifies the field is of type `double`. All options are enabled by default.

- **DefaultValue** (*float*) –

A value to use for the field if the field isn't specified for a document. This can be important if you are using the field in an expression and that field is not present in every document.

- **SourceField** (*string*) –

The name of the source field to map to the field.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

- **LiteralOptions** (*dict*) –

Options for literal field. Present if `IndexFieldType` specifies the field is of type `literal`. All options are enabled by default.

- **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.

- **SourceField** (*string*) –

A string that represents the name of an index field. CloudSearch supports regular index fields as well as dynamic fields. A dynamic field's name defines a pattern that begins or ends with a wildcard. Any document fields that don't map to a

regular index field but do match a dynamic field's pattern are configured with the dynamic field's indexing options.

Regular field names begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and `_` (underscore). Dynamic field names must begin or end with a wildcard (`*`). The wildcard can also be the only character in a dynamic field name. Multiple wildcards, and wildcards embedded within a string are not supported.

The name `score` is reserved and cannot be used as a field name. To reference a document's ID, you can use the name `_id`.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

- **TextOptions** (*dict*) –

Options for text field. Present if `IndexFieldType` specifies the field is of type `text`. A `text` field is always searchable. All options are enabled by default.

- **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.
- **SourceField** (*string*) –

A string that represents the name of an index field. CloudSearch supports regular index fields as well as dynamic fields. A dynamic field's name defines a pattern that begins or ends with a wildcard. Any document fields that don't map to a regular index field but do match a dynamic field's pattern are configured with the dynamic field's indexing options.

Regular field names begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and `_` (underscore). Dynamic field names must begin or end with a wildcard (`*`). The wildcard can also be the only character in a dynamic field name. Multiple wildcards, and wildcards embedded within a string are not supported.

The name `score` is reserved and cannot be used as a field name. To reference a document's ID, you can use the name `_id`.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

- **HighlightEnabled** (*boolean*) –

Whether highlights can be returned for the field.

- **AnalysisScheme** (*string*) –

The name of an analysis scheme for a `text` field.

- **DateOptions** (*dict*) –

Options for a date field. Dates and times are specified in UTC (Coordinated Universal Time) according to IETF RFC3339: `yyyy-mm-ddT00:00:00Z`. Present if `IndexFieldType` specifies the field is of type `date`. All options are enabled by default.

- **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.

- **SourceField** (*string*) –

A string that represents the name of an index field. CloudSearch supports regular index fields as well as dynamic fields. A dynamic field's name defines a pattern that begins or ends with a wildcard. Any document fields that don't map to a regular index field but do match a dynamic field's pattern are configured with the dynamic field's indexing options.

Regular field names begin with a letter and can contain the following characters: `a-z` (lowercase), `0-9`, and `_` (underscore). Dynamic field names must begin or end with a wildcard (`*`). The wildcard can also be the only character in a dynamic field name. Multiple wildcards, and wildcards embedded within a string are not supported.

The name `score` is reserved and cannot be used as a field name. To reference a document's ID, you can use the name `_id`.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

- **LatLonOptions** (*dict*) –

Options for a latlon field. A latlon field contains a location stored as a latitude and longitude value pair. Present if `IndexFieldType` specifies the field is of type `latlon`. All options are enabled by default.

- **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.

- **SourceField** (*string*) –

A string that represents the name of an index field. CloudSearch supports regular index fields as well as dynamic fields. A dynamic field's name defines a pattern that begins or ends

with a wildcard. Any document fields that don't map to a regular index field but do match a dynamic field's pattern are configured with the dynamic field's indexing options.

Regular field names begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and `_` (underscore). Dynamic field names must begin or end with a wildcard (`*`). The wildcard can also be the only character in a dynamic field name. Multiple wildcards, and wildcards embedded within a string are not supported.

The name `score` is reserved and cannot be used as a field name. To reference a document's ID, you can use the name `_id`.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **SortEnabled** (*boolean*) –

Whether the field can be used to sort the search results.

- **IntArrayOptions** (*dict*) –

Options for a field that contains an array of 64-bit signed integers. Present if `IndexFieldType` specifies the field is of type `int-array`. All options are enabled by default.

- **DefaultValue** (*integer*) – A value to use for the field if the field isn't specified for a document.

- **SourceFields** (*string*) –

A list of source fields to map to the field.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –

Whether the contents of the field are searchable.

- **ReturnEnabled** (*boolean*) –

Whether the contents of the field can be returned in the search results.

- **DoubleArrayOptions** (*dict*) –

Options for a field that contains an array of double-precision 64-bit floating point values. Present if `IndexFieldType` specifies the field is of type `double-array`. All options are enabled by default.

- **DefaultValue** (*float*) – A value to use for the field if the field isn't specified for a document.

- **SourceFields** (*string*) –

A list of source fields to map to the field.

- **FacetEnabled** (*boolean*) –

Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –
Whether the contents of the field are searchable.
- **ReturnEnabled** (*boolean*) –
Whether the contents of the field can be returned in the search results.
- **LiteralArrayOptions** (*dict*) –
Options for a field that contains an array of literal strings. Present if `IndexFieldType` specifies the field is of type `literal-array`. All options are enabled by default.
- **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.
- **SourceFields** (*string*) –
A list of source fields to map to the field.
- **FacetEnabled** (*boolean*) –
Whether facet information can be returned for the field.
- **SearchEnabled** (*boolean*) –
Whether the contents of the field are searchable.
- **ReturnEnabled** (*boolean*) –
Whether the contents of the field can be returned in the search results.
- **TextArrayOptions** (*dict*) –
Options for a field that contains an array of text strings. Present if `IndexFieldType` specifies the field is of type `text-array`. A `text-array` field is always searchable. All options are enabled by default.
- **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.
- **SourceFields** (*string*) –
A list of source fields to map to the field.
- **ReturnEnabled** (*boolean*) –
Whether the contents of the field can be returned in the search results.
- **HighlightEnabled** (*boolean*) –
Whether highlights can be returned for the field.
- **AnalysisScheme** (*string*) –
The name of an analysis scheme for a `text-array` field.
- **DateArrayOptions** (*dict*) –
Options for a field that contains an array of dates. Present if `IndexFieldType` specifies the field is of type `date-array`. All options are enabled by default.
- **DefaultValue** (*string*) – A value to use for the field if the field isn't specified for a document.
- **SourceFields** (*string*) –
A list of source fields to map to the field.
- **FacetEnabled** (*boolean*) –
Whether facet information can be returned for the field.

- **SearchEnabled** (*boolean*) –
Whether the contents of the field are searchable.
- **ReturnEnabled** (*boolean*) –
Whether the contents of the field can be returned in the search results.
- **Status** (*dict*) –
The status of domain configuration option.
- **CreationDate** (*datetime*) –
A timestamp for when this option was created.
- **UpdateDate** (*datetime*) –
A timestamp for when this option was last updated.
- **UpdateVersion** (*integer*) –
A unique integer that indicates when this option was last updated.
- **State** (*string*) –
The state of processing a change to an option. Possible values:
 - `RequiresIndexDocuments` : the option's latest value will not be deployed until `IndexDocuments` has been called and indexing is complete.
 - `Processing` : the option's latest value is in the process of being activated.
 - `Active` : the option's latest value is completely deployed.
 - `FailedToValidate` : the option value is not compatible with the domain's data and cannot be used to index the data. You must either modify the option value or update or remove the incompatible documents.
- **PendingDeletion** (*boolean*) –
Indicates that the option will be deleted once processing is complete.

describe_scaling_parameters (***kwargs*)

Gets the scaling parameters configured for a domain. A domain's scaling parameters specify the desired search instance type and replication count. For more information, see [Configuring Scaling Options](#) in the *Amazon CloudSearch Developer Guide*.

Request Syntax

```
response = client.describe_scaling_parameters(  
    DomainName='string'  
)
```

Parameters `DomainName` (*string*) – **[REQUIRED]**

A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

Return type dict

Returns**Response Syntax**


```
{
  'ScalingParameters': {
    'Options': {
      'DesiredInstanceType': 'search.m1.small'|'search.m1.large'|'search.m2.xlarge',
      'DesiredReplicationCount': 123,
      'DesiredPartitionCount': 123
    },
    'Status': {
      'CreationDate': datetime(2015, 1, 1),
      'UpdateDate': datetime(2015, 1, 1),
      'UpdateVersion': 123,
      'State': 'RequiresIndexDocuments'|'Processing'|'Active'|'FailedToValidate'|'PendingDeletion',
      'PendingDeletion': True|False
    }
  }
}
```

Response Structure

- (*dict*) –

The result of a `DescribeScalingParameters` request. Contains the scaling parameters configured for the domain specified in the request.

- **ScalingParameters** (*dict*) –

The status and configuration of a search domain's scaling parameters.

- * **Options** (*dict*) –

The desired instance type and desired number of replicas of each index partition.

- **DesiredInstanceType** (*string*) –

The instance type that you want to preconfigure for your domain. For example, `search.m1.small`.

- **DesiredReplicationCount** (*integer*) –

The number of replicas you want to preconfigure for each index partition.

- **DesiredPartitionCount** (*integer*) –

The number of partitions you want to preconfigure for your domain. Only valid when you select `m2.xlarge` as the desired instance type.

- * **Status** (*dict*) –

The status of domain configuration option.

- **CreationDate** (*datetime*) –

A timestamp for when this option was created.

- **UpdateDate** (*datetime*) –

A timestamp for when this option was last updated.

- **UpdateVersion** (*integer*) –

A unique integer that indicates when this option was last updated.

- **State** (*string*) –

The state of processing a change to an option. Possible values:

- `RequiresIndexDocuments` : the option's latest value will not be deployed until `IndexDocuments` has been called and indexing is complete.
- `Processing` : the option's latest value is in the process of being activated.
- `Active` : the option's latest value is completely deployed.
- `FailedToValidate` : the option value is not compatible with the domain's data and cannot be used to index the data. You must either modify the option value or update or remove the incompatible documents.
- **PendingDeletion** (*boolean*) –

Indicates that the option will be deleted once processing is complete.

describe_service_access_policies (***kwargs*)

Gets information about the access policies that control access to the domain's document and search endpoints. By default, shows the configuration with any pending changes. Set the `Deployed` option to `true` to show the active configuration and exclude pending changes. For more information, see [Configuring Access for a Search Domain](#) in the *Amazon CloudSearch Developer Guide*.

Request Syntax

```
response = client.describe_service_access_policies(
    DomainName='string',
    Deployed=True|False
)
```

Parameters

- **DomainName** (*string*) – **[REQUIRED]**
The name of the domain you want to describe.
- **Deployed** (*boolean*) – Whether to display the deployed configuration (`true`) or include any pending changes (`false`). Defaults to `false`.

Return type dict

Returns

Response Syntax

```
{
  'AccessPolicies': {
    'Options': 'string',
    'Status': {
      'CreationDate': datetime(2015, 1, 1),
      'UpdateDate': datetime(2015, 1, 1),
      'UpdateVersion': 123,
      'State': 'RequiresIndexDocuments'|'Processing'|'Active'|'FailedToValid
      'PendingDeletion': True|False
    }
  }
}
```

Response Structure

- (*dict*) –

The result of a `DescribeServiceAccessPolicies` request.

- **AccessPolicies** (*dict*) –

The access rules configured for the domain specified in the request.

* **Options** (*string*) –

Access rules for a domain's document or search service endpoints. For more information, see [Configuring Access for a Search Domain](#) in the *Amazon CloudSearch Developer Guide* . The maximum size of a policy document is 100 KB.

* **Status** (*dict*) –

The status of domain configuration option.

· **CreationDate** (*datetime*) –

A timestamp for when this option was created.

· **UpdateDate** (*datetime*) –

A timestamp for when this option was last updated.

· **UpdateVersion** (*integer*) –

A unique integer that indicates when this option was last updated.

· **State** (*string*) –

The state of processing a change to an option. Possible values:

- **RequiresIndexDocuments** : the option's latest value will not be deployed until IndexDocuments has been called and indexing is complete.
- **Processing** : the option's latest value is in the process of being activated.
- **Active** : the option's latest value is completely deployed.
- **FailedToValidate** : the option value is not compatible with the domain's data and cannot be used to index the data. You must either modify the option value or update or remove the incompatible documents.
- **PendingDeletion** (*boolean*) –

Indicates that the option will be deleted once processing is complete.

describe_suggesters (***kwargs*)

Gets the suggesters configured for a domain. A suggester enables you to display possible matches before users finish typing their queries. Can be limited to specific suggesters by name. By default, shows all suggesters and includes any pending changes to the configuration. Set the `Deployed` option to `true` to show the active configuration and exclude pending changes. For more information, see [Getting Search Suggestions](#) in the *Amazon CloudSearch Developer Guide* .

Request Syntax

```
response = client.describe_suggesters(
    DomainName='string',
    SuggesterNames=[
        'string',
    ],
    Deployed=True|False
)
```

Parameters

- **DomainName** (*string*) – [REQUIRED]

The name of the domain you want to describe.

- **SuggesterNames** (*list*) – The suggesters you want to describe.

– (*string*) –

Names must begin with a letter and can contain the following characters:
a-z (lowercase), 0-9, and _ (underscore).

- **Deployed** (*boolean*) – Whether to display the deployed configuration (`true`) or include any pending changes (`false`). Defaults to `false`.

Return type dict

Returns

Response Syntax

```
{
  'Suggesters': [
    {
      'Options': {
        'SuggesterName': 'string',
        'DocumentSuggesterOptions': {
          'SourceField': 'string',
          'FuzzyMatching': 'none'|'low'|'high',
          'SortExpression': 'string'
        }
      },
      'Status': {
        'CreationDate': datetime(2015, 1, 1),
        'UpdateDate': datetime(2015, 1, 1),
        'UpdateVersion': 123,
        'State': 'RequiresIndexDocuments'|'Processing'|'Active'|'FailedToV
        'PendingDeletion': True|False
      }
    },
  ]
}
```

Response Structure

- (*dict*) –

The result of a `DescribeSuggesters` request.

– **Suggesters** (*list*) –

The suggesters configured for the domain specified in the request.

* (*dict*) –

The value of a `Suggester` and its current status.

- **Options** (*dict*) –

Configuration information for a search suggester. Each suggester has a unique name and specifies the text field you want to use for suggestions. The following options can be configured for a suggester: `FuzzyMatching`, `SortExpression`.

- **SuggesterName** (*string*) –

Names must begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and _ (underscore).

- **DocumentSuggesterOptions** (*dict*) –

Options for a search suggester.

- **SourceField** (*string*) –

The name of the index field you want to use for suggestions.

- **FuzzyMatching** (*string*) –

The level of fuzziness allowed when suggesting matches for a string: `none`, `low`, or `high`. With `none`, the specified string is treated as an exact prefix. With `low`, suggestions must differ from the specified string by no more than one character. With `high`, suggestions can differ by up to two characters. The default is `none`.

- **SortExpression** (*string*) –

An expression that computes a score for each suggestion to control how they are sorted. The scores are rounded to the nearest integer, with a floor of 0 and a ceiling of $2^{31}-1$. A document's relevance score is not computed for suggestions, so sort expressions cannot reference the `_score` value. To sort suggestions using a numeric field or existing expression, simply specify the name of the field or expression. If no expression is configured for the suggester, the suggestions are sorted with the closest matches listed first.

- **Status** (*dict*) –

The status of domain configuration option.

- **CreationDate** (*datetime*) –

A timestamp for when this option was created.

- **UpdateDate** (*datetime*) –

A timestamp for when this option was last updated.

- **UpdateVersion** (*integer*) –

A unique integer that indicates when this option was last updated.

- **State** (*string*) –

The state of processing a change to an option. Possible values:

- `RequiresIndexDocuments`: the option's latest value will not be deployed until `IndexDocuments` has been called and indexing is complete.
- `Processing`: the option's latest value is in the process of being activated.
- `Active`: the option's latest value is completely deployed.
- `FailedToValidate`: the option value is not compatible with the domain's data and cannot be used to index the data. You must either modify the option value or update or remove the incompatible documents.
- **PendingDeletion** (*boolean*) –

Indicates that the option will be deleted once processing is complete.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)

- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_waiter (*waiter_name*)

index_documents (***kwargs*)

Tells the search domain to start indexing its documents using the latest indexing options. This operation must be invoked to activate options whose `OptionStatus` is `RequiresIndexDocuments`.

Request Syntax

```
response = client.index_documents(  
    DomainName='string'  
)
```

Parameters **DomainName** (*string*) – [REQUIRED]

A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

Return type `dict`

Returns

Response Syntax

```
{  
    'FieldNames': [  
        'string',  
    ]  
}
```

Response Structure

- (*dict*) –

The result of an `IndexDocuments` request. Contains the status of the indexing operation, including the fields being indexed.

- **FieldNames** (*list*) –

The names of the fields that are currently being indexed.

- * (*string*) –

A string that represents the name of an index field. CloudSearch supports regular index fields as well as dynamic fields. A dynamic field's name defines a pattern that begins or ends with a wildcard. Any document fields that don't map to a regular index field but do

match a dynamic field's pattern are configured with the dynamic field's indexing options.

Regular field names begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and _ (underscore). Dynamic field names must begin or end with a wildcard (*). The wildcard can also be the only character in a dynamic field name. Multiple wildcards, and wildcards embedded within a string are not supported.

The name `score` is reserved and cannot be used as a field name. To reference a document's ID, you can use the name `_id`.

list_domain_names()

Lists all search domains owned by an account.

Request Syntax

```
response = client.list_domain_names()
```

Return type dict

Returns

Response Syntax

```
{
    'DomainNames': {
        'string': 'string'
    }
}
```

Response Structure

- (*dict*) –

The result of a `ListDomainNames` request. Contains a list of the domains owned by an account.

- **DomainNames** (*dict*) –

The names of the search domains owned by an account.

- * (*string*) –

A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

- (*string*) –

The Amazon CloudSearch API version for a domain: 2011-02-01 or 2013-01-01.

update_availability_options(kwargs)**

Configures the availability options for a domain. Enabling the Multi-AZ option expands an Amazon CloudSearch domain to an additional Availability Zone in the same Region to increase fault tolerance in the event of a service disruption. Changes to the Multi-AZ option can take about half an hour to become active. For more information, see [Configuring Availability Options](#) in the *Amazon CloudSearch Developer Guide*.

Request Syntax

```
response = client.update_availability_options(  
    DomainName='string',  
    MultiAZ=True|False  
)
```

Parameters

- **DomainName** (*string*) – [REQUIRED]

A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

- **MultiAZ** (*boolean*) – [REQUIRED]

You expand an existing search domain to a second Availability Zone by setting the Multi-AZ option to true. Similarly, you can turn off the Multi-AZ option to downgrade the domain to a single Availability Zone by setting the Multi-AZ option to false.

Return type dict

Returns

Response Syntax

```
{  
    'AvailabilityOptions': {  
        'Options': True|False,  
        'Status': {  
            'CreationDate': datetime(2015, 1, 1),  
            'UpdateDate': datetime(2015, 1, 1),  
            'UpdateVersion': 123,  
            'State': 'RequiresIndexDocuments'|'Processing'|'Active'|'FailedToValid  
            'PendingDeletion': True|False  
        }  
    }  
}
```

Response Structure

- (*dict*) –

The result of a `UpdateAvailabilityOptions` request. Contains the status of the domain's availability options.

- **AvailabilityOptions** (*dict*) –

The newly-configured availability options. Indicates whether Multi-AZ is enabled for the domain.

- * **Options** (*boolean*) –

The availability options configured for the domain.

- * **Status** (*dict*) –

The status of domain configuration option.

- **CreationDate** (*datetime*) –

A timestamp for when this option was created.

- **UpdateDate** (*datetime*) –

A timestamp for when this option was last updated.

- **UpdateVersion** (*integer*) –

A unique integer that indicates when this option was last updated.

- **State** (*string*) –

The state of processing a change to an option. Possible values:

- **RequiresIndexDocuments** : the option's latest value will not be deployed until IndexDocuments has been called and indexing is complete.
- **Processing** : the option's latest value is in the process of being activated.
- **Active** : the option's latest value is completely deployed.
- **FailedToValidate** : the option value is not compatible with the domain's data and cannot be used to index the data. You must either modify the option value or update or remove the incompatible documents.
- **PendingDeletion** (*boolean*) –

Indicates that the option will be deleted once processing is complete.

update_scaling_parameters (**kwargs)

Configures scaling parameters for a domain. A domain's scaling parameters specify the desired search instance type and replication count. Amazon CloudSearch will still automatically scale your domain based on the volume of data and traffic, but not below the desired instance type and replication count. If the Multi-AZ option is enabled, these values control the resources used per Availability Zone. For more information, see [Configuring Scaling Options](#) in the *Amazon CloudSearch Developer Guide*.

Request Syntax

```
response = client.update_scaling_parameters(
    DomainName='string',
    ScalingParameters={
        'DesiredInstanceType': 'search.m1.small'|'search.m1.large'|'search.m2.xlarge'|'search.m2.xlarge',
        'DesiredReplicationCount': 123,
        'DesiredPartitionCount': 123
    }
)
```

Parameters

- **DomainName** (*string*) – **[REQUIRED]**

A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

- **ScalingParameters** (*dict*) – **[REQUIRED]**

The desired instance type and desired number of replicas of each index partition.

- **DesiredInstanceType** (*string*) –

The instance type that you want to preconfigure for your domain. For example, `search.m1.small`.

- **DesiredReplicationCount** (*integer*) –

The number of replicas you want to preconfigure for each index partition.

- **DesiredPartitionCount** (*integer*) –

The number of partitions you want to preconfigure for your domain. Only valid when you select `m2.2xlarge` as the desired instance type.

Return type dict

Returns

Response Syntax

```
{
  'ScalingParameters': {
    'Options': {
      'DesiredInstanceType': 'search.m1.small'|'search.m1.large'|'search.m2.xlarge',
      'DesiredReplicationCount': 123,
      'DesiredPartitionCount': 123
    },
    'Status': {
      'CreationDate': datetime(2015, 1, 1),
      'UpdateDate': datetime(2015, 1, 1),
      'UpdateVersion': 123,
      'State': 'RequiresIndexDocuments'|'Processing'|'Active'|'FailedToValidate'|'PendingDeletion',
      'PendingDeletion': True|False
    }
  }
}
```

Response Structure

- (*dict*) –

The result of a `UpdateScalingParameters` request. Contains the status of the newly-configured scaling parameters.

- **ScalingParameters** (*dict*) –

The status and configuration of a search domain's scaling parameters.

- * **Options** (*dict*) –

The desired instance type and desired number of replicas of each index partition.

- **DesiredInstanceType** (*string*) –

The instance type that you want to preconfigure for your domain. For example, `search.m1.small`.

- **DesiredReplicationCount** (*integer*) –

The number of replicas you want to preconfigure for each index partition.

- **DesiredPartitionCount** (*integer*) –

The number of partitions you want to preconfigure for your domain. Only valid when you select `m2.2xlarge` as the desired instance type.

- * **Status** (*dict*) –

The status of domain configuration option.

- **CreationDate** (*datetime*) –

A timestamp for when this option was created.

- **UpdateDate** (*datetime*) –

A timestamp for when this option was last updated.

- **UpdateVersion** (*integer*) –

A unique integer that indicates when this option was last updated.

- **State** (*string*) –

The state of processing a change to an option. Possible values:

- **RequiresIndexDocuments** : the option's latest value will not be deployed until IndexDocuments has been called and indexing is complete.
- **Processing** : the option's latest value is in the process of being activated.
- **Active** : the option's latest value is completely deployed.
- **FailedToValidate** : the option value is not compatible with the domain's data and cannot be used to index the data. You must either modify the option value or update or remove the incompatible documents.
- **PendingDeletion** (*boolean*) –

Indicates that the option will be deleted once processing is complete.

update_service_access_policies (***kwargs*)

Configures the access rules that control access to the domain's document and search endpoints. For more information, see [Configuring Access for an Amazon CloudSearch Domain](#).

Request Syntax

```
response = client.update_service_access_policies(
    DomainName='string',
    AccessPolicies='string'
)
```

Parameters

- **DomainName** (*string*) – **[REQUIRED]**

A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

- **AccessPolicies** (*string*) – **[REQUIRED]**

The access rules you want to configure. These rules replace any existing rules.

Return type dict

Returns

Response Syntax

```
{
  'AccessPolicies': {
    'Options': 'string',
    'Status': {
      'CreationDate': datetime(2015, 1, 1),
      'UpdateDate': datetime(2015, 1, 1),
      'UpdateVersion': 123,
      'State': 'RequiresIndexDocuments'|'Processing'|'Active'|'FailedToValid
      'PendingDeletion': True|False
    }
  }
}
```

Response Structure

- (*dict*) –

The result of an `UpdateServiceAccessPolicies` request. Contains the new access policies.

- **AccessPolicies** (*dict*) –

The access rules configured for the domain.

- * **Options** (*string*) –

Access rules for a domain's document or search service endpoints. For more information, see [Configuring Access for a Search Domain](#) in the *Amazon CloudSearch Developer Guide* . The maximum size of a policy document is 100 KB.

- * **Status** (*dict*) –

The status of domain configuration option.

- **CreationDate** (*datetime*) –

A timestamp for when this option was created.

- **UpdateDate** (*datetime*) –

A timestamp for when this option was last updated.

- **UpdateVersion** (*integer*) –

A unique integer that indicates when this option was last updated.

- **State** (*string*) –

The state of processing a change to an option. Possible values:

- **RequiresIndexDocuments** : the option's latest value will not be deployed until `IndexDocuments` has been called and indexing is complete.
- **Processing** : the option's latest value is in the process of being activated.
- **Active** : the option's latest value is completely deployed.
- **FailedToValidate** : the option value is not compatible with the domain's data and cannot be used to index the data. You must either modify the option value or update or remove the incompatible documents.
- **PendingDeletion** (*boolean*) –

Indicates that the option will be deleted once processing is complete.

CloudSearchDomain

Table of Contents

- [CloudSearchDomain](#)
 - [Client](#)

Client

class `CloudSearchDomain.Client`

A low-level client representing Amazon CloudSearch Domain:

```
import boto3
```

```
client = boto3.client('cloudsearchdomain')
```

These are the available methods:

- `can_paginate()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_waiter()`
- `search()`
- `suggest()`
- `upload_documents()`

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to *ClientMethod*.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type L{botocore.paginate.Paginator}

Returns A paginator object.

get_waiter (*waiter_name*)

search (***kwargs*)

Retrieves a list of documents that match the specified search criteria. How you specify the search criteria depends on which query parser you use. Amazon CloudSearch supports four query parsers:

- **simple**: search all text and text-array fields for the specified string. Search for phrases, individual terms, and prefixes.
- **structured**: search specific fields, construct compound queries using Boolean operators, and use advanced features such as term boosting and proximity searching.
- **lucene**: specify search criteria using the Apache Lucene query parser syntax.
- **dismax**: specify search criteria using the simplified subset of the Apache Lucene query parser

syntax defined by the DisMax query parser.

For more information, see [Searching Your Data](#) in the *Amazon CloudSearch Developer Guide*.

The endpoint for submitting `Search` requests is domain-specific. You submit search requests to a domain's search endpoint. To get the search endpoint for your domain, use the Amazon CloudSearch configuration service `DescribeDomains` action. A domain's endpoints are also displayed on the domain dashboard in the Amazon CloudSearch console.

Request Syntax

```
response = client.search(
    cursor='string',
    expr='string',
    facet='string',
    filterQuery='string',
    highlight='string',
    partial=True|False,
    query='string',
    queryOptions='string',
    queryParser='simple'|'structured'|'lucene'|'dismax',
    return='string',
    size=123,
    sort='string',
    start=123
)
```

Parameters

- **cursor** (*string*) – Retrieves a cursor value you can use to page through large result sets. Use the `size` parameter to control the number of hits to include in each response. You can specify either the `cursor` or `start` parameter in a request; they are mutually exclusive. To get the first cursor, set the `cursor` value to `initial`. In subsequent requests, specify the cursor value returned in the hits section of the response.

For more information, see [Paginating Results](#) in the *Amazon CloudSearch Developer Guide*.

- **expr** (*string*) – Defines one or more numeric expressions that can be used to sort results or specify search or filter criteria. You can also specify expressions as return fields.

You specify the expressions in JSON using the form `{"EXPRESSIONNAME": "EXPRESSION"}`. You can define and use multiple expressions in a search request. For example:

```
{"expression1": "_score*rating", "expression2": "(1/rank)*year"}
```

For information about the variables, operators, and functions you can use in expressions, see [Writing Expressions](#) in the *Amazon CloudSearch Developer Guide*.

- **facet** (*string*) – Specifies one or more fields for which to get facet information, and options that control how the facet information is returned. Each specified field must be facet-enabled in the domain configuration. The fields and options are specified in JSON using the form `{"FIELD": {"OPTION": VALUE, "OPTION": "STRING"}, "FIELD": {"OPTION": VALUE, "OPTION": "STRING"}}`.

You can specify the following faceting options:

- `buckets` specifies an array of the facet values or ranges to count. Ranges are specified using the same syntax that you use to search for a range of

values. For more information, see [Searching for a Range of Values](#) in the *Amazon CloudSearch Developer Guide*. Buckets are returned in the order they are specified in the request. The `sort` and `size` options are not valid if you specify `buckets`.

- `size` specifies the maximum number of facets to include in the results. By default, Amazon CloudSearch returns counts for the top 10. The `size` parameter is only valid when you specify the `sort` option; it cannot be used in conjunction with `buckets`.
- `sort` specifies how you want to sort the facets in the results: `bucket` or `count`. Specify `bucket` to sort alphabetically or numerically by facet value (in ascending order). Specify `count` to sort by the facet counts computed for each facet value (in descending order). To retrieve facet counts for particular values or ranges of values, use the `buckets` option instead of `sort`.

If no facet options are specified, facet counts are computed for all field values, the facets are sorted by facet count, and the top 10 facets are returned in the results.

To count particular buckets of values, use the `buckets` option. For example, the following request uses the `buckets` option to calculate and return facet counts by decade.

```
{"year":{"buckets":["[1970,1979]","[1980,1989]","[1990,1999]","[2000,2009]"]}}
```

To sort facets by facet count, use the `count` option. For example, the following request sets the `sort` option to `count` to sort the facet values by facet count, with the facet values that have the most matching documents listed first. Setting the `size` option to 3 returns only the top three facet values.

```
{"year":{"sort":"count","size":3}}
```

To sort the facets by value, use the `bucket` option. For example, the following request sets the `sort` option to `bucket` to sort the facet values numerically by year, with earliest year listed first.

```
{"year":{"sort":"bucket"}}
```

For more information, see [Getting and Using Facet Information](#) in the *Amazon CloudSearch Developer Guide*.

- **filterQuery** (*string*) – Specifies a structured query that filters the results of a search without affecting how the results are scored and sorted. You use `filterQuery` in conjunction with the `query` parameter to filter the documents that match the constraints specified in the `query` parameter. Specifying a filter controls only which matching documents are included in the results, it has no effect on how they are scored and sorted. The `filterQuery` parameter supports the full structured query syntax.

For more information about using filters, see [Filtering Matching Documents](#) in the *Amazon CloudSearch Developer Guide*.

- **highlight** (*string*) – Retrieves highlights for matches in the specified `text` or `text-array` fields. Each specified field must be `highlight` enabled in the domain configuration. The fields and options are specified in JSON using the form `{"FIELD":{"OPTION":VALUE,"OPTION":"STRING"},"FIELD":{"OPTION":VALUE,"OPTION":"STRING"}}`.

You can specify the following highlight options:

- `format` : specifies the format of the data in the text field: `text` or `html`. When data is returned as HTML, all non-alphanumeric characters are

encoded. The default is `html`.

- `max_phrases`: specifies the maximum number of occurrences of the search term(s) you want to highlight. By default, the first occurrence is highlighted.
- `pre_tag`: specifies the string to prepend to an occurrence of a search term. The default for HTML highlights is `lt;emgt;`. The default for text highlights is `*`.
- `post_tag`: specifies the string to append to an occurrence of a search term. The default for HTML highlights is `lt;/emgt;`. The default for text highlights is `*`.

If no highlight options are specified for a field, the returned field text is treated as HTML and the first match is highlighted with emphasis tags: `lt;emsearch-term``lt;/emgt;`.

For example, the following request retrieves highlights for the `actors` and `title` fields.

```
{ "actors": {}, "title": {"format":
"text", "max_phrases": 2, "pre_tag":
"***", "post_tag": "***" } }
```

- **partial** (*boolean*) – Enables partial results to be returned if one or more index partitions are unavailable. When your search index is partitioned across multiple search instances, by default Amazon CloudSearch only returns results if every partition can be queried. This means that the failure of a single search instance can result in 5xx (internal server) errors. When you enable partial results, Amazon CloudSearch returns whatever results are available and includes the percentage of documents searched in the search results (`percent-searched`). This enables you to more gracefully degrade your users' search experience. For example, rather than displaying no results, you could display the partial results and a message indicating that the results might be incomplete due to a temporary system outage.
- **query** (*string*) – **[REQUIRED]**

Specifies the search criteria for the request. How you specify the search criteria depends on the query parser used for the request and the parser options specified in the `queryOptions` parameter. By default, the `simple` query parser is used to process requests. To use the `structured`, `lucene`, or `dismax` query parser, you must also specify the `queryParser` parameter.

For more information about specifying search criteria, see [Searching Your Data](#) in the *Amazon CloudSearch Developer Guide*.

- **queryOptions** (*string*) – Configures options for the query parser specified in the `queryParser` parameter. You specify the options in JSON using the following form `{"OPTION1": "VALUE1", "OPTION2": "VALUE2" ... "OPTIONN": "VALUEN"}`.

The options you can configure vary according to which parser you use:

- `defaultOperator`: The default operator used to combine individual terms in the search string. For example: `defaultOperator: 'or'`. For the `dismax` parser, you specify a percentage that represents the percentage of terms in the search string (rounded down) that must match, rather than a default operator. A value of 0% is the equivalent to OR, and a value of 100% is equivalent to AND. The percentage must be specified as a value in the range 0-100 followed by the percent (%) symbol. For example, `defaultOperator: 50%`. Valid values: `and`, `or`, a percentage in the range 0%-100% (`dismax`). Default: `and` (`simple`, `structured`, `lucene`) or 100 (`dismax`). Valid for: `simple`,

- structured, lucene, and dismax.
- `fields`: An array of the fields to search when no fields are specified in a search. If no fields are specified in a search and this option is not specified, all text and text-array fields are searched. You can specify a weight for each field to control the relative importance of each field when Amazon CloudSearch calculates relevance scores. To specify a field weight, append a caret (^) symbol and the weight to the field name. For example, to boost the importance of the `title` field over the `description` field you could specify: `"fields":["title^5","description"]`. Valid values: The name of any configured field and an optional numeric value greater than zero. Default: All text and text-array fields. Valid for: simple, structured, lucene, and dismax.
 - `operators`: An array of the operators or special characters you want to disable for the simple query parser. If you disable the `and`, `or`, or `not` operators, the corresponding operators (`+`, `|`, `-`) have no special meaning and are dropped from the search string. Similarly, disabling `prefix` disables the wildcard operator (`*`) and disabling `phrase` disables the ability to search for phrases by enclosing phrases in double quotes. Disabling `precedence` disables the ability to control order of precedence using parentheses. Disabling `near` disables the ability to use the `~` operator to perform a sloppy phrase search. Disabling the `fuzzy` operator disables the ability to use the `~` operator to perform a fuzzy search. `escape` disables the ability to use a backslash (`\`) to escape special characters within the search string. Disabling `whitespace` is an advanced option that prevents the parser from tokenizing on whitespace, which can be useful for Vietnamese. (It prevents Vietnamese words from being split incorrectly.) For example, you could disable all operators other than the phrase operator to support just simple term and phrase queries: `"operators":["and","not","or","prefix"]`. Valid values: `and`, `escape`, `fuzzy`, `near`, `not`, `or`, `phrase`, `precedence`, `prefix`, `whitespace`. Default: All operators and special characters are enabled. Valid for: simple.
 - `phraseFields`: An array of the text or text-array fields you want to use for phrase searches. When the terms in the search string appear in close proximity within a field, the field scores higher. You can specify a weight for each field to boost that score. The `phraseSlop` option controls how much the matches can deviate from the search string and still be boosted. To specify a field weight, append a caret (^) symbol and the weight to the field name. For example, to boost phrase matches in the `title` field over the `abstract` field, you could specify: `"phraseFields":["title^3","plot"]`. Valid values: The name of any text or text-array field and an optional numeric value greater than zero. Default: No fields. If you don't specify any fields with `phraseFields`, proximity scoring is disabled even if `phraseSlop` is specified. Valid for: dismax.
 - `phraseSlop`: An integer value that specifies how much matches can deviate from the search phrase and still be boosted according to the weights specified in the `phraseFields` option; for example, `phraseSlop: 2`. You must also specify `phraseFields` to enable proximity scoring. Valid values: positive integers. Default: 0. Valid for: dismax.
 - `explicitPhraseSlop`: An integer value that specifies how much a match can deviate from the search phrase when the phrase is enclosed in double quotes in the search string. (Phrases that exceed this proximity distance are not considered a match.) For example, to

specify a slop of three for dismax phrase queries, you would specify `"explicitPhraseSlop":3`. Valid values: positive integers. Default: 0. Valid for: `dismax`.

- `tieBreaker`: When a term in the search string is found in a document's field, a score is calculated for that field based on how common the word is in that field compared to other documents. If the term occurs in multiple fields within a document, by default only the highest scoring field contributes to the document's overall score. You can specify a `tieBreaker` value to enable the matches in lower-scoring fields to contribute to the document's score. That way, if two documents have the same max field score for a particular term, the score for the document that has matches in more fields will be higher. The formula for calculating the score with a `tieBreaker` is $(\text{max field score}) + (\text{tieBreaker}) * (\text{sum of the scores for the rest of the matching fields})$. Set `tieBreaker` to 0 to disregard all but the highest scoring field (pure max): `"tieBreaker":0`. Set to 1 to sum the scores from all fields (pure sum): `"tieBreaker":1`. Valid values: 0.0 to 1.0. Default: 0.0. Valid for: `dismax`.
- **queryParser** (*string*) – Specifies which query parser to use to process the request. If `queryParser` is not specified, Amazon CloudSearch uses the simple query parser.

Amazon CloudSearch supports four query parsers:

- `simple`: perform simple searches of `text` and `text-array` fields. By default, the simple query parser searches all `text` and `text-array` fields. You can specify which fields to search by with the `queryOptions` parameter. If you prefix a search term with a plus sign (+) documents must contain the term to be considered a match. (This is the default, unless you configure the default operator with the `queryOptions` parameter.) You can use the `-` (NOT), `|` (OR), and `*` (wildcard) operators to exclude particular terms, find results that match any of the specified terms, or search for a prefix. To search for a phrase rather than individual terms, enclose the phrase in double quotes. For more information, see [Searching for Text](#) in the *Amazon CloudSearch Developer Guide*.
- `structured`: perform advanced searches by combining multiple expressions to define the search criteria. You can also search within particular fields, search for values and ranges of values, and use advanced options such as term boosting, `matchall`, and `near`. For more information, see [Constructing Compound Queries](#) in the *Amazon CloudSearch Developer Guide*.
- `lucene`: search using the Apache Lucene query parser syntax. For more information, see [Apache Lucene Query Parser Syntax](#).
- `dismax`: search using the simplified subset of the Apache Lucene query parser syntax defined by the DisMax query parser. For more information, see [DisMax Query Parser Syntax](#).
- **return** (*string*) – Specifies the field and expression values to include in the response. Multiple fields or expressions are specified as a comma-separated list. By default, a search response includes all return enabled fields (`_all_fields`). To return only the document IDs for the matching documents, specify `_no_fields`. To retrieve the relevance score calculated for each document, specify `_score`.
- **size** (*integer*) – Specifies the maximum number of search hits to include in the response.

- **sort** (*string*) – Specifies the fields or custom expressions to use to sort the search results. Multiple fields or expressions are specified as a comma-separated list. You must specify the sort direction (`asc` or `desc`) for each field; for example, `year desc,title asc`. To use a field to sort results, the field must be sort-enabled in the domain configuration. Array type fields cannot be used for sorting. If no `sort` parameter is specified, results are sorted by their default relevance scores in descending order: `_score desc`. You can also sort by document ID (`_id asc`) and version (`_version desc`).

For more information, see [Sorting Results](#) in the *Amazon CloudSearch Developer Guide*.

- **start** (*integer*) – Specifies the offset of the first search hit you want to return. Note that the result set is zero-based; the first result is at index 0. You can specify either the `start` or `cursor` parameter in a request, they are mutually exclusive.

For more information, see [Paginating Results](#) in the *Amazon CloudSearch Developer Guide*.

Return type dict

Returns

Response Syntax

```
{
  'status': {
    'timems': 123,
    'rid': 'string'
  },
  'hits': {
    'found': 123,
    'start': 123,
    'cursor': 'string',
    'hit': [
      {
        'id': 'string',
        'fields': {
          'string': [
            'string',
          ]
        },
        'exprs': {
          'string': 'string'
        },
        'highlights': {
          'string': 'string'
        }
      },
    ],
  ],
  'facets': {
    'string': {
      'buckets': [
        {
          'value': 'string',
          'count': 123
        },
      ],
    },
  ]
}
```

```
}  
}
```

Response Structure

- *(dict)* –

The result of a `Search` request. Contains the documents that match the specified search criteria and any requested fields, highlights, and facet information.

- **status** (*dict*) –

The status information returned for the search request.

- * **timems** (*integer*) –

How long it took to process the request, in milliseconds.

- * **rid** (*string*) –

The encrypted resource ID for the request.

- **hits** (*dict*) –

The documents that match the search criteria.

- * **found** (*integer*) –

The total number of documents that match the search request.

- * **start** (*integer*) –

The index of the first matching document.

- * **cursor** (*string*) –

A cursor that can be used to retrieve the next set of matching documents when you want to page through a large result set.

- * **hit** (*list*) –

A document that matches the search request.

- (*dict*) –

Information about a document that matches the search request.

- **id** (*string*) –

The document ID of a document that matches the search request.

- **fields** (*dict*) –

The fields returned from a document that matches the search request.

- (*string*) –

- (*list*) –

- (*string*) –

- **exprs** (*dict*) –

The expressions returned from a document that matches the search request.

- (*string*) –

- (*string*) –

- **highlights** (*dict*) –

The highlights returned from a document that matches the search request.

- (*string*) –

- (*string*) –

– **facets** (*dict*) –

The requested facet information.

* (*string*) –

· (*dict*) –

A container for the calculated facet values and counts.

· **buckets** (*list*) –

A list of the calculated facet values and counts.

· (*dict*) –

A container for facet information.

· **value** (*string*) –

The facet value being counted.

· **count** (*integer*) –

The number of hits that contain the facet value in the specified facet field.

suggest (***kwargs*)

Retrieves autocomplete suggestions for a partial query string. You can use suggestions enable you to display likely matches before users finish typing. In Amazon CloudSearch, suggestions are based on the contents of a particular text field. When you request suggestions, Amazon CloudSearch finds all of the documents whose values in the suggester field start with the specified query string. The beginning of the field must match the query string to be considered a match.

For more information about configuring suggesters and retrieving suggestions, see [Getting Suggestions](#) in the *Amazon CloudSearch Developer Guide*.

The endpoint for submitting Suggest requests is domain-specific. You submit suggest requests to a domain's search endpoint. To get the search endpoint for your domain, use the Amazon CloudSearch configuration service `DescribeDomains` action. A domain's endpoints are also displayed on the domain dashboard in the Amazon CloudSearch console.

Request Syntax

```
response = client.suggest(
    query='string',
    suggester='string',
    size=123
)
```

Parameters

• **query** (*string*) – **[REQUIRED]**

Specifies the string for which you want to get suggestions.

• **suggester** (*string*) – **[REQUIRED]**

Specifies the name of the suggester to use to find suggested matches.

• **size** (*integer*) – Specifies the maximum number of suggestions to return.

Return type dict

Returns

Response Syntax

```
{
    'status': {
        'timems': 123,
        'rid': 'string'
    }
}
```

```
    },
    'suggest': {
        'query': 'string',
        'found': 123,
        'suggestions': [
            {
                'suggestion': 'string',
                'score': 123,
                'id': 'string'
            },
        ]
    }
}
```

Response Structure

- *(dict)* –

Contains the response to a `Suggest` request.

- **status** (*dict*) –

The status of a `SuggestRequest`. Contains the resource ID (`rid`) and how long it took to process the request (`timems`).

- * **timems** (*integer*) –

How long it took to process the request, in milliseconds.

- * **rid** (*string*) –

The encrypted resource ID for the request.

- **suggest** (*dict*) –

Container for the matching search suggestion information.

- * **query** (*string*) –

The query string specified in the suggest request.

- * **found** (*integer*) –

The number of documents that were found to match the query string.

- * **suggestions** (*list*) –

The documents that match the query string.

- (*dict*) –

An autocomplete suggestion that matches the query string specified in a `SuggestRequest`.

- **suggestion** (*string*) –

The string that matches the query string specified in the `SuggestRequest`.

- **score** (*integer*) –

The relevance score of a suggested match.

- **id** (*string*) –

The document ID of the suggested document.

upload_documents (***kwargs*)

Posts a batch of documents to a search domain for indexing. A document batch is a collection of add and delete operations that represent the documents you want to add, update, or delete from your domain. Batches can be described in either JSON or XML. Each item that you want Amazon CloudSearch to return as a search result (such as a product) is represented as a document. Every document has a unique ID and one or more fields that contain the data that you want to search and return in results. Individual

documents cannot contain more than 1 MB of data. The entire batch cannot exceed 5 MB. To get the best possible upload performance, group add and delete operations in batches that are close the 5 MB limit. Submitting a large volume of single-document batches can overload a domain's document service.

The endpoint for submitting `UploadDocuments` requests is domain-specific. To get the document endpoint for your domain, use the Amazon CloudSearch configuration service `DescribeDomains` action. A domain's endpoints are also displayed on the domain dashboard in the Amazon CloudSearch console.

For more information about formatting your data for Amazon CloudSearch, see [Preparing Your Data](#) in the *Amazon CloudSearch Developer Guide*. For more information about uploading data for indexing, see [Uploading Data](#) in the *Amazon CloudSearch Developer Guide*.

Request Syntax

```
response = client.upload_documents(
    documents=b'bytes',
    contentType='application/json'|'application/xml'
)
```

Parameters

- **documents** (*bytes*) – [REQUIRED]

A batch of documents formatted in JSON or HTML.

- **contentType** (*string*) – [REQUIRED]

The format of the batch you are uploading. Amazon CloudSearch supports two document batch formats:

- `application/json`
- `application/xml`

Return type dict

Returns

Response Syntax

```
{
    'status': 'string',
    'adds': 123,
    'deletes': 123,
    'warnings': [
        {
            'message': 'string'
        },
    ],
}
```

Response Structure

- (*dict*) –

Contains the response to an `UploadDocuments` request.

- **status** (*string*) –

The status of an `UploadDocumentsRequest`.

- **adds** (*integer*) –

The number of documents that were added to the search domain.

- **deletes** (*integer*) –

The number of documents that were deleted from the search domain.

– **warnings** (*list*) –

Any warnings returned by the document service about the documents being uploaded.

* (*dict*) –

A warning returned by the document service when an issue is discovered while processing an upload request.

· **message** (*string*) –

The description for a warning returned by the document service.

CloudTrail

Table of Contents

- CloudTrail
 - Client

Client

class CloudTrail.Client

A low-level client representing AWS CloudTrail:

```
import boto3

client = boto3.client('cloudtrail')
```

These are the available methods:

- `can_paginate()`
- `create_trail()`
- `delete_trail()`
- `describe_trails()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_trail_status()`
- `get_waiter()`
- `lookup_events()`
- `start_logging()`
- `stop_logging()`
- `update_trail()`

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

create_trail (***kwargs*)

From the command line, use `create-subscription`.

Creates a trail that specifies the settings for delivery of log data to an Amazon S3 bucket.

Request Syntax

```
response = client.create_trail(
    Name='string',
    S3BucketName='string',
    S3KeyPrefix='string',
    SnsTopicName='string',
    IncludeGlobalServiceEvents=True|False,
    CloudWatchLogsLogGroupArn='string',
    CloudWatchLogsRoleArn='string'
)
```

Parameters

- **Name** (*string*) – **[REQUIRED]**
Specifies the name of the trail.
- **S3BucketName** (*string*) – **[REQUIRED]**
Specifies the name of the Amazon S3 bucket designated for publishing log files.
- **S3KeyPrefix** (*string*) – Specifies the Amazon S3 key prefix that precedes the name of the bucket you have designated for log file delivery.
- **SnsTopicName** (*string*) – Specifies the name of the Amazon SNS topic defined for notification of log file delivery.
- **IncludeGlobalServiceEvents** (*boolean*) – Specifies whether the trail is publishing events from global services such as IAM to the log files.
- **CloudWatchLogsLogGroupArn** (*string*) – Specifies a log group name using an Amazon Resource Name (ARN), a unique identifier that represents the log group to which CloudTrail logs will be delivered. Not required unless you specify CloudWatchLogsRoleArn.
- **CloudWatchLogsRoleArn** (*string*) – Specifies the role for the CloudWatch Logs endpoint to assume to write to a user's log group.

Return type dict

Returns

Response Syntax

```
{
    'Name': 'string',
    'S3BucketName': 'string',
    'S3KeyPrefix': 'string',
    'SnsTopicName': 'string',
    'IncludeGlobalServiceEvents': True|False,
    'CloudWatchLogsLogGroupArn': 'string',
    'CloudWatchLogsRoleArn': 'string'
}
```

Response Structure

- (*dict*) – Returns the objects or data listed below if successful. Otherwise, returns an error.
 - **Name** (*string*) –
Specifies the name of the trail.
 - **S3BucketName** (*string*) –
Specifies the name of the Amazon S3 bucket designated for publishing log files.

- **S3KeyPrefix** (*string*) –
Specifies the Amazon S3 key prefix that precedes the name of the bucket you have designated for log file delivery.
- **SnsTopicName** (*string*) –
Specifies the name of the Amazon SNS topic defined for notification of log file delivery.
- **IncludeGlobalServiceEvents** (*boolean*) –
Specifies whether the trail is publishing events from global services such as IAM to the log files.
- **CloudWatchLogsLogGroupArn** (*string*) –
Specifies the Amazon Resource Name (ARN) of the log group to which CloudTrail logs will be delivered.
- **CloudWatchLogsRoleArn** (*string*) –
Specifies the role for the CloudWatch Logs endpoint to assume to write to a user's log group.

delete_trail (***kwargs*)

Deletes a trail.

Request Syntax

```
response = client.delete_trail(  
    Name='string'  
)
```

Parameters **Name** (*string*) – **[REQUIRED]**

The name of a trail to be deleted.

Return type dict

Returns

Response Syntax

```
{}
```

Response Structure

- (*dict*) –

Returns the objects or data listed below if successful. Otherwise, returns an error.

describe_trails (***kwargs*)

Retrieves settings for the trail associated with the current region for your account.

Request Syntax

```
response = client.describe_trails(  
    trailNameList=[  
        'string',  
    ]  
)
```

Parameters **trailNameList** (*list*) – The trail returned.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'trailList': [
    {
      'Name': 'string',
      'S3BucketName': 'string',
      'S3KeyPrefix': 'string',
      'SnsTopicName': 'string',
      'IncludeGlobalServiceEvents': True|False,
      'CloudWatchLogsLogGroupArn': 'string',
      'CloudWatchLogsRoleArn': 'string'
    },
  ]
}
```

Response Structure

- (*dict*) –

Returns the objects or data listed below if successful. Otherwise, returns an error.

- **trailList** (*list*) –

The list of trails.

- * (*dict*) –

The settings for a trail.

- **Name** (*string*) –

Name of the trail set by calling CreateTrail .

- **S3BucketName** (*string*) –

Name of the Amazon S3 bucket into which CloudTrail delivers your trail files.

- **S3KeyPrefix** (*string*) –

Value of the Amazon S3 prefix.

- **SnsTopicName** (*string*) –

Name of the existing Amazon SNS topic that CloudTrail uses to notify the account owner when new CloudTrail log files have been delivered.

- **IncludeGlobalServiceEvents** (*boolean*) –

Set to **True** to include AWS API calls from AWS global services such as IAM. Otherwise, **False** .

- **CloudWatchLogsLogGroupArn** (*string*) –

Specifies an Amazon Resource Name (ARN), a unique identifier that represents the log group to which CloudTrail logs will be delivered.

- **CloudWatchLogsRoleArn** (*string*) –

Specifies the role for the CloudWatch Logs endpoint to assume to write to a user's log group.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for

- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_trail_status (***kwargs*)

Returns a JSON-formatted list of information about the specified trail. Fields include information on delivery errors, Amazon SNS and Amazon S3 errors, and start and stop logging times for each trail.

Request Syntax

```
response = client.get_trail_status(
    Name='string'
)
```

Parameters **Name** (*string*) – **[REQUIRED]**

The name of the trail for which you are requesting the current status.

Return type `dict`

Returns

Response Syntax

```
{
    'IsLogging': True|False,
    'LatestDeliveryError': 'string',
    'LatestNotificationError': 'string',
    'LatestDeliveryTime': datetime(2015, 1, 1),
    'LatestNotificationTime': datetime(2015, 1, 1),
    'StartLoggingTime': datetime(2015, 1, 1),
    'StopLoggingTime': datetime(2015, 1, 1),
    'LatestCloudWatchLogsDeliveryError': 'string',
    'LatestCloudWatchLogsDeliveryTime': datetime(2015, 1, 1)
}
```

Response Structure

- (*dict*) –

Returns the objects or data listed below if successful. Otherwise, returns an error.

- **IsLogging** (*boolean*) –

Whether the CloudTrail is currently logging AWS API calls.

- **LatestDeliveryError** (*string*) –

Displays any Amazon S3 error that CloudTrail encountered when attempting to deliver log files to the designated bucket. For more information see the topic [Error Responses](#) in the Amazon S3 API Reference.

– **LatestNotificationError** (*string*) –

Displays any Amazon SNS error that CloudTrail encountered when attempting to send a notification. For more information about Amazon SNS errors, see the [Amazon SNS Developer Guide](#).

– **LatestDeliveryTime** (*datetime*) –

Specifies the date and time that CloudTrail last delivered log files to an account's Amazon S3 bucket.

– **LatestNotificationTime** (*datetime*) –

Specifies the date and time of the most recent Amazon SNS notification that CloudTrail has written a new log file to an account's Amazon S3 bucket.

– **StartLoggingTime** (*datetime*) –

Specifies the most recent date and time when CloudTrail started recording API calls for an AWS account.

– **StopLoggingTime** (*datetime*) –

Specifies the most recent date and time when CloudTrail stopped recording API calls for an AWS account.

– **LatestCloudWatchLogsDeliveryError** (*string*) –

Displays any CloudWatch Logs error that CloudTrail encountered when attempting to deliver logs to CloudWatch Logs.

– **LatestCloudWatchLogsDeliveryTime** (*datetime*) –

Displays the most recent date and time when CloudTrail delivered logs to CloudWatch Logs.

get_waiter (*waiter_name*)

lookup_events (***kwargs*)

Looks up API activity events captured by CloudTrail that create, update, or delete resources in your account. Events for a region can be looked up for the times in which you had CloudTrail turned on in that region during the last seven days. Lookup supports five different attributes: time range (defined by a start time and end time), user name, event name, resource type, and resource name. All attributes are optional. The maximum number of attributes that can be specified in any one lookup request are time range and one other attribute. The default number of results returned is 10, with a maximum of 50 possible. The response includes a token that you can use to get the next page of results. The rate of lookup requests is limited to one per second per account.

Warning: Events that occurred during the selected time range will not be available for lookup if CloudTrail logging was not enabled when the events occurred.

Request Syntax

```
response = client.lookup_events(
    LookupAttributes=[
        {
            'AttributeKey': 'EventId' | 'EventName' | 'Username' | 'ResourceType' | 'ResourceName',
            'AttributeValue': 'string'
        },
    ],
```

```
StartTime=datetime(2015, 1, 1),
EndTime=datetime(2015, 1, 1),
MaxResults=123,
NextToken='string'
)
```

Parameters

- **LookupAttributes** (*list*) – Contains a list of lookup attributes. Currently the list can contain only one item.
 - (*dict*) –
 - Specifies an attribute and value that filter the events returned.
 - * **AttributeKey** (*string*) – [REQUIRED]
 - Specifies an attribute on which to filter the events returned.
 - * **AttributeValue** (*string*) – [REQUIRED]
 - Specifies a value for the specified AttributeKey.
 - **StartTime** (*datetime*) – Specifies that only events that occur after or at the specified time are returned. If the specified start time is after the specified end time, an error is returned.
 - **EndTime** (*datetime*) – Specifies that only events that occur before or at the specified time are returned. If the specified end time is before the specified start time, an error is returned.
 - **MaxResults** (*integer*) – The number of events to return. Possible values are 1 through 50. The default is 10.
 - **NextToken** (*string*) – The token to use to get the next page of results after a previous API call. This token must be passed in with the same parameters that were specified in the the original call. For example, if the original call specified an AttributeKey of 'Username' with a value of 'root', the call with NextToken should include those same parameters.

Return type dict

Returns

Response Syntax

```
{
  'Events': [
    {
      'EventId': 'string',
      'EventName': 'string',
      'EventTime': datetime(2015, 1, 1),
      'Username': 'string',
      'Resources': [
        {
          'ResourceType': 'string',
          'ResourceName': 'string'
        },
      ],
      'CloudTrailEvent': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains a response to a LookupEvents action.

– **Events** (*list*) –

A list of events returned based on the lookup attributes specified and the CloudTrail event. The events list is sorted by time. The most recent event is listed first.

* (*dict*) –

Contains information about an event that was returned by a lookup request. The result includes a representation of a CloudTrail event.

· **EventId** (*string*) –

The CloudTrail ID of the event returned.

· **EventName** (*string*) –

The name of the event returned.

· **EventTime** (*datetime*) –

The date and time of the event returned.

· **Username** (*string*) –

A user name or role name of the requester that called the API in the event returned.

· **Resources** (*list*) –

A list of resources referenced by the event returned.

· (*dict*) –

Specifies the type and name of a resource referenced by an event.

· **ResourceType** (*string*) –

The type of a resource referenced by the event returned. When the resource type cannot be determined, null is returned. Some examples of resource types are: **Instance** for EC2, **Trail** for CloudTrail, **DBInstance** for RDS, and **AccessKey** for IAM. For a list of resource types supported for event lookup, see [Resource Types Supported for Event Lookup](#).

· **ResourceName** (*string*) –

The name of the resource referenced by the event returned. These are user-created names whose values will depend on the environment. For example, the resource name might be “auto-scaling-test-group” for an Auto Scaling Group or “i-1234567” for an EC2 Instance.

· **CloudTrailEvent** (*string*) –

A JSON string that contains a representation of the event returned.

– **NextToken** (*string*) –

The token to use to get the next page of results after a previous API call. If the token does not appear, there are no more results to return. The token must be passed in with the same parameters as the previous call. For example, if the original call specified an AttributeKey of ‘Username’ with a value of ‘root’, the call with NextToken should include those same parameters.

start_logging (***kwargs*)

Starts the recording of AWS API calls and log file delivery for a trail.

Request Syntax

```
response = client.start_logging(  
    Name='string'  
)
```

Parameters *Name* (*string*) – [REQUIRED]

The name of the trail for which CloudTrail logs AWS API calls.

Return type dict

Returns**Response Syntax**

```
{}
```

Response Structure

- (*dict*) –

Returns the objects or data listed below if successful. Otherwise, returns an error.

stop_logging (***kwargs*)

Suspends the recording of AWS API calls and log file delivery for the specified trail. Under most circumstances, there is no need to use this action. You can update a trail without stopping it first. This action is the only way to stop recording.

Request Syntax

```
response = client.stop_logging(  
    Name='string'  
)
```

Parameters *Name* (*string*) – [REQUIRED]

Communicates to CloudTrail the name of the trail for which to stop logging AWS API calls.

Return type dict

Returns**Response Syntax**

```
{}
```

Response Structure

- (*dict*) –

Returns the objects or data listed below if successful. Otherwise, returns an error.

update_trail (***kwargs*)

From the command line, use `update-subscription`.

Updates the settings that specify delivery of log files. Changes to a trail do not require stopping the CloudTrail service. Use this action to designate an existing bucket for log delivery. If the existing bucket has previously been a target for CloudTrail log files, an IAM policy exists for the bucket.

Request Syntax


```
response = client.update_trail(
    Name='string',
    S3BucketName='string',
    S3KeyPrefix='string',
    SnsTopicName='string',
    IncludeGlobalServiceEvents=True|False,
    CloudWatchLogsLogGroupArn='string',
    CloudWatchLogsRoleArn='string'
)
```

Parameters

- **Name** (*string*) – **[REQUIRED]**
Specifies the name of the trail.
- **S3BucketName** (*string*) – Specifies the name of the Amazon S3 bucket designated for publishing log files.
- **S3KeyPrefix** (*string*) – Specifies the Amazon S3 key prefix that precedes the name of the bucket you have designated for log file delivery.
- **SnsTopicName** (*string*) – Specifies the name of the Amazon SNS topic defined for notification of log file delivery.
- **IncludeGlobalServiceEvents** (*boolean*) – Specifies whether the trail is publishing events from global services such as IAM to the log files.
- **CloudWatchLogsLogGroupArn** (*string*) – Specifies a log group name using an Amazon Resource Name (ARN), a unique identifier that represents the log group to which CloudTrail logs will be delivered. Not required unless you specify CloudWatchLogsRoleArn.
- **CloudWatchLogsRoleArn** (*string*) – Specifies the role for the CloudWatch Logs endpoint to assume to write to a user's log group.

Return type dict

Returns

Response Syntax

```
{
    'Name': 'string',
    'S3BucketName': 'string',
    'S3KeyPrefix': 'string',
    'SnsTopicName': 'string',
    'IncludeGlobalServiceEvents': True|False,
    'CloudWatchLogsLogGroupArn': 'string',
    'CloudWatchLogsRoleArn': 'string'
}
```

Response Structure

- (*dict*) – Returns the objects or data listed below if successful. Otherwise, returns an error.
 - **Name** (*string*) –
Specifies the name of the trail.
 - **S3BucketName** (*string*) –
Specifies the name of the Amazon S3 bucket designated for publishing log files.
 - **S3KeyPrefix** (*string*) –
Specifies the Amazon S3 key prefix that precedes the name of the bucket you have designated for log file delivery.

- **SnsTopicName** (*string*) –
Specifies the name of the Amazon SNS topic defined for notification of log file delivery.
- **IncludeGlobalServiceEvents** (*boolean*) –
Specifies whether the trail is publishing events from global services such as IAM to the log files.
- **CloudWatchLogsLogGroupArn** (*string*) –
Specifies the Amazon Resource Name (ARN) of the log group to which CloudTrail logs will be delivered.
- **CloudWatchLogsRoleArn** (*string*) –
Specifies the role for the CloudWatch Logs endpoint to assume to write to a user's log group.

CloudWatch

Table of Contents

- CloudWatch
 - Client
 - Paginators

Client

class CloudWatch.Client

A low-level client representing Amazon CloudWatch:

```
import boto3

client = boto3.client('cloudwatch')
```

These are the available methods:

- `can_paginate()`
- `delete_alarms()`
- `describe_alarm_history()`
- `describe_alarms()`
- `describe_alarms_for_metric()`
- `disable_alarm_actions()`
- `enable_alarm_actions()`
- `generate_presigned_url()`
- `get_metric_statistics()`
- `get_paginator()`
- `get_waiter()`
- `list_metrics()`
- `put_metric_alarm()`
- `put_metric_data()`
- `set_alarm_state()`

can_paginate (*operation_name*)
Check if an operation can be paginated.

Parameters `operation_name` (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

delete_alarms (***kwargs*)

Deletes all specified alarms. In the event of an error, no alarms are deleted.

Request Syntax

```
response = client.delete_alarms(
    AlarmNames=[
        'string',
    ]
)
```

Parameters `AlarmNames` (*list*) – [REQUIRED]

A list of alarms to be deleted.

- (*string*) –

Returns None

describe_alarm_history (***kwargs*)

Retrieves history for the specified alarm. Filter alarms by date range or item type. If an alarm name is not specified, Amazon CloudWatch returns histories for all of the owner's alarms.

Request Syntax

```
response = client.describe_alarm_history(
    AlarmName='string',
    HistoryItemType='ConfigurationUpdate'|'StateUpdate'|'Action',
    StartDate=datetime(2015, 1, 1),
    EndDate=datetime(2015, 1, 1),
    MaxRecords=123,
    NextToken='string'
)
```

Parameters

- **AlarmName** (*string*) – The name of the alarm.
- **HistoryItemType** (*string*) – The type of alarm histories to retrieve.
- **StartDate** (*datetime*) – The starting date to retrieve alarm history.
- **EndDate** (*datetime*) – The ending date to retrieve alarm history.
- **MaxRecords** (*integer*) – The maximum number of alarm history records to retrieve.
- **NextToken** (*string*) – The token returned by a previous call to indicate that there is more data available.

Return type dict

Returns

Response Syntax

```
{
    'AlarmHistoryItems': [
        {
            'AlarmName': 'string',
            'Timestamp': datetime(2015, 1, 1),
```

```
        'HistoryItemType': 'ConfigurationUpdate'|'StateUpdate'|'Action',
        'HistorySummary': 'string',
        'HistoryData': 'string'
    },
],
'NextToken': 'string'
}
```

Response Structure

- *(dict)* –

The output for the DescribeAlarmHistory action.

- **AlarmHistoryItems** (*list*) –

A list of alarm histories in JSON format.

- * *(dict)* –

The `AlarmHistoryItem` data type contains descriptive information about the history of a specific alarm. If you call `DescribeAlarmHistory`, Amazon CloudWatch returns this data type as part of the `DescribeAlarmHistoryResult` data type.

- **AlarmName** (*string*) –

The descriptive name for the alarm.

- **Timestamp** (*datetime*) –

The time stamp for the alarm history item. Amazon CloudWatch uses Coordinated Universal Time (UTC) when returning time stamps, which do not accommodate seasonal adjustments such as daylight savings time. For more information, see [Time stamps](#) in the *Amazon CloudWatch Developer Guide*.

- **HistoryItemType** (*string*) –

The type of alarm history item.

- **HistorySummary** (*string*) –

A human-readable summary of the alarm history.

- **HistoryData** (*string*) –

Machine-readable data about the alarm in JSON format.

- **NextToken** (*string*) –

A string that marks the start of the next batch of returned results.

`describe_alarms` (***kwargs*)

Retrieves alarms with the specified names. If no name is specified, all alarms for the user are returned. Alarms can be retrieved by using only a prefix for the alarm name, the alarm state, or a prefix for any action.

Request Syntax

```
response = client.describe_alarms(
    AlarmNames=[
        'string',
    ],
    AlarmNamePrefix='string',
    StateValue='OK'|'ALARM'|'INSUFFICIENT_DATA',
    ActionPrefix='string',
    MaxRecords=123,
```

```

NextToken='string'
)

```

Parameters

- **AlarmNames** (*list*) – A list of alarm names to retrieve information for.
– (*string*) –
- **AlarmNamePrefix** (*string*) – The alarm name prefix. AlarmNames cannot be specified if this parameter is specified.
- **StateValue** (*string*) – The state value to be used in matching alarms.
- **ActionPrefix** (*string*) – The action name prefix.
- **MaxRecords** (*integer*) – The maximum number of alarm descriptions to retrieve.
- **NextToken** (*string*) – The token returned by a previous call to indicate that there is more data available.

Return type dict

Returns

Response Syntax

```

{
  'MetricAlarms': [
    {
      'AlarmName': 'string',
      'AlarmArn': 'string',
      'AlarmDescription': 'string',
      'AlarmConfigurationUpdatedTimestamp': datetime(2015, 1, 1),
      'ActionsEnabled': True|False,
      'OKActions': [
        'string',
      ],
      'AlarmActions': [
        'string',
      ],
      'InsufficientDataActions': [
        'string',
      ],
      'StateValue': 'OK'|'ALARM'|'INSUFFICIENT_DATA',
      'StateReason': 'string',
      'StateReasonData': 'string',
      'StateUpdatedTimestamp': datetime(2015, 1, 1),
      'MetricName': 'string',
      'Namespace': 'string',
      'Statistic': 'SampleCount'|'Average'|'Sum'|'Minimum'|'Maximum',
      'Dimensions': [
        {
          'Name': 'string',
          'Value': 'string'
        },
      ],
      'Period': 123,
      'Unit': 'Seconds'|'Microseconds'|'Milliseconds'|'Bytes'|'Kilobytes'|'M
      'EvaluationPeriods': 123,
      'Threshold': 123.0,
      'ComparisonOperator': 'GreaterThanOrEqualToThreshold'|'GreaterThanThre
    },
  ],
  'NextToken': 'string'
}

```

Response Structure

- *(dict)* –

The output for the DescribeAlarms action.

- **MetricAlarms** (*list*) –

A list of information for the specified alarms.

- * *(dict)* –

The MetricAlarm data type represents an alarm. You can use PutMetricAlarm to create or update an alarm.

- **AlarmName** (*string*) –

The name of the alarm.

- **AlarmArn** (*string*) –

The Amazon Resource Name (ARN) of the alarm.

- **AlarmDescription** (*string*) –

The description for the alarm.

- **AlarmConfigurationUpdatedTimestamp** (*datetime*) –

The time stamp of the last update to the alarm configuration. Amazon CloudWatch uses Coordinated Universal Time (UTC) when returning time stamps, which do not accommodate seasonal adjustments such as daylight savings time. For more information, see [Time stamps](#) in the *Amazon CloudWatch Developer Guide*.

- **ActionsEnabled** (*boolean*) –

Indicates whether actions should be executed during any changes to the alarm's state.

- **OKActions** (*list*) –

The list of actions to execute when this alarm transitions into an OK state from any other state. Each action is specified as an Amazon Resource Number (ARN). Currently the only actions supported are publishing to an Amazon SNS topic and triggering an Auto Scaling policy.

- (*string*) –

- **AlarmActions** (*list*) –

The list of actions to execute when this alarm transitions into an ALARM state from any other state. Each action is specified as an Amazon Resource Number (ARN). Currently the only actions supported are publishing to an Amazon SNS topic and triggering an Auto Scaling policy.

- (*string*) –

- **InsufficientDataActions** (*list*) –

The list of actions to execute when this alarm transitions into an INSUFFICIENT_DATA state from any other state. Each action is specified as an Amazon Resource Number (ARN). Currently the only actions supported are publishing to an Amazon SNS topic or triggering an Auto Scaling policy.

Warning: The current WSDL lists this attribute as `UnknownActions`.

- (*string*) –

- **StateValue** (*string*) –
The state value for the alarm.
- **StateReason** (*string*) –
A human-readable explanation for the alarm’s state.
- **StateReasonData** (*string*) –
An explanation for the alarm’s state in machine-readable JSON format
- **StateUpdatedTimestamp** (*datetime*) –
The time stamp of the last update to the alarm’s state. Amazon CloudWatch uses Coordinated Universal Time (UTC) when returning time stamps, which do not accommodate seasonal adjustments such as daylight savings time. For more information, see [Time stamps](#) in the *Amazon CloudWatch Developer Guide*.
- **MetricName** (*string*) –
The name of the alarm’s metric.
- **Namespace** (*string*) –
The namespace of alarm’s associated metric.
- **Statistic** (*string*) –
The statistic to apply to the alarm’s associated metric.
- **Dimensions** (*list*) –
The list of dimensions associated with the alarm’s associated metric.
- (*dict*) –
The `Dimension` data type further expands on the identity of a metric using a Name, Value pair.

For examples that use one or more dimensions, see `PutMetricData`.
- **Name** (*string*) –
The name of the dimension.
- **Value** (*string*) –
The value representing the dimension measurement
- **Period** (*integer*) –
The period in seconds over which the statistic is applied.
- **Unit** (*string*) –
The unit of the alarm’s associated metric.
- **EvaluationPeriods** (*integer*) –
The number of periods over which data is compared to the specified threshold.
- **Threshold** (*float*) –
The value against which the specified statistic is compared.
- **ComparisonOperator** (*string*) –
The arithmetic operation to use when comparing the specified `Statistic` and `Threshold`. The specified `Statistic` value is used as the first operand.

– **NextToken** (*string*) –

A string that marks the start of the next batch of returned results.

describe_alarms_for_metric (***kwargs*)

Retrieves all alarms for a single metric. Specify a statistic, period, or unit to filter the set of alarms further.

Request Syntax

```
response = client.describe_alarms_for_metric(
    MetricName='string',
    Namespace='string',
    Statistic='SampleCount'|'Average'|'Sum'|'Minimum'|'Maximum',
    Dimensions=[
        {
            'Name': 'string',
            'Value': 'string'
        },
    ],
    Period=123,
    Unit='Seconds'|'Microseconds'|'Milliseconds'|'Bytes'|'Kilobytes'|'Megabytes'|'Gigabytes'
)
```

Parameters

- **MetricName** (*string*) – **[REQUIRED]**

The name of the metric.

- **Namespace** (*string*) – **[REQUIRED]**

The namespace of the metric.

- **Statistic** (*string*) – The statistic for the metric.
- **Dimensions** (*list*) – The list of dimensions associated with the metric.
 - (*dict*) –

The `Dimension` data type further expands on the identity of a metric using a Name, Value pair.

For examples that use one or more dimensions, see `PutMetricData` .

- * **Name** (*string*) – **[REQUIRED]**

The name of the dimension.

- * **Value** (*string*) – **[REQUIRED]**

The value representing the dimension measurement

- **Period** (*integer*) – The period in seconds over which the statistic is applied.
- **Unit** (*string*) – The unit for the metric.

Return type dict

Returns

Response Syntax

```
{
    'MetricAlarms': [
        {
            'AlarmName': 'string',
            'AlarmArn': 'string',
            'AlarmDescription': 'string',
            'AlarmConfigurationUpdatedTimestamp': datetime(2015, 1, 1),
            'ActionsEnabled': True|False,
            'OKActions': [
```



```

        'string',
    ],
    'AlarmActions': [
        'string',
    ],
    'InsufficientDataActions': [
        'string',
    ],
    'StateValue': 'OK'|'ALARM'|'INSUFFICIENT_DATA',
    'StateReason': 'string',
    'StateReasonData': 'string',
    'StateUpdatedTimestamp': datetime(2015, 1, 1),
    'MetricName': 'string',
    'Namespace': 'string',
    'Statistic': 'SampleCount'|'Average'|'Sum'|'Minimum'|'Maximum',
    'Dimensions': [
        {
            'Name': 'string',
            'Value': 'string'
        },
    ],
    'Period': 123,
    'Unit': 'Seconds'|'Microseconds'|'Milliseconds'|'Bytes'|'Kilobytes'|'M
    'EvaluationPeriods': 123,
    'Threshold': 123.0,
    'ComparisonOperator': 'GreaterThanOrEqualToThreshold'|'GreaterThanThre
    },
]
}

```

Response Structure

- (*dict*) –

The output for the DescribeAlarmsForMetric action.

- **MetricAlarms** (*list*) –

A list of information for each alarm with the specified metric.

- * (*dict*) –

The MetricAlarm data type represents an alarm. You can use PutMetricAlarm to create or update an alarm.

- **AlarmName** (*string*) –

The name of the alarm.

- **AlarmArn** (*string*) –

The Amazon Resource Name (ARN) of the alarm.

- **AlarmDescription** (*string*) –

The description for the alarm.

- **AlarmConfigurationUpdatedTimestamp** (*datetime*) –

The time stamp of the last update to the alarm configuration. Amazon CloudWatch uses Coordinated Universal Time (UTC) when returning time stamps, which do not accommodate seasonal adjustments such as daylight savings time. For more information, see [Time stamps](#) in the *Amazon CloudWatch Developer Guide*.

- **ActionsEnabled** (*boolean*) –

Indicates whether actions should be executed during any changes to the alarm's state.

- **OKActions** (*list*) –

The list of actions to execute when this alarm transitions into an OK state from any other state. Each action is specified as an Amazon Resource Number (ARN). Currently the only actions supported are publishing to an Amazon SNS topic and triggering an Auto Scaling policy.

- (*string*) –

- **AlarmActions** (*list*) –

The list of actions to execute when this alarm transitions into an ALARM state from any other state. Each action is specified as an Amazon Resource Number (ARN). Currently the only actions supported are publishing to an Amazon SNS topic and triggering an Auto Scaling policy.

- (*string*) –

- **InsufficientDataActions** (*list*) –

The list of actions to execute when this alarm transitions into an INSUFFICIENT_DATA state from any other state. Each action is specified as an Amazon Resource Number (ARN). Currently the only actions supported are publishing to an Amazon SNS topic or triggering an Auto Scaling policy.

Warning: The current WSDL lists this attribute as `UnknownActions`.

- (*string*) –

- **StateValue** (*string*) –

The state value for the alarm.

- **StateReason** (*string*) –

A human-readable explanation for the alarm's state.

- **StateReasonData** (*string*) –

An explanation for the alarm's state in machine-readable JSON format

- **StateUpdatedTimestamp** (*datetime*) –

The time stamp of the last update to the alarm's state. Amazon CloudWatch uses Coordinated Universal Time (UTC) when returning time stamps, which do not accommodate seasonal adjustments such as daylight savings time. For more information, see [Time stamps](#) in the *Amazon CloudWatch Developer Guide*.

- **MetricName** (*string*) –

The name of the alarm's metric.

- **Namespace** (*string*) –

The namespace of alarm's associated metric.

- **Statistic** (*string*) –

The statistic to apply to the alarm's associated metric.

- **Dimensions** (*list*) –

The list of dimensions associated with the alarm's associated metric.

- *(dict)* –

The `Dimension` data type further expands on the identity of a metric using a Name, Value pair.

For examples that use one or more dimensions, see `PutMetricData`.

- **Name** (*string*) –

The name of the dimension.

- **Value** (*string*) –

The value representing the dimension measurement

- **Period** (*integer*) –

The period in seconds over which the statistic is applied.

- **Unit** (*string*) –

The unit of the alarm's associated metric.

- **EvaluationPeriods** (*integer*) –

The number of periods over which data is compared to the specified threshold.

- **Threshold** (*float*) –

The value against which the specified statistic is compared.

- **ComparisonOperator** (*string*) –

The arithmetic operation to use when comparing the specified `Statistic` and `Threshold`. The specified `Statistic` value is used as the first operand.

disable_alarm_actions (***kwargs*)

Disables actions for the specified alarms. When an alarm's actions are disabled the alarm's state may change, but none of the alarm's actions will execute.

Request Syntax

```
response = client.disable_alarm_actions(
    AlarmNames=[
        'string',
    ]
)
```

Parameters `AlarmNames` (*list*) – [REQUIRED]

The names of the alarms to disable actions for.

- (*string*) –

Returns None

enable_alarm_actions (***kwargs*)

Enables actions for the specified alarms.

Request Syntax

```
response = client.enable_alarm_actions(
    AlarmNames=[
        'string',
    ]
)
```

```
]
)
```

Parameters `AlarmNames` (*list*) – [REQUIRED]

The names of the alarms to enable actions for.

- (*string*) –

Returns `None`**generate_presigned_url** (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url**get_metric_statistics** (***kwargs*)

Gets statistics for the specified metric.

The maximum number of data points returned from a single `GetMetricStatistics` request is 1,440, whereas the maximum number of data points that can be queried is 50,850. If you make a request that generates more than 1,440 data points, Amazon CloudWatch returns an error. In such a case, you can alter the request by narrowing the specified time range or increasing the specified period. Alternatively, you can make multiple requests across adjacent time ranges.

Amazon CloudWatch aggregates data points based on the length of the `period` that you specify. For example, if you request statistics with a one-minute granularity, Amazon CloudWatch aggregates data points with time stamps that fall within the same one-minute period. In such a case, the data points queried can greatly outnumber the data points returned.

The following examples show various statistics allowed by the data point query maximum of 50,850 when you call `GetMetricStatistics` on Amazon EC2 instances with detailed (one-minute) monitoring enabled:

- Statistics for up to 400 instances for a span of one hour
- Statistics for up to 35 instances over a span of 24 hours
- Statistics for up to 2 instances over a span of 2 weeks

For information about the namespace, metric names, and dimensions that other Amazon Web Services products use to send metrics to Cloudwatch, go to [Amazon CloudWatch Metrics, Namespaces, and Dimensions Reference](#) in the *Amazon CloudWatch Developer Guide*.

Request Syntax

```
response = client.get_metric_statistics(
    Namespace='string',
    MetricName='string',
    Dimensions=[
        {
            'Name': 'string',
            'Value': 'string'
        },
    ],
    StartTime=datetime(2015, 1, 1),
    EndTime=datetime(2015, 1, 1),
    Period=123,
```

```

Statistics=[
    'SampleCount' | 'Average' | 'Sum' | 'Minimum' | 'Maximum',
],
Unit='Seconds' | 'Microseconds' | 'Milliseconds' | 'Bytes' | 'Kilobytes' | 'Megabytes' | 'Gigabytes'
)

```

Parameters

- **Namespace** (*string*) – [REQUIRED]
The namespace of the metric, with or without spaces.
- **MetricName** (*string*) – [REQUIRED]
The name of the metric, with or without spaces.
- **Dimensions** (*list*) – A list of dimensions describing qualities of the metric.
 - (*dict*) –
The `Dimension` data type further expands on the identity of a metric using a Name, Value pair.
For examples that use one or more dimensions, see `PutMetricData` .
 - * **Name** (*string*) – [REQUIRED]
The name of the dimension.
 - * **Value** (*string*) – [REQUIRED]
The value representing the dimension measurement
- **StartTime** (*datetime*) – [REQUIRED]
The time stamp to use for determining the first datapoint to return. The value specified is inclusive; results include datapoints with the time stamp specified.
- **EndTime** (*datetime*) – [REQUIRED]
The time stamp to use for determining the last datapoint to return. The value specified is exclusive; results will include datapoints up to the time stamp specified.
- **Period** (*integer*) – [REQUIRED]
The granularity, in seconds, of the returned datapoints. `Period` must be at least 60 seconds and must be a multiple of 60. The default value is 60.
- **Statistics** (*list*) – [REQUIRED]
The metric statistics to return. For information about specific statistics returned by `GetMetricStatistics`, go to [Statistics](#) in the *Amazon CloudWatch Developer Guide* .

Valid Values: Average | Sum | SampleCount | Maximum | Minimum
– (*string*) –
- **Unit** (*string*) – The unit for the metric.

Return type dict

Returns

Response Syntax

```

{
    'Label': 'string',
    'Datapoints': [
        {
            'Timestamp': datetime(2015, 1, 1),
            'SampleCount': 123.0,

```

```
        'Average': 123.0,  
        'Sum': 123.0,  
        'Minimum': 123.0,  
        'Maximum': 123.0,  
        'Unit': 'Seconds'|'Microseconds'|'Milliseconds'|'Bytes'|'Kilobytes'|'M  
    },  
]  
}
```

Response Structure

- *(dict)* –

The output for the GetMetricStatistics action.

- **Label** (*string*) –

A label describing the specified metric.

- **Datapoints** (*list*) –

The datapoints for the specified metric.

- * *(dict)* –

The Datapoint data type encapsulates the statistical data that Amazon CloudWatch computes from metric data.

- **Timestamp** (*datetime*) –

The time stamp used for the datapoint. Amazon CloudWatch uses Coordinated Universal Time (UTC) when returning time stamps, which do not accommodate seasonal adjustments such as daylight savings time. For more information, see [Time stamps](#) in the *Amazon CloudWatch Developer Guide*.

- **SampleCount** (*float*) –

The number of metric values that contributed to the aggregate value of this datapoint.

- **Average** (*float*) –

The average of metric values that correspond to the datapoint.

- **Sum** (*float*) –

The sum of metric values used for the datapoint.

- **Minimum** (*float*) –

The minimum metric value used for the datapoint.

- **Maximum** (*float*) –

The maximum of the metric value used for the datapoint.

- **Unit** (*string*) –

The standard unit used for the datapoint.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type L{botocore.paginate.Paginator}

Returns A paginator object.

get_waiter (*waiter_name*)

list_metrics (**kwargs)

Returns a list of valid metrics stored for the AWS account owner. Returned metrics can be used with GetMetricStatistics to obtain statistical data for a given metric.

Request Syntax

```
response = client.list_metrics(
    Namespace='string',
    MetricName='string',
    Dimensions=[
        {
            'Name': 'string',
            'Value': 'string'
        },
    ],
    NextToken='string'
)
```

Parameters

- **Namespace** (*string*) – The namespace to filter against.
- **MetricName** (*string*) – The name of the metric to filter against.
- **Dimensions** (*list*) – A list of dimensions to filter against.
 - (*dict*) –

The DimensionFilter data type is used to filter ListMetrics results.

* **Name** (*string*) – **[REQUIRED]**

The dimension name to be matched.

* **Value** (*string*) –

The value of the dimension to be matched.

- **NextToken** (*string*) – The token returned by a previous call to indicate that there is more data available.

Return type dict

Returns

Response Syntax

```
{
    'Metrics': [
        {
            'Namespace': 'string',
            'MetricName': 'string',
            'Dimensions': [
                {
                    'Name': 'string',
                    'Value': 'string'
                },
            ],
        },
    ],
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

The output for the ListMetrics action.

- **Metrics** (*list*) –

A list of metrics used to generate statistics for an AWS account.

- * (*dict*) –

The `Metric` data type contains information about a specific metric. If you call `ListMetrics`, Amazon CloudWatch returns information contained by this data type.

The example in the Examples section publishes two metrics named `buffers` and `latency`. Both metrics are in the `examples` namespace. Both metrics have two dimensions, `InstanceID` and `InstanceType`.

- **Namespace** (*string*) –

The namespace of the metric.

- **MetricName** (*string*) –

The name of the metric.

- **Dimensions** (*list*) –

A list of dimensions associated with the metric.

- (*dict*) –

The `Dimension` data type further expands on the identity of a metric using a `Name`, `Value` pair.

For examples that use one or more dimensions, see `PutMetricData`.

- **Name** (*string*) –

The name of the dimension.

- **Value** (*string*) –

The value representing the dimension measurement

- **NextToken** (*string*) –

A string that marks the start of the next batch of returned results.

put_metric_alarm (***kwargs*)

Creates or updates an alarm and associates it with the specified Amazon CloudWatch metric. Optionally, this operation can associate one or more Amazon Simple Notification Service resources with the alarm.

When this operation creates an alarm, the alarm state is immediately set to `INSUFFICIENT_DATA`. The alarm is evaluated and its `StateValue` is set appropriately. Any actions associated with the `StateValue` is then executed.

Request Syntax

```
response = client.put_metric_alarm(
    AlarmName='string',
    AlarmDescription='string',
    ActionsEnabled=True|False,
    OKActions=[
        'string',
    ],
    AlarmActions=[
        'string',
    ],
    InsufficientDataActions=[
```



```

        'string',
    ],
    MetricName='string',
    Namespace='string',
    Statistic='SampleCount'|'Average'|'Sum'|'Minimum'|'Maximum',
    Dimensions=[
        {
            'Name': 'string',
            'Value': 'string'
        },
    ],
    Period=123,
    Unit='Seconds'|'Microseconds'|'Milliseconds'|'Bytes'|'Kilobytes'|'Megabytes'|'Gigabytes',
    EvaluationPeriods=123,
    Threshold=123.0,
    ComparisonOperator='GreaterThanOrEqualToThreshold'|'GreaterThanThreshold'|'LessThanThreshold'
)

```

Parameters

- **AlarmName** (*string*) – [REQUIRED]

The descriptive name for the alarm. This name must be unique within the user's AWS account

- **AlarmDescription** (*string*) – The description for the alarm.
- **ActionsEnabled** (*boolean*) – Indicates whether or not actions should be executed during any changes to the alarm's state.
- **OKActions** (*list*) – The list of actions to execute when this alarm transitions into an OK state from any other state. Each action is specified as an Amazon Resource Number (ARN). Currently the only action supported is publishing to an Amazon SNS topic or an Amazon Auto Scaling policy.

– (*string*) –

- **AlarmActions** (*list*) – The list of actions to execute when this alarm transitions into an ALARM state from any other state. Each action is specified as an Amazon Resource Number (ARN). Currently the only action supported is publishing to an Amazon SNS topic or an Amazon Auto Scaling policy.

– (*string*) –

- **InsufficientDataActions** (*list*) – The list of actions to execute when this alarm transitions into an INSUFFICIENT_DATA state from any other state. Each action is specified as an Amazon Resource Number (ARN). Currently the only action supported is publishing to an Amazon SNS topic or an Amazon Auto Scaling policy.

– (*string*) –

- **MetricName** (*string*) – [REQUIRED]

The name for the alarm's associated metric.

- **Namespace** (*string*) – [REQUIRED]

The namespace for the alarm's associated metric.

- **Statistic** (*string*) – [REQUIRED]

The statistic to apply to the alarm's associated metric.

- **Dimensions** (*list*) – The dimensions for the alarm's associated metric.

– (*dict*) –

The `Dimension` data type further expands on the identity of a metric using a Name, Value pair.

For examples that use one or more dimensions, see `PutMetricData`.

* **Name** (*string*) – [REQUIRED]

The name of the dimension.

* **Value** (*string*) – [REQUIRED]

The value representing the dimension measurement

• **Period** (*integer*) – [REQUIRED]

The period in seconds over which the specified statistic is applied.

• **Unit** (*string*) – The unit for the alarm's associated metric.

• **EvaluationPeriods** (*integer*) – [REQUIRED]

The number of periods over which data is compared to the specified threshold.

• **Threshold** (*float*) – [REQUIRED]

The value against which the specified statistic is compared.

• **ComparisonOperator** (*string*) – [REQUIRED]

The arithmetic operation to use when comparing the specified `Statistic` and `Threshold`. The specified `Statistic` value is used as the first operand.

Returns None

put_metric_data (***kwargs*)

Publishes metric data points to Amazon CloudWatch. Amazon Cloudwatch associates the data points with the specified metric. If the specified metric does not exist, Amazon CloudWatch creates the metric. It can take up to fifteen minutes for a new metric to appear in calls to the `ListMetrics` action.

The size of a `PutMetricDataRequest` is limited to 8 KB for HTTP GET requests and 40 KB for HTTP POST requests.

Warning: Although the `Value` parameter accepts numbers of type `Double`, Amazon CloudWatch truncates values with very large exponents. Values with base-10 exponents greater than 126 (1×10^{126}) are truncated. Likewise, values with base-10 exponents less than -130 (1×10^{-130}) are also truncated.

Data that is timestamped 24 hours or more in the past may take in excess of 48 hours to become available from submission time using `GetMetricStatistics`.

Request Syntax

```
response = client.put_metric_data(
    Namespace='string',
    MetricData=[
        {
            'MetricName': 'string',
            'Dimensions': [
                {
                    'Name': 'string',
                    'Value': 'string'
                },
            ],
            'Timestamp': datetime(2015, 1, 1),
            'Value': 123.0,
            'StatisticValues': {
                'SampleCount': 123.0,
                'Sum': 123.0,
                'Minimum': 123.0,
                'Maximum': 123.0
            },
            'Unit': 'Seconds'|'Microseconds'|'Milliseconds'|'Bytes'|'Kilobytes'|'Megabytes'
```

```
    },
  ]
)
```

Parameters

- **Namespace** (*string*) – [REQUIRED]

The namespace for the metric data.

- **MetricData** (*list*) – [REQUIRED]

A list of data describing the metric.

– (*dict*) –

The `MetricDatum` data type encapsulates the information sent with `PutMetricData` to either create a new metric or add new values to be aggregated into an existing metric.

- * **MetricName** (*string*) – [REQUIRED]

The name of the metric.

- * **Dimensions** (*list*) –

A list of dimensions associated with the metric. Note, when using the `Dimensions` value in a query, you need to append `.member.N` to it (e.g., `Dimensions.member.N`).

· (*dict*) –

The `Dimension` data type further expands on the identity of a metric using a `Name`, `Value` pair.

For examples that use one or more dimensions, see `PutMetricData`.

- **Name** (*string*) – [REQUIRED]

The name of the dimension.

- **Value** (*string*) – [REQUIRED]

The value representing the dimension measurement

- * **Timestamp** (*datetime*) –

The time stamp used for the metric. If not specified, the default value is set to the time the metric data was received. Amazon CloudWatch uses Coordinated Universal Time (UTC) when returning time stamps, which do not accommodate seasonal adjustments such as daylight savings time. For more information, see [Time stamps](#) in the *Amazon CloudWatch Developer Guide*.

- * **Value** (*float*) –

The value for the metric.

Warning: Although the `Value` parameter accepts numbers of type `Double`, Amazon CloudWatch truncates values with very large exponents. Values with base-10 exponents greater than 126 (1×10^{126}) are truncated. Likewise, values with base-10 exponents less than -130 (1×10^{-130}) are also truncated.

- * **StatisticValues** (*dict*) –

A set of statistical values describing the metric.

- **SampleCount** (*float*) – [REQUIRED]

The number of samples used for the statistic set.

- **Sum** (*float*) – [REQUIRED]
The sum of values for the sample set.
- **Minimum** (*float*) – [REQUIRED]
The minimum value of the sample set.
- **Maximum** (*float*) – [REQUIRED]
The maximum value of the sample set.
- * **Unit** (*string*) –
The unit of the metric.

Returns None

set_alarm_state (***kwargs*)

Temporarily sets the state of an alarm. When the updated `StateValue` differs from the previous value, the action configured for the appropriate state is invoked. This is not a permanent change. The next periodic alarm check (in about a minute) will set the alarm to its actual state.

Request Syntax

```
response = client.set_alarm_state(  
    AlarmName='string',  
    StateValue='OK' | 'ALARM' | 'INSUFFICIENT_DATA',  
    StateReason='string',  
    StateReasonData='string'  
)
```

Parameters

- **AlarmName** (*string*) – [REQUIRED]
The descriptive name for the alarm. This name must be unique within the user's AWS account. The maximum length is 255 characters.
- **StateValue** (*string*) – [REQUIRED]
The value of the state.
- **StateReason** (*string*) – [REQUIRED]
The reason that this alarm is set to this specific state (in human-readable text format)
- **StateReasonData** (*string*) – The reason that this alarm is set to this specific state (in machine-readable JSON format)

Returns None

Paginators

The available paginators are:

- `CloudWatch.Paginator.DescribeAlarmHistory`
- `CloudWatch.Paginator.DescribeAlarms`
- `CloudWatch.Paginator.ListMetrics`

class `CloudWatch.Paginator.DescribeAlarmHistory`

```
paginator = client.get_paginator('describe_alarm_history')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `CloudWatch.Client.describe_alarm_history()`.

Request Syntax

```
response_iterator = paginator.paginate(
    AlarmName='string',
    HistoryItemType='ConfigurationUpdate'|'StateUpdate'|'Action',
    StartDate=datetime(2015, 1, 1),
    EndDate=datetime(2015, 1, 1),
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **AlarmName** (*string*) – The name of the alarm.
- **HistoryItemType** (*string*) – The type of alarm histories to retrieve.
- **StartDate** (*datetime*) – The starting date to retrieve alarm history.
- **EndDate** (*datetime*) – The ending date to retrieve alarm history.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns**Response Syntax**

```
{
    'AlarmHistoryItems': [
        {
            'AlarmName': 'string',
            'Timestamp': datetime(2015, 1, 1),
            'HistoryItemType': 'ConfigurationUpdate'|'StateUpdate'|'Action',
            'HistorySummary': 'string',
            'HistoryData': 'string'
        },
    ],
}
```

Response Structure

- (*dict*) –
The output for the `DescribeAlarmHistory` action.

– **AlarmHistoryItems** (*list*) –

A list of alarm histories in JSON format.

* (*dict*) –

The `AlarmHistoryItem` data type contains descriptive information about the history of a specific alarm. If you call `DescribeAlarmHistory`, Amazon CloudWatch returns this data type as part of the `DescribeAlarmHistoryResult` data type.

· **AlarmName** (*string*) –

The descriptive name for the alarm.

· **Timestamp** (*datetime*) –

The time stamp for the alarm history item. Amazon CloudWatch uses Coordinated Universal Time (UTC) when returning time stamps, which do not accommodate seasonal adjustments such as daylight savings time. For more information, see [Time stamps](#) in the *Amazon CloudWatch Developer Guide*.

· **HistoryItemType** (*string*) –

The type of alarm history item.

· **HistorySummary** (*string*) –

A human-readable summary of the alarm history.

· **HistoryData** (*string*) –

Machine-readable data about the alarm in JSON format.

class `CloudWatch.Paginator.DescribeAlarms`

```
paginator = client.get_paginator('describe_alarms')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `CloudWatch.Client.describe_alarms()`.

Request Syntax

```
response_iterator = paginator.paginate(
    AlarmNames=[
        'string',
    ],
    AlarmNamePrefix='string',
    StateValue='OK'|'ALARM'|'INSUFFICIENT_DATA',
    ActionPrefix='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **AlarmNames** (*list*) – A list of alarm names to retrieve information for.
 - (*string*) –
- **AlarmNamePrefix** (*string*) – The alarm name prefix. `AlarmNames` cannot be specified if this parameter is specified.

- **StateValue** (*string*) – The state value to be used in matching alarms.
- **ActionPrefix** (*string*) – The action name prefix.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'MetricAlarms': [
    {
      'AlarmName': 'string',
      'AlarmArn': 'string',
      'AlarmDescription': 'string',
      'AlarmConfigurationUpdatedTimestamp': datetime(2015, 1, 1),
      'ActionsEnabled': True|False,
      'OKActions': [
        'string',
      ],
      'AlarmActions': [
        'string',
      ],
      'InsufficientDataActions': [
        'string',
      ],
      'StateValue': 'OK'|'ALARM'|'INSUFFICIENT_DATA',
      'StateReason': 'string',
      'StateReasonData': 'string',
      'StateUpdatedTimestamp': datetime(2015, 1, 1),
      'MetricName': 'string',
      'Namespace': 'string',
      'Statistic': 'SampleCount'|'Average'|'Sum'|'Minimum'|'Maximum',
      'Dimensions': [
        {
          'Name': 'string',
          'Value': 'string'
        },
      ],
      'Period': 123,
      'Unit': 'Seconds'|'Microseconds'|'Milliseconds'|'Bytes'|'Kilobytes'|'M
      'EvaluationPeriods': 123,
      'Threshold': 123.0,
      'ComparisonOperator': 'GreaterThanOrEqualToThreshold'|'GreaterThanThre
    ],
  ],
}
```

```
}
```

Response Structure

- (*dict*) –

The output for the DescribeAlarms action.

- **MetricAlarms** (*list*) –

A list of information for the specified alarms.

- * (*dict*) –

The MetricAlarm data type represents an alarm. You can use PutMetricAlarm to create or update an alarm.

- **AlarmName** (*string*) –

The name of the alarm.

- **AlarmArn** (*string*) –

The Amazon Resource Name (ARN) of the alarm.

- **AlarmDescription** (*string*) –

The description for the alarm.

- **AlarmConfigurationUpdatedTimestamp** (*datetime*) –

The time stamp of the last update to the alarm configuration. Amazon CloudWatch uses Coordinated Universal Time (UTC) when returning time stamps, which do not accommodate seasonal adjustments such as daylight savings time. For more information, see [Time stamps](#) in the *Amazon CloudWatch Developer Guide*.

- **ActionsEnabled** (*boolean*) –

Indicates whether actions should be executed during any changes to the alarm's state.

- **OKActions** (*list*) –

The list of actions to execute when this alarm transitions into an OK state from any other state. Each action is specified as an Amazon Resource Number (ARN). Currently the only actions supported are publishing to an Amazon SNS topic and triggering an Auto Scaling policy.

- (*string*) –

- **AlarmActions** (*list*) –

The list of actions to execute when this alarm transitions into an ALARM state from any other state. Each action is specified as an Amazon Resource Number (ARN). Currently the only actions supported are publishing to an Amazon SNS topic and triggering an Auto Scaling policy.

- (*string*) –

- **InsufficientDataActions** (*list*) –

The list of actions to execute when this alarm transitions into an INSUFFICIENT_DATA state from any other state. Each action is specified as an Amazon Resource Number (ARN). Currently the only actions supported are publishing to an Amazon SNS topic or triggering an Auto Scaling policy.

Warning: The current WSDL lists this attribute as `UnknownActions`.

- *(string)* –
- **StateValue** (*string*) –
The state value for the alarm.
- **StateReason** (*string*) –
A human-readable explanation for the alarm’s state.
- **StateReasonData** (*string*) –
An explanation for the alarm’s state in machine-readable JSON format
- **StateUpdatedTimestamp** (*datetime*) –
The time stamp of the last update to the alarm’s state. Amazon CloudWatch uses Coordinated Universal Time (UTC) when returning time stamps, which do not accommodate seasonal adjustments such as daylight savings time. For more information, see [Time stamps](#) in the *Amazon CloudWatch Developer Guide*.
- **MetricName** (*string*) –
The name of the alarm’s metric.
- **Namespace** (*string*) –
The namespace of alarm’s associated metric.
- **Statistic** (*string*) –
The statistic to apply to the alarm’s associated metric.
- **Dimensions** (*list*) –
The list of dimensions associated with the alarm’s associated metric.
- (*dict*) –
The `Dimension` data type further expands on the identity of a metric using a Name, Value pair.

For examples that use one or more dimensions, see `PutMetricData`.
- **Name** (*string*) –
The name of the dimension.
- **Value** (*string*) –
The value representing the dimension measurement
- **Period** (*integer*) –
The period in seconds over which the statistic is applied.
- **Unit** (*string*) –
The unit of the alarm’s associated metric.
- **EvaluationPeriods** (*integer*) –
The number of periods over which data is compared to the specified threshold.
- **Threshold** (*float*) –
The value against which the specified statistic is compared.

- **ComparisonOperator** (*string*) –

The arithmetic operation to use when comparing the specified `Statistic` and `Threshold`. The specified `Statistic` value is used as the first operand.

class `CloudWatch.Paginator.ListMetrics`

```
paginator = client.get_paginator('list_metrics')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `CloudWatch.Client.list_metrics()`.

Request Syntax

```
response_iterator = paginator.paginate(
    Namespace='string',
    MetricName='string',
    Dimensions=[
        {
            'Name': 'string',
            'Value': 'string'
        },
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **Namespace** (*string*) – The namespace to filter against.
- **MetricName** (*string*) – The name of the metric to filter against.
- **Dimensions** (*list*) – A list of dimensions to filter against.
 - (*dict*) –

The `DimensionFilter` data type is used to filter `ListMetrics` results.

- * **Name** (*string*) – **[REQUIRED]**

The dimension name to be matched.

- * **Value** (*string*) –

The value of the dimension to be matched.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```

{
    'Metrics': [
        {
            'Namespace': 'string',
            'MetricName': 'string',
            'Dimensions': [
                {
                    'Name': 'string',
                    'Value': 'string'
                },
            ],
        },
    ],
}

```

Response Structure

- (*dict*) –

The output for the ListMetrics action.

- **Metrics** (*list*) –

A list of metrics used to generate statistics for an AWS account.

- * (*dict*) –

The `Metric` data type contains information about a specific metric. If you call `ListMetrics`, Amazon CloudWatch returns information contained by this data type.

The example in the Examples section publishes two metrics named `buffers` and `latency`. Both metrics are in the examples namespace. Both metrics have two dimensions, `InstanceID` and `InstanceType`.

- **Namespace** (*string*) –

The namespace of the metric.

- **MetricName** (*string*) –

The name of the metric.

- **Dimensions** (*list*) –

A list of dimensions associated with the metric.

- (*dict*) –

The `Dimension` data type further expands on the identity of a metric using a `Name`, `Value` pair.

For examples that use one or more dimensions, see `PutMetricData`.

- **Name** (*string*) –

The name of the dimension.

- **Value** (*string*) –

The value representing the dimension measurement

CodeCommit

Table of Contents

- CodeCommit
 - Client

Client

class `CodeCommit.Client`

A low-level client representing AWS CodeCommit:

```
import boto3
```

```
client = boto3.client('codecommit')
```

These are the available methods:

- `batch_get_repositories()`
- `can_paginate()`
- `create_branch()`
- `create_repository()`
- `delete_repository()`
- `generate_presigned_url()`
- `get_branch()`
- `get_paginator()`
- `get_repository()`
- `get_waiter()`
- `list_branches()`
- `list_repositories()`
- `update_default_branch()`
- `update_repository_description()`
- `update_repository_name()`

`batch_get_repositories` (***kwargs*)

Gets information about one or more repositories.

Note: The description field for a repository accepts all HTML characters and all valid Unicode characters. Applications that do not HTML-encode the description and display it in a web page could expose users to potentially malicious code. Make sure that you HTML-encode the description field in any application that uses this API to display the repository description on a web page.

Request Syntax

```
response = client.batch_get_repositories(
    repositoryNames=[
        'string',
    ]
)
```

Parameters `repositoryNames` (*list*) – [REQUIRED]

The names of the repositories to get information about.

- (*string*) – Repository name is restricted to alphanumeric characters (a-z, A-Z, 0-9), ".", "_", and "-". Additionally, the suffix ".git" is prohibited in a repository name.

Return type dict

Returns

Response Syntax

```
{
  'repositories': [
    {
      'accountId': 'string',
      'repositoryId': 'string',
      'repositoryName': 'string',
      'repositoryDescription': 'string',
      'defaultBranch': 'string',
      'lastModifiedDate': datetime(2015, 1, 1),
      'creationDate': datetime(2015, 1, 1),
      'cloneUrlHttp': 'string',
      'cloneUrlSsh': 'string',
      'Arn': 'string'
    },
  ],
  'repositoriesNotFound': [
    'string',
  ]
}
```

Response Structure

- (*dict*) –

Represents the output of a batch get repositories operation.

- **repositories** (*list*) –

A list of repositories returned by the batch get repositories operation.

- * (*dict*) –

Information about a repository.

- **accountId** (*string*) –
The ID of the AWS account associated with the repository.
- **repositoryId** (*string*) –
The ID of the repository.
- **repositoryName** (*string*) –
The repository's name.
- **repositoryDescription** (*string*) –
A comment or description about the repository.
- **defaultBranch** (*string*) –
The repository's default branch name.
- **lastModifiedDate** (*datetime*) –
The date and time the repository was last modified, in timestamp format.
- **creationDate** (*datetime*) –
The date and time the repository was created, in timestamp format.

- **cloneUrlHttp** (*string*) –
The URL to use for cloning the repository over HTTPS.
- **cloneUrlSsh** (*string*) –
The URL to use for cloning the repository over SSH.
- **Arn** (*string*) –
The Amazon Resource Name (ARN) of the repository.

– **repositoriesNotFound** (*list*) –

Returns a list of repository names for which information could not be found.

- * (*string*) – Repository name is restricted to alphanumeric characters (a-z, A-Z, 0-9), ””, “_”, and “-”. Additionally, the suffix “.git” is prohibited in a repository name.

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you’d normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

create_branch (***kwargs*)

Creates a new branch in a repository and points the branch to a commit.

Note: Calling the create branch operation does not set a repository’s default branch. To do this, call the update default branch operation.

Request Syntax

```
response = client.create_branch(  
    repositoryName='string',  
    branchName='string',  
    commitId='string'  
)
```

Parameters

- **repositoryName** (*string*) – [REQUIRED]
The name of the repository in which you want to create the new branch.
- **branchName** (*string*) – [REQUIRED]
The name of the new branch to create.
- **commitId** (*string*) – [REQUIRED]
The ID of the commit to point the new branch to.

Note: If this commit ID is not specified, the new branch will point to the commit that is pointed to by the repository’s default branch.

Returns None

create_repository (***kwargs*)

Creates a new, empty repository.

Request Syntax

```
response = client.create_repository(
    repositoryName='string',
    repositoryDescription='string'
)
```

Parameters

- **repositoryName** (*string*) – [REQUIRED]

The name of the new repository to be created.

Note: The repository name must be unique across the calling AWS account. In addition, repository names are restricted to alphanumeric characters. The suffix “.git” is prohibited.

- **repositoryDescription** (*string*) – A comment or description about the new repository.

Return type dict

Returns

Response Syntax

```
{
  'repositoryMetadata': {
    'accountId': 'string',
    'repositoryId': 'string',
    'repositoryName': 'string',
    'repositoryDescription': 'string',
    'defaultBranch': 'string',
    'lastModifiedDate': datetime(2015, 1, 1),
    'creationDate': datetime(2015, 1, 1),
    'cloneUrlHttp': 'string',
    'cloneUrlSsh': 'string',
    'Arn': 'string'
  }
}
```

Response Structure

- (*dict*) –

Represents the output of a create repository operation.

- **repositoryMetadata** (*dict*) –

Information about the newly created repository.

- * **accountId** (*string*) –

The ID of the AWS account associated with the repository.

- * **repositoryId** (*string*) –

The ID of the repository.

- * **repositoryName** (*string*) –

The repository’s name.

- * **repositoryDescription** (*string*) –

A comment or description about the repository.

- * **defaultBranch** (*string*) –

The repository’s default branch name.

- * **lastModifiedDate** (*datetime*) –

The date and time the repository was last modified, in timestamp format.

* **creationDate** (*datetime*) –

The date and time the repository was created, in timestamp format.

* **cloneUrlHttp** (*string*) –

The URL to use for cloning the repository over HTTPS.

* **cloneUrlSsh** (*string*) –

The URL to use for cloning the repository over SSH.

* **Arn** (*string*) –

The Amazon Resource Name (ARN) of the repository.

delete_repository (***kwargs*)

Deletes a repository. If a specified repository was already deleted, a null repository ID will be returned.

Warning: Deleting a repository also deletes all associated objects and metadata. After a repository is deleted, all future push calls to the deleted repository will fail.

Request Syntax

```
response = client.delete_repository(  
    repositoryName='string'  
)
```

Parameters **repositoryName** (*string*) – [REQUIRED]

The name of the repository to delete.

Return type dict

Returns

Response Syntax

```
{  
    'repositoryId': 'string'  
}
```

Response Structure

- (*dict*) –

Represents the output of a delete repository operation.

- **repositoryId** (*string*) –

The ID of the repository that was deleted.

generate_presigned_url (*ClientMethod, Params=None, ExpiresIn=3600, HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to *ClientMethod*.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_branch (***kwargs*)

Retrieves information about a repository branch, including its name and the last commit ID.

Request Syntax

```
response = client.get_branch(
    repositoryName='string',
    branchName='string'
)
```

Parameters

- **repositoryName** (*string*) – Repository name is restricted to alphanumeric characters (a-z, A-Z, 0-9), ".", "_", and "-". Additionally, the suffix ".git" is prohibited in a repository name.
- **branchName** (*string*) – The name of the branch for which you want to retrieve information.

Return type dict

Returns**Response Syntax**

```
{
    'branch': {
        'branchName': 'string',
        'commitId': 'string'
    }
}
```

Response Structure

- (*dict*) –

Represents the output of a get branch operation.

- **branch** (*dict*) –

The name of the branch.

- * **branchName** (*string*) –

The name of the branch.

- * **commitId** (*string*) –

The ID of the last commit made to the branch.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type L{botocore.paginate.Paginator}

Returns A paginator object.

get_repository (***kwargs*)

Gets information about a repository.

Note: The description field for a repository accepts all HTML characters and all valid Unicode characters. Applications that do not HTML-encode the description and display it in a web page could expose users

to potentially malicious code. Make sure that you HTML-encode the description field in any application that uses this API to display the repository description on a web page.

Request Syntax

```
response = client.get_repository(  
    repositoryName='string'  
)
```

Parameters `repositoryName` (*string*) – **[REQUIRED]**

The name of the repository to get information about.

Return type dict

Returns

Response Syntax

```
{  
    'repositoryMetadata': {  
        'accountId': 'string',  
        'repositoryId': 'string',  
        'repositoryName': 'string',  
        'repositoryDescription': 'string',  
        'defaultBranch': 'string',  
        'lastModifiedDate': datetime(2015, 1, 1),  
        'creationDate': datetime(2015, 1, 1),  
        'cloneUrlHttp': 'string',  
        'cloneUrlSsh': 'string',  
        'Arn': 'string'  
    }  
}
```

Response Structure

- (*dict*) –

Represents the output of a get repository operation.

- **repositoryMetadata** (*dict*) –

Information about the repository.

- * **accountId** (*string*) –

The ID of the AWS account associated with the repository.

- * **repositoryId** (*string*) –

The ID of the repository.

- * **repositoryName** (*string*) –

The repository's name.

- * **repositoryDescription** (*string*) –

A comment or description about the repository.

- * **defaultBranch** (*string*) –

The repository's default branch name.

- * **lastModifiedDate** (*datetime*) –

The date and time the repository was last modified, in timestamp format.

- * **creationDate** (*datetime*) –

The date and time the repository was created, in timestamp format.

- * **cloneUrlHttp** (*string*) –
The URL to use for cloning the repository over HTTPS.
- * **cloneUrlSsh** (*string*) –
The URL to use for cloning the repository over SSH.
- * **Arn** (*string*) –
The Amazon Resource Name (ARN) of the repository.

get_waiter (*waiter_name*)

list_branches (***kwargs*)

Gets information about one or more branches in a repository.

Request Syntax

```
response = client.list_branches(
    repositoryName='string',
    nextToken='string'
)
```

Parameters

- **repositoryName** (*string*) – [REQUIRED]
The name of the repository that contains the branches.
- **nextToken** (*string*) – An enumeration token that allows the operation to batch the results.

Return type dict

Returns

Response Syntax

```
{
    'branches': [
        'string',
    ],
    'nextToken': 'string'
}
```

Response Structure

- (*dict*) –
Represents the output of a list branches operation.
 - **branches** (*list*) –
The list of branch names.
 - * (*string*) –
 - **nextToken** (*string*) –
An enumeration token that returns the batch of the results.

list_repositories (***kwargs*)

Gets information about one or more repositories.

Request Syntax

```
response = client.list_repositories(
    nextToken='string',
    sortBy='repositoryName' | 'lastModifiedDate',
```

```
order='ascending'|'descending'  
)
```

Parameters

- **nextToken** (*string*) – An enumeration token that allows the operation to batch the results of the operation. Batch sizes are 1,000 for list repository operations. When the client sends the token back to AWS CodeCommit, another page of 1,000 records is retrieved.
- **sortBy** (*string*) – The criteria used to sort the results of a list repositories operation.
- **order** (*string*) – The order in which to sort the results of a list repositories operation.

Return type dict**Returns****Response Syntax**

```
{  
    'repositories': [  
        {  
            'repositoryName': 'string',  
            'repositoryId': 'string'  
        },  
    ],  
    'nextToken': 'string'  
}
```

Response Structure

- (*dict*) –

Represents the output of a list repositories operation.

- **repositories** (*list*) –

Lists the repositories called by the list repositories operation.

- * (*dict*) –

Information about a repository name and ID.

- **repositoryName** (*string*) – Repository name is restricted to alphanumeric characters (a-z, A-Z, 0-9), “.”, “_”, and “-”. Additionally, the suffix “.git” is prohibited in a repository name.
- **repositoryId** (*string*) –

The ID associated with the repository name.

- **nextToken** (*string*) –

An enumeration token that allows the operation to batch the results of the operation. Batch sizes are 1,000 for list repository operations. When the client sends the token back to AWS CodeCommit, another page of 1,000 records is retrieved.

update_default_branch (***kwargs*)

Sets or changes the default branch name for the specified repository.

Note: If you use this operation to change the default branch name to the current default branch name, a success message is returned even though the default branch did not change.

Request Syntax

```
response = client.update_default_branch(
    repositoryName='string',
    defaultBranchName='string'
)
```

Parameters

- **repositoryName** (*string*) – [REQUIRED]
The name of the repository to set or change the default branch for.
- **defaultBranchName** (*string*) – [REQUIRED]
The name of the branch to set as the default.

Returns None

update_repository_description (***kwargs*)

Sets or changes the comment or description for a repository.

Note: The description field for a repository accepts all HTML characters and all valid Unicode characters. Applications that do not HTML-encode the description and display it in a web page could expose users to potentially malicious code. Make sure that you HTML-encode the description field in any application that uses this API to display the repository description on a web page.

Request Syntax

```
response = client.update_repository_description(
    repositoryName='string',
    repositoryDescription='string'
)
```

Parameters

- **repositoryName** (*string*) – [REQUIRED]
The name of the repository to set or change the comment or description for.
- **repositoryDescription** (*string*) – The new comment or description for the specified repository.

Returns None

update_repository_name (***kwargs*)

Renames a repository.

Request Syntax

```
response = client.update_repository_name(
    oldName='string',
    newName='string'
)
```

Parameters

- **oldName** (*string*) – [REQUIRED] Repository name is restricted to alphanumeric characters (a-z, A-Z, 0-9), ".", "_", and "-". Additionally, the suffix ".git" is prohibited in a repository name.
- **newName** (*string*) – [REQUIRED] Repository name is restricted to alphanumeric characters (a-z, A-Z, 0-9), ".", "_", and "-". Additionally, the suffix ".git" is prohibited in a repository name.

Returns None

CodeDeploy

Table of Contents

- CodeDeploy
 - Client

Client

class CodeDeploy.**Client**

A low-level client representing AWS CodeDeploy:

```
import boto3
```

```
client = boto3.client('codedeploy')
```

These are the available methods:

- `add_tags_to_on_premises_instances()`
- `batch_get_applications()`
- `batch_get_deployments()`
- `batch_get_on_premises_instances()`
- `can_paginate()`
- `create_application()`
- `create_deployment()`
- `create_deployment_config()`
- `create_deployment_group()`
- `delete_application()`
- `delete_deployment_config()`
- `delete_deployment_group()`
- `deregister_on_premises_instance()`
- `generate_presigned_url()`
- `get_application()`
- `get_application_revision()`
- `get_deployment()`
- `get_deployment_config()`
- `get_deployment_group()`
- `get_deployment_instance()`
- `get_on_premises_instance()`
- `get_paginator()`
- `get_waiter()`
- `list_application_revisions()`
- `list_applications()`
- `list_deployment_configs()`
- `list_deployment_groups()`
- `list_deployment_instances()`
- `list_deployments()`
- `list_on_premises_instances()`
- `register_application_revision()`
- `register_on_premises_instance()`
- `remove_tags_from_on_premises_instances()`
- `stop_deployment()`
- `update_application()`

• `update_deployment_group()`
add_tags_to_on_premises_instances (***kwargs*)
 Adds tags to on-premises instances.

Request Syntax

```
response = client.add_tags_to_on_premises_instances(
    tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
    instanceNames=[
        'string',
    ]
)
```

Parameters

- **tags** (*list*) – [REQUIRED]

The tag key-value pairs to add to the on-premises instances.

Keys and values are both required. Keys cannot be nulls or empty strings. Value-only tags are not allowed.

– (*dict*) –

Information about a tag.

* **Key** (*string*) –

The tag’s key.

* **Value** (*string*) –

The tag’s value.

- **instanceNames** (*list*) – [REQUIRED]

The names of the on-premises instances to add tags to.

– (*string*) –

Returns None

batch_get_applications (***kwargs*)
 Gets information about one or more applications.

Request Syntax

```
response = client.batch_get_applications(
    applicationNames=[
        'string',
    ]
)
```

Parameters **applicationNames** (*list*) – A list of application names, with multiple application names separated by spaces.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'applicationsInfo': [
    {
      'applicationId': 'string',
      'applicationName': 'string',
      'createTime': datetime(2015, 1, 1),
      'linkedToGitHub': True|False
    },
  ],
}
```

Response Structure

- (*dict*) –

Represents the output of a batch get applications operation.

- **applicationsInfo** (*list*) –

Information about the applications.

- * (*dict*) –

Information about an application.

- **applicationId** (*string*) –

The application ID.

- **applicationName** (*string*) –

The application name.

- **createTime** (*datetime*) –

The time that the application was created.

- **linkedToGitHub** (*boolean*) –

True if the user has authenticated with GitHub for the specified application; otherwise, false.

batch_get_deployments (**kwargs)

Gets information about one or more deployments.

Request Syntax

```
response = client.batch_get_deployments(
    deploymentIds=[
        'string',
    ]
)
```

Parameters **deploymentIds** (*list*) – A list of deployment IDs, with multiple deployment IDs separated by spaces.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'deploymentsInfo': [
    {
      'applicationName': 'string',
      'deploymentGroupName': 'string',
      'deploymentConfigName': 'string',
    },
  ],
}
```



```

'deploymentId': 'string',
'revision': {
    'revisionType': 'S3'|'GitHub',
    's3Location': {
        'bucket': 'string',
        'key': 'string',
        'bundleType': 'tar'|'tgz'|'zip',
        'version': 'string',
        'eTag': 'string'
    },
    'gitHubLocation': {
        'repository': 'string',
        'commitId': 'string'
    }
},
'status': 'Created'|'Queued'|'InProgress'|'Succeeded'|'Failed'|'Stopped',
'errorInformation': {
    'code': 'DEPLOYMENT_GROUP_MISSING'|'APPLICATION_MISSING'|'REVISION_MISSING',
    'message': 'string'
},
'createTime': datetime(2015, 1, 1),
'startTime': datetime(2015, 1, 1),
'completeTime': datetime(2015, 1, 1),
'deploymentOverview': {
    'Pending': 123,
    'InProgress': 123,
    'Succeeded': 123,
    'Failed': 123,
    'Skipped': 123
},
'description': 'string',
'creator': 'user'|'autoscaling',
'ignoreApplicationStopFailures': True|False
}
]
}

```

Response Structure

- *(dict)* –

Represents the output of a batch get deployments operation.

- **deploymentsInfo** (*list*) –

Information about the deployments.

- * *(dict)* –

Information about a deployment.

- **applicationName** (*string*) –

The application name.

- **deploymentGroupName** (*string*) –

The deployment group name.

- **deploymentConfigName** (*string*) –

The deployment configuration name.

- **deploymentId** (*string*) –

The deployment ID.

- **revision** (*dict*) –

Information about the location of application artifacts that are stored and the service to retrieve them from.

- **revisionType** (*string*) –

The application revision's type:

- S3: An application revision stored in Amazon S3.
- GitHub: An application revision stored in GitHub.

- **s3Location** (*dict*) –

Information about the location of application artifacts that are stored in Amazon S3.

- **bucket** (*string*) –

The name of the Amazon S3 bucket where the application revision is stored.

- **key** (*string*) –

The name of the Amazon S3 object that represents the bundled artifacts for the application revision.

- **bundleType** (*string*) –

The file type of the application revision. Must be one of the following:

- tar: A tar archive file.
- tgz: A compressed tar archive file.
- zip: A zip archive file.

- **version** (*string*) –

A specific version of the Amazon S3 object that represents the bundled artifacts for the application revision.

If the version is not specified, the system will use the most recent version by default.

- **eTag** (*string*) –

The ETag of the Amazon S3 object that represents the bundled artifacts for the application revision.

If the ETag is not specified as an input parameter, ETag validation of the object will be skipped.

- **gitHubLocation** (*dict*) –

Information about the location of application artifacts that are stored in GitHub.

- **repository** (*string*) –

The GitHub account and repository pair that stores a reference to the commit that represents the bundled artifacts for the application revision.

Specified as account/repository.

- **commitId** (*string*) –

The SHA1 commit ID of the GitHub commit that references the that represents the bundled artifacts for the application revision.

- **status** (*string*) –

The current state of the deployment as a whole.

- **errorInformation** (*dict*) –

Information about any error associated with this deployment.

- **code** (*string*) –

The error code:

- **APPLICATION_MISSING**: The application was missing. Note that this error code will most likely be raised if the application is deleted after the deployment is created but before it starts.
- **DEPLOYMENT_GROUP_MISSING**: The deployment group was missing. Note that this error code will most likely be raised if the deployment group is deleted after the deployment is created but before it starts.
- **HEALTH_CONSTRAINTS**: The deployment failed on too many instances to be able to successfully deploy within the specified instance health constraints.
- **HEALTH_CONSTRAINTS_INVALID**: The revision can never successfully deploy within the instance health constraints as specified.
- **IAM_ROLE_MISSING**: The service role cannot be accessed.
- **IAM_ROLE_PERMISSIONS**: The service role does not have the correct permissions.
- **INTERNAL_ERROR**: There was an internal error.
- **NO_EC2_SUBSCRIPTION**: The calling account is not subscribed to the Amazon EC2 service.
- **NO_INSTANCES**: No instances were specified, or no instances can be found.
- **OVER_MAX_INSTANCES**: The maximum number of instances was exceeded.
- **THROTTLED**: The operation was throttled because the calling account exceeded the throttling limits of one or more AWS services.
- **TIMEOUT**: The deployment has timed out.
- **REVISION_MISSING**: The revision ID was missing. Note that this error code will most likely be raised if the revision is deleted after the deployment is created but before it starts.

- **message** (*string*) –

An accompanying error message.

- **createTime** (*datetime*) –

A timestamp indicating when the deployment was created.

- **startTime** (*datetime*) –

A timestamp indicating when the deployment began deploying to the deployment group.

Note that in some cases, the reported value of the start time may be later than the complete time. This is due to differences in the clock settings of various back-end servers that participate in the overall deployment process.

- **completeTime** (*datetime*) –

A timestamp indicating when the deployment was completed.

- **deploymentOverview** (*dict*) –

A summary of the deployment status of the instances in the deployment.

- **Pending** (*integer*) –
The number of instances that are pending in the deployment.
- **InProgress** (*integer*) –
The number of instances that are in progress in the deployment.
- **Succeeded** (*integer*) –
The number of instances that have succeeded in the deployment.
- **Failed** (*integer*) –
The number of instances that have failed in the deployment.
- **Skipped** (*integer*) –
The number of instances that have been skipped in the deployment.
- **description** (*string*) –
A comment about the deployment.
- **creator** (*string*) –
How the deployment was created:
 - user: A user created the deployment.
 - autoscaling: Auto Scaling created the deployment.
- **ignoreApplicationStopFailures** (*boolean*) –
If true, then if the deployment causes the ApplicationStop deployment lifecycle event to fail to a specific instance, the deployment will not be considered to have failed to that instance at that point and will continue on to the BeforeInstall deployment lifecycle event.

If false or not specified, then if the deployment causes the ApplicationStop deployment lifecycle event to fail to a specific instance, the deployment will stop to that instance, and the deployment to that instance will be considered to have failed.

batch_get_on_premises_instances (***kwargs*)
Gets information about one or more on-premises instances.

Request Syntax

```
response = client.batch_get_on_premises_instances(  
    instanceNames=[  
        'string',  
    ]  
)
```

Parameters **instanceNames** (*list*) – The names of the on-premises instances to get information about.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{  
    'instanceInfos': [  
        {  
            'name': 'string',  
            'type': 'string',  
            'description': 'string',  
            'creator': 'string',  
            'ignoreApplicationStopFailures': boolean,  
            'pending': integer,  
            'inProgress': integer,  
            'succeeded': integer,  
            'failed': integer,  
            'skipped': integer,  
            'tags': {  
                'key': 'string',  
                'value': 'string',  
            },  
        },  
    ],  
}
```

```

    {
        'instanceName': 'string',
        'iamUserArn': 'string',
        'instanceArn': 'string',
        'registerTime': datetime(2015, 1, 1),
        'deregisterTime': datetime(2015, 1, 1),
        'tags': [
            {
                'Key': 'string',
                'Value': 'string'
            },
        ]
    },
]
}

```

Response Structure

- (*dict*) –

Represents the output of a batch get on-premises instances operation.

- **instanceInfos** (*list*) –

Information about the on-premises instances.

- * (*dict*) –

Information about an on-premises instance.

- **instanceName** (*string*) –
The name of the on-premises instance.
- **iamUserArn** (*string*) –
The IAM user ARN associated with the on-premises instance.
- **instanceArn** (*string*) –
The ARN of the on-premises instance.
- **registerTime** (*datetime*) –
The time that the on-premises instance was registered.
- **deregisterTime** (*datetime*) –
If the on-premises instance was deregistered, the time that the on-premises instance was deregistered.
- **tags** (*list*) –
The tags that are currently associated with the on-premises instance.
- (*dict*) –
Information about a tag.
- **Key** (*string*) –
The tag's key.
- **Value** (*string*) –
The tag's value.

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`,

if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

create_application (**kwargs)

Creates a new application.

Request Syntax

```
response = client.create_application(
    applicationName='string'
)
```

Parameters `applicationName` (*string*) – [REQUIRED]

The name of the application. This name must be unique with the applicable IAM user or AWS account.

Return type dict

Returns

Response Syntax

```
{
    'applicationId': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a create application operation.

- **applicationId** (*string*) –

A unique application ID.

create_deployment (**kwargs)

Deploys an application revision through the specified deployment group.

Request Syntax

```
response = client.create_deployment(
    applicationName='string',
    deploymentGroupName='string',
    revision={
        'revisionType': 'S3'|'GitHub',
        's3Location': {
            'bucket': 'string',
            'key': 'string',
            'bundleType': 'tar'|'tgz'|'zip',
            'version': 'string',
            'eTag': 'string'
        },
        'gitHubLocation': {
            'repository': 'string',
            'commitId': 'string'
        }
    },
    deploymentConfigName='string',
    description='string',
    ignoreApplicationStopFailures=True|False
)
```

Parameters

- **applicationName** (*string*) – [REQUIRED]

The name of an existing AWS CodeDeploy application associated with the applicable IAM user or AWS account.

- **deploymentGroupName** (*string*) – The deployment group’s name.
- **revision** (*dict*) – The type of revision to deploy, along with information about the revision’s location.
 - **revisionType** (*string*) –

The application revision’s type:

- * **S3**: An application revision stored in Amazon S3.
- * **GitHub**: An application revision stored in GitHub.

- **s3Location** (*dict*) –

Information about the location of application artifacts that are stored in Amazon S3.

- * **bucket** (*string*) –

The name of the Amazon S3 bucket where the application revision is stored.

- * **key** (*string*) –

The name of the Amazon S3 object that represents the bundled artifacts for the application revision.

- * **bundleType** (*string*) –

The file type of the application revision. Must be one of the following:

- **tar**: A tar archive file.
- **tgz**: A compressed tar archive file.
- **zip**: A zip archive file.

- * **version** (*string*) –

A specific version of the Amazon S3 object that represents the bundled artifacts for the application revision.

If the version is not specified, the system will use the most recent version by default.

- * **eTag** (*string*) –

The ETag of the Amazon S3 object that represents the bundled artifacts for the application revision.

If the ETag is not specified as an input parameter, ETag validation of the object will be skipped.

- **gitHubLocation** (*dict*) –

Information about the location of application artifacts that are stored in GitHub.

- * **repository** (*string*) –

The GitHub account and repository pair that stores a reference to the commit that represents the bundled artifacts for the application revision.

Specified as account/repository.

- * **commitId** (*string*) –

The SHA1 commit ID of the GitHub commit that references the that represents the bundled artifacts for the application revision.

- **deploymentConfigName** (*string*) – The name of an existing deployment configuration associated with the applicable IAM user or AWS account.

If not specified, the value configured in the deployment group will be used as the default. If the deployment group does not have a deployment configuration associated with it, then `CodeDeployDefault.OneAtATime` will be used by default.

- **description** (*string*) – A comment about the deployment.
- **ignoreApplicationStopFailures** (*boolean*) – If set to true, then if the deployment causes the `ApplicationStop` deployment lifecycle event to fail to a specific instance, the deployment will not be considered to have failed to that instance at that point and will continue on to the `BeforeInstall` deployment lifecycle event.

If set to false or not specified, then if the deployment causes the `ApplicationStop` deployment lifecycle event to fail to a specific instance, the deployment will stop to that instance, and the deployment to that instance will be considered to have failed.

Return type dict

Returns

Response Syntax

```
{
    'deploymentId': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a create deployment operation.

- **deploymentId** (*string*) –

A unique deployment ID.

create_deployment_config (***kwargs*)

Creates a new deployment configuration.

Request Syntax

```
response = client.create_deployment_config(
    deploymentConfigName='string',
    minimumHealthyHosts={
        'value': 123,
        'type': 'HOST_COUNT' | 'FLEET_PERCENT'
    }
)
```

Parameters

- **deploymentConfigName** (*string*) – [REQUIRED]

The name of the deployment configuration to create.

- **minimumHealthyHosts** (*dict*) – The minimum number of healthy instances that should be available at any time during the deployment. There are two parameters expected in the input: type and value.

The type parameter takes either of the following values:

- `HOST_COUNT`: The value parameter represents the minimum number of healthy instances, as an absolute value.
- `FLEET_PERCENT`: The value parameter represents the minimum number of healthy instances, as a percentage of the total number of instances in

the deployment. If you specify `FLEET_PERCENT`, then at the start of the deployment AWS CodeDeploy converts the percentage to the equivalent number of instances and rounds fractional instances up.

The value parameter takes an integer.

For example, to set a minimum of 95% healthy instances, specify a type of `FLEET_PERCENT` and a value of 95.

– **value** (*integer*) –

The minimum healthy instances value.

– **type** (*string*) –

The minimum healthy instances type:

* `HOST_COUNT`: The minimum number of healthy instances, as an absolute value.

* `FLEET_PERCENT`: The minimum number of healthy instances, as a percentage of the total number of instances in the deployment.

For example, for 9 instances, if a `HOST_COUNT` of 6 is specified, deploy to up to 3 instances at a time. The deployment succeeds if 6 or more instances are successfully deployed to; otherwise, the deployment fails. If a `FLEET_PERCENT` of 40 is specified, deploy to up to 5 instances at a time. The deployment succeeds if 4 or more instances are successfully deployed to; otherwise, the deployment fails.

Note: In a call to the get deployment configuration operation, `CodeDeployDefault.OneAtATime` will return a minimum healthy instances type of `MOST_CONCURRENCY` and a value of 1. This means a deployment to only one instances at a time. (You cannot set the type to `MOST_CONCURRENCY`, only to `HOST_COUNT` or `FLEET_PERCENT`.)

Return type dict

Returns

Response Syntax

```
{
    'deploymentConfigId': 'string'
}
```

Response Structure

• (*dict*) –

Represents the output of a create deployment configuration operation.

– **deploymentConfigId** (*string*) –

A unique deployment configuration ID.

create_deployment_group (***kwargs*)

Creates a new deployment group for application revisions to be deployed to.

Request Syntax

```
response = client.create_deployment_group(
    applicationName='string',
    deploymentGroupName='string',
    deploymentConfigName='string',
    ec2TagFilters=[
        {
```

```
        'Key': 'string',
        'Value': 'string',
        'Type': 'KEY_ONLY' | 'VALUE_ONLY' | 'KEY_AND_VALUE'
    },
],
onPremisesInstanceTagFilters=[
    {
        'Key': 'string',
        'Value': 'string',
        'Type': 'KEY_ONLY' | 'VALUE_ONLY' | 'KEY_AND_VALUE'
    },
],
autoScalingGroups=[
    'string',
],
serviceRoleArn='string'
)
```

Parameters

- **applicationName** (*string*) – **[REQUIRED]**

The name of an existing AWS CodeDeploy application associated with the applicable IAM user or AWS account.

- **deploymentGroupName** (*string*) – **[REQUIRED]**

The name of an existing deployment group for the specified application.

- **deploymentConfigName** (*string*) – If specified, the deployment configuration name must be one of the predefined values, or it can be a custom deployment configuration:

- **CodeDeployDefault.AllAtOnce** deploys an application revision to up to all of the instances at once. The overall deployment succeeds if the application revision deploys to at least one of the instances. The overall deployment fails after the application revision fails to deploy to all of the instances. For example, for 9 instances, deploy to up to all 9 instances at once. The overall deployment succeeds if any of the 9 instances is successfully deployed to, and it fails if all 9 instances fail to be deployed to.
- **CodeDeployDefault.HalfAtATime** deploys to up to half of the instances at a time (with fractions rounded down). The overall deployment succeeds if the application revision deploys to at least half of the instances (with fractions rounded up); otherwise, the deployment fails. For example, for 9 instances, deploy to up to 4 instances at a time. The overall deployment succeeds if 5 or more instances are successfully deployed to; otherwise, the deployment fails. Note that the deployment may successfully deploy to some instances, even if the overall deployment fails.
- **CodeDeployDefault.OneAtATime** deploys the application revision to only one of the instances at a time. The overall deployment succeeds if the application revision deploys to all of the instances. The overall deployment fails after the application revision first fails to deploy to any one instances. For example, for 9 instances, deploy to one instance at a time. The overall deployment succeeds if all 9 instances are successfully deployed to, and it fails if any of one of the 9 instances fail to be deployed to. Note that the deployment may successfully deploy to some instances, even if the overall deployment fails. This is the default deployment configuration if a configuration isn't specified for either the deployment or the deployment group.

To create a custom deployment configuration, call the create deployment config-

uration operation.

- **ec2TagFilters** (*list*) – The Amazon EC2 tags to filter on.
– (*dict*) –

Information about a tag filter.

- * **Key** (*string*) –

The tag filter key.

- * **Value** (*string*) –

The tag filter value.

- * **Type** (*string*) –

The tag filter type:

- KEY_ONLY: Key only.
- VALUE_ONLY: Value only.
- KEY_AND_VALUE: Key and value.

- **onPremisesInstanceTagFilters** (*list*) – The on-premises instance tags to filter on.
– (*dict*) –

Information about an on-premises instance tag filter.

- * **Key** (*string*) –

The on-premises instance tag filter key.

- * **Value** (*string*) –

The on-premises instance tag filter value.

- * **Type** (*string*) –

The on-premises instance tag filter type:

- KEY_ONLY: Key only.
- VALUE_ONLY: Value only.
- KEY_AND_VALUE: Key and value.

- **autoScalingGroups** (*list*) – A list of associated Auto Scaling groups.
– (*string*) –

- **serviceRoleArn** (*string*) – **[REQUIRED]**

A service role ARN that allows AWS CodeDeploy to act on the user's behalf when interacting with AWS services.

Return type dict

Returns

Response Syntax

```
{
    'deploymentGroupId': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a create deployment group operation.

- **deploymentGroupId** (*string*) –

A unique deployment group ID.

delete_application (***kwargs*)

Deletes an application.

Request Syntax

```
response = client.delete_application(
    applicationName='string'
)
```

Parameters `applicationName` (*string*) – [REQUIRED]

The name of an existing AWS CodeDeploy application associated with the applicable IAM user or AWS account.

Returns None

`delete_deployment_config` (***kwargs*)
Deletes a deployment configuration.

Note: A deployment configuration cannot be deleted if it is currently in use. Also, predefined configurations cannot be deleted.

Request Syntax

```
response = client.delete_deployment_config(
    deploymentConfigName='string'
)
```

Parameters `deploymentConfigName` (*string*) – [REQUIRED]

The name of an existing deployment configuration associated with the applicable IAM user or AWS account.

Returns None

`delete_deployment_group` (***kwargs*)
Deletes a deployment group.

Request Syntax

```
response = client.delete_deployment_group(
    applicationName='string',
    deploymentGroupName='string'
)
```

Parameters

- **`applicationName`** (*string*) – [REQUIRED]

The name of an existing AWS CodeDeploy application associated with the applicable IAM user or AWS account.

- **`deploymentGroupName`** (*string*) – [REQUIRED]

The name of an existing deployment group for the specified application.

Return type dict

Returns

Response Syntax

```
{
    'hooksNotCleanedUp': [
        {
            'name': 'string',
            'hook': 'string'
        },
    ],
}
```

```
    ]
}
```

Response Structure

- *(dict)* –

Represents the output of a delete deployment group operation.

- **hooksNotCleanedUp** (*list*) –

If the output contains no data, and the corresponding deployment group contained at least one Auto Scaling group, AWS CodeDeploy successfully removed all corresponding Auto Scaling lifecycle event hooks from the Amazon EC2 instances in the Auto Scaling. If the output does contain data, AWS CodeDeploy could not remove some Auto Scaling lifecycle event hooks from the Amazon EC2 instances in the Auto Scaling group.

- * *(dict)* –

Information about an Auto Scaling group.

- **name** (*string*) –

The Auto Scaling group name.

- **hook** (*string*) –

An Auto Scaling lifecycle event hook name.

deregister_on_premises_instance (***kwargs*)

Deregisters an on-premises instance.

Request Syntax

```
response = client.deregister_on_premises_instance(
    instanceName='string'
)
```

Parameters **instanceName** (*string*) – [REQUIRED]

The name of the on-premises instance to deregister.

Returns None

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to *ClientMethod*.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_application (***kwargs*)

Gets information about an application.

Request Syntax

```
response = client.get_application(
    applicationName='string'
)
```

Parameters `applicationName` (*string*) – **[REQUIRED]**

The name of an existing AWS CodeDeploy application associated with the applicable IAM user or AWS account.

Return type dict**Returns****Response Syntax**

```
{
  'application': {
    'applicationId': 'string',
    'applicationName': 'string',
    'createTime': datetime(2015, 1, 1),
    'linkedToGitHub': True|False
  }
}
```

Response Structure

- (*dict*) –

Represents the output of a get application operation.

- **application** (*dict*) –

Information about the application.

- * **applicationId** (*string*) –

The application ID.

- * **applicationName** (*string*) –

The application name.

- * **createTime** (*datetime*) –

The time that the application was created.

- * **linkedToGitHub** (*boolean*) –

True if the user has authenticated with GitHub for the specified application; otherwise, false.

get_application_revision (***kwargs*)

Gets information about an application revision.

Request Syntax

```
response = client.get_application_revision(
    applicationName='string',
    revision={
        'revisionType': 'S3'|'GitHub',
        's3Location': {
            'bucket': 'string',
            'key': 'string',
            'bundleType': 'tar'|'tgz'|'zip',
            'version': 'string',
            'eTag': 'string'
        },
        'gitHubLocation': {
            'repository': 'string',
            'commitId': 'string'
        }
    }
)
```

Parameters

- **applicationName** (*string*) – **[REQUIRED]**

The name of the application that corresponds to the revision.

- **revision** (*dict*) – **[REQUIRED]**

Information about the application revision to get, including the revision's type and its location.

- **revisionType** (*string*) –

The application revision's type:

- * **S3**: An application revision stored in Amazon S3.
- * **GitHub**: An application revision stored in GitHub.

- **s3Location** (*dict*) –

Information about the location of application artifacts that are stored in Amazon S3.

- * **bucket** (*string*) –

The name of the Amazon S3 bucket where the application revision is stored.

- * **key** (*string*) –

The name of the Amazon S3 object that represents the bundled artifacts for the application revision.

- * **bundleType** (*string*) –

The file type of the application revision. Must be one of the following:

- **tar**: A tar archive file.
- **tgz**: A compressed tar archive file.
- **zip**: A zip archive file.

- * **version** (*string*) –

A specific version of the Amazon S3 object that represents the bundled artifacts for the application revision.

If the version is not specified, the system will use the most recent version by default.

- * **eTag** (*string*) –

The ETag of the Amazon S3 object that represents the bundled artifacts for the application revision.

If the ETag is not specified as an input parameter, ETag validation of the object will be skipped.

- **gitHubLocation** (*dict*) –

Information about the location of application artifacts that are stored in GitHub.

- * **repository** (*string*) –

The GitHub account and repository pair that stores a reference to the commit that represents the bundled artifacts for the application revision.

Specified as account/repository.

- * **commitId** (*string*) –

The SHA1 commit ID of the GitHub commit that references the that represents the bundled artifacts for the application revision.

Return type dict

Returns

Response Syntax

```
{
  'applicationName': 'string',
  'revision': {
    'revisionType': 'S3'|'GitHub',
    's3Location': {
      'bucket': 'string',
      'key': 'string',
      'bundleType': 'tar'|'tgz'|'zip',
      'version': 'string',
      'eTag': 'string'
    },
    'gitHubLocation': {
      'repository': 'string',
      'commitId': 'string'
    }
  },
  'revisionInfo': {
    'description': 'string',
    'deploymentGroups': [
      'string',
    ],
    'firstUsedTime': datetime(2015, 1, 1),
    'lastUsedTime': datetime(2015, 1, 1),
    'registerTime': datetime(2015, 1, 1)
  }
}
```

Response Structure

- (*dict*) –

Represents the output of a get application revision operation.

- **applicationName** (*string*) –

The name of the application that corresponds to the revision.

- **revision** (*dict*) –

Additional information about the revision, including the revision’s type and its location.

- * **revisionType** (*string*) –

The application revision’s type:

- S3: An application revision stored in Amazon S3.
- GitHub: An application revision stored in GitHub.

- * **s3Location** (*dict*) –

Information about the location of application artifacts that are stored in Amazon S3.

- **bucket** (*string*) –

The name of the Amazon S3 bucket where the application revision is stored.

- **key** (*string*) –

The name of the Amazon S3 object that represents the bundled artifacts for the application revision.

- **bundleType** (*string*) –

The file type of the application revision. Must be one of the following:

- tar: A tar archive file.
- tgz: A compressed tar archive file.
- zip: A zip archive file.

- **version** (*string*) –

A specific version of the Amazon S3 object that represents the bundled artifacts for the application revision.

If the version is not specified, the system will use the most recent version by default.

- **eTag** (*string*) –

The ETag of the Amazon S3 object that represents the bundled artifacts for the application revision.

If the ETag is not specified as an input parameter, ETag validation of the object will be skipped.

- * **gitHubLocation** (*dict*) –

Information about the location of application artifacts that are stored in GitHub.

- **repository** (*string*) –

The GitHub account and repository pair that stores a reference to the commit that represents the bundled artifacts for the application revision.

Specified as account/repository.

- **commitId** (*string*) –

The SHA1 commit ID of the GitHub commit that references the that represents the bundled artifacts for the application revision.

- **revisionInfo** (*dict*) –

General information about the revision.

- * **description** (*string*) –

A comment about the revision.

- * **deploymentGroups** (*list*) –

A list of deployment groups that use this revision.

- (*string*) –

- * **firstUsedTime** (*datetime*) –

When the revision was first used by AWS CodeDeploy.

- * **lastUsedTime** (*datetime*) –

When the revision was last used by AWS CodeDeploy.

- * **registerTime** (*datetime*) –

When the revision was registered with AWS CodeDeploy.

get_deployment (***kwargs*)

Gets information about a deployment.

Request Syntax

```
response = client.get_deployment(
    deploymentId='string'
)
```

Parameters `deploymentId` (*string*) – **[REQUIRED]**

An existing deployment ID associated with the applicable IAM user or AWS account.

Return type dict

Returns

Response Syntax

```
{
    'deploymentInfo': {
        'applicationName': 'string',
        'deploymentGroupName': 'string',
        'deploymentConfigName': 'string',
        'deploymentId': 'string',
        'revision': {
            'revisionType': 'S3'|'GitHub',
            's3Location': {
                'bucket': 'string',
                'key': 'string',
                'bundleType': 'tar'|'tgz'|'zip',
                'version': 'string',
                'eTag': 'string'
            },
            'gitHubLocation': {
                'repository': 'string',
                'commitId': 'string'
            }
        },
        'status': 'Created'|'Queued'|'InProgress'|'Succeeded'|'Failed'|'Stopped',
        'errorInformation': {
            'code': 'DEPLOYMENT_GROUP_MISSING'|'APPLICATION_MISSING'|'REVISION_MIS',
            'message': 'string'
        },
        'createTime': datetime(2015, 1, 1),
        'startTime': datetime(2015, 1, 1),
        'completeTime': datetime(2015, 1, 1),
        'deploymentOverview': {
            'Pending': 123,
            'InProgress': 123,
            'Succeeded': 123,
            'Failed': 123,
            'Skipped': 123
        },
        'description': 'string',
        'creator': 'user'|'autoscaling',
        'ignoreApplicationStopFailures': True|False
    }
}
```

Response Structure

- (*dict*) –

Represents the output of a get deployment operation.

- **deploymentInfo** (*dict*) –

Information about the deployment.

- * **applicationName** (*string*) –

The application name.

- * **deploymentGroupName** (*string*) –

The deployment group name.

- * **deploymentConfigName** (*string*) –

The deployment configuration name.

- * **deploymentId** (*string*) –

The deployment ID.

- * **revision** (*dict*) –

Information about the location of application artifacts that are stored and the service to retrieve them from.

- **revisionType** (*string*) –

The application revision's type:

- S3: An application revision stored in Amazon S3.
- GitHub: An application revision stored in GitHub.

- **s3Location** (*dict*) –

Information about the location of application artifacts that are stored in Amazon S3.

- **bucket** (*string*) –

The name of the Amazon S3 bucket where the application revision is stored.

- **key** (*string*) –

The name of the Amazon S3 object that represents the bundled artifacts for the application revision.

- **bundleType** (*string*) –

The file type of the application revision. Must be one of the following:

- tar: A tar archive file.
- tgz: A compressed tar archive file.
- zip: A zip archive file.

- **version** (*string*) –

A specific version of the Amazon S3 object that represents the bundled artifacts for the application revision.

If the version is not specified, the system will use the most recent version by default.

- **eTag** (*string*) –

The ETag of the Amazon S3 object that represents the bundled artifacts for the application revision.

If the ETag is not specified as an input parameter, ETag validation of the object will be skipped.

- **githubLocation** (*dict*) –

Information about the location of application artifacts that are stored in GitHub.

- **repository** (*string*) –

The GitHub account and repository pair that stores a reference to the commit that represents the bundled artifacts for the application revision.

Specified as account/repository.

- **commitId** (*string*) –

The SHA1 commit ID of the GitHub commit that references the that represents the bundled artifacts for the application revision.

- * **status** (*string*) –

The current state of the deployment as a whole.

- * **errorInformation** (*dict*) –

Information about any error associated with this deployment.

- **code** (*string*) –

The error code:

- **APPLICATION_MISSING**: The application was missing. Note that this error code will most likely be raised if the application is deleted after the deployment is created but before it starts.
- **DEPLOYMENT_GROUP_MISSING**: The deployment group was missing. Note that this error code will most likely be raised if the deployment group is deleted after the deployment is created but before it starts.
- **HEALTH_CONSTRAINTS**: The deployment failed on too many instances to be able to successfully deploy within the specified instance health constraints.
- **HEALTH_CONSTRAINTS_INVALID**: The revision can never successfully deploy within the instance health constraints as specified.
- **IAM_ROLE_MISSING**: The service role cannot be accessed.
- **IAM_ROLE_PERMISSIONS**: The service role does not have the correct permissions.
- **INTERNAL_ERROR**: There was an internal error.
- **NO_EC2_SUBSCRIPTION**: The calling account is not subscribed to the Amazon EC2 service.
- **NO_INSTANCES**: No instances were specified, or no instances can be found.
- **OVER_MAX_INSTANCES**: The maximum number of instances was exceeded.
- **THROTTLED**: The operation was throttled because the calling account exceeded the throttling limits of one or more AWS services.
- **TIMEOUT**: The deployment has timed out.
- **REVISION_MISSING**: The revision ID was missing. Note that this error code will most likely be raised if the revision is deleted after the deployment is created but before it starts.
- **message** (*string*) –

An accompanying error message.

- * **createTime** (*datetime*) –

A timestamp indicating when the deployment was created.

- * **startTime** (*datetime*) –

A timestamp indicating when the deployment began deploying to the deployment group.

Note that in some cases, the reported value of the start time may be later than the complete time. This is due to differences in the clock settings of various back-end servers that participate in the overall deployment process.

* **completeTime** (*datetime*) –

A timestamp indicating when the deployment was completed.

* **deploymentOverview** (*dict*) –

A summary of the deployment status of the instances in the deployment.

- **Pending** (*integer*) –

The number of instances that are pending in the deployment.

- **InProgress** (*integer*) –

The number of instances that are in progress in the deployment.

- **Succeeded** (*integer*) –

The number of instances that have succeeded in the deployment.

- **Failed** (*integer*) –

The number of instances that have failed in the deployment.

- **Skipped** (*integer*) –

The number of instances that have been skipped in the deployment.

* **description** (*string*) –

A comment about the deployment.

* **creator** (*string*) –

How the deployment was created:

- **user**: A user created the deployment.
- **autoscaling**: Auto Scaling created the deployment.

* **ignoreApplicationStopFailures** (*boolean*) –

If true, then if the deployment causes the ApplicationStop deployment lifecycle event to fail to a specific instance, the deployment will not be considered to have failed to that instance at that point and will continue on to the BeforeInstall deployment lifecycle event.

If false or not specified, then if the deployment causes the ApplicationStop deployment lifecycle event to fail to a specific instance, the deployment will stop to that instance, and the deployment to that instance will be considered to have failed.

get_deployment_config (***kwargs*)

Gets information about a deployment configuration.

Request Syntax

```
response = client.get_deployment_config(
    deploymentConfigName='string'
)
```

Parameters `deploymentConfigName` (*string*) – **[REQUIRED]**

The name of an existing deployment configuration associated with the applicable IAM user or AWS account.

Return type dict

Returns

Response Syntax

```
{
  'deploymentConfigInfo': {
    'deploymentConfigId': 'string',
    'deploymentConfigName': 'string',
    'minimumHealthyHosts': {
      'value': 123,
      'type': 'HOST_COUNT' | 'FLEET_PERCENT'
    },
    'createTime': datetime(2015, 1, 1)
  }
}
```

Response Structure

- (*dict*) –

Represents the output of a get deployment configuration operation.

- **deploymentConfigInfo** (*dict*) –

Information about the deployment configuration.

- * **deploymentConfigId** (*string*) –

The deployment configuration ID.

- * **deploymentConfigName** (*string*) –

The deployment configuration name.

- * **minimumHealthyHosts** (*dict*) –

Information about the number or percentage of minimum healthy instances.

- **value** (*integer*) –

The minimum healthy instances value.

- **type** (*string*) –

The minimum healthy instances type:

- **HOST_COUNT**: The minimum number of healthy instances, as an absolute value.

- **FLEET_PERCENT**: The minimum number of healthy instances, as a percentage of the total number of instances in the deployment.

For example, for 9 instances, if a **HOST_COUNT** of 6 is specified, deploy to up to 3 instances at a time. The deployment succeeds if 6 or more instances are successfully deployed to; otherwise, the deployment fails. If a **FLEET_PERCENT** of 40 is specified, deploy to up to 5 instances at a time. The deployment succeeds if 4 or more instances are successfully deployed to; otherwise, the deployment fails.

Note: In a call to the get deployment configuration operation, `CodeDeployDefault.OneAtATime` will return a minimum healthy instances type of `MOST_CONCURRENCY` and a

value of 1. This means a deployment to only one instances at a time. (You cannot set the type to MOST_CONCURRENCY, only to HOST_COUNT or FLEET_PERCENT.)

* **createTime** (*datetime*) –

The time that the deployment configuration was created.

get_deployment_group (***kwargs*)

Gets information about a deployment group.

Request Syntax

```
response = client.get_deployment_group(
    applicationName='string',
    deploymentGroupName='string'
)
```

Parameters

- **applicationName** (*string*) – **[REQUIRED]**

The name of an existing AWS CodeDeploy application associated with the applicable IAM user or AWS account.

- **deploymentGroupName** (*string*) – **[REQUIRED]**

The name of an existing deployment group for the specified application.

Return type dict

Returns

Response Syntax

```
{
  'deploymentGroupInfo': {
    'applicationName': 'string',
    'deploymentGroupId': 'string',
    'deploymentGroupName': 'string',
    'deploymentConfigName': 'string',
    'ec2TagFilters': [
      {
        'Key': 'string',
        'Value': 'string',
        'Type': 'KEY_ONLY' | 'VALUE_ONLY' | 'KEY_AND_VALUE'
      },
    ],
    'onPremisesInstanceTagFilters': [
      {
        'Key': 'string',
        'Value': 'string',
        'Type': 'KEY_ONLY' | 'VALUE_ONLY' | 'KEY_AND_VALUE'
      },
    ],
    'autoScalingGroups': [
      {
        'name': 'string',
        'hook': 'string'
      },
    ],
    'serviceRoleArn': 'string',
    'targetRevision': {
      'revisionType': 'S3' | 'GitHub',
```

```
        's3Location': {
            'bucket': 'string',
            'key': 'string',
            'bundleType': 'tar'|'tgz'|'zip',
            'version': 'string',
            'eTag': 'string'
        },
        'gitHubLocation': {
            'repository': 'string',
            'commitId': 'string'
        }
    }
}
```

Response Structure

- (*dict*) –

Represents the output of a get deployment group operation.

- **deploymentGroupInfo** (*dict*) –

Information about the deployment group.

- * **applicationName** (*string*) –

The application name.

- * **deploymentGroupId** (*string*) –

The deployment group ID.

- * **deploymentGroupName** (*string*) –

The deployment group name.

- * **deploymentConfigName** (*string*) –

The deployment configuration name.

- * **ec2TagFilters** (*list*) –

The Amazon EC2 tags to filter on.

- (*dict*) –

Information about a tag filter.

- **Key** (*string*) –

The tag filter key.

- **Value** (*string*) –

The tag filter value.

- **Type** (*string*) –

The tag filter type:

- KEY_ONLY: Key only.
- VALUE_ONLY: Value only.
- KEY_AND_VALUE: Key and value.

- * **onPremisesInstanceTagFilters** (*list*) –

The on-premises instance tags to filter on.

- (*dict*) –

Information about an on-premises instance tag filter.

- **Key** (*string*) –

The on-premises instance tag filter key.

- **Value** (*string*) –
The on-premises instance tag filter value.
- **Type** (*string*) –
The on-premises instance tag filter type:
 - KEY_ONLY: Key only.
 - VALUE_ONLY: Value only.
 - KEY_AND_VALUE: Key and value.
- * **autoScalingGroups** (*list*) –
A list of associated Auto Scaling groups.
 - (*dict*) –
Information about an Auto Scaling group.
 - **name** (*string*) –
The Auto Scaling group name.
 - **hook** (*string*) –
An Auto Scaling lifecycle event hook name.
- * **serviceRoleArn** (*string*) –
A service role ARN.
- * **targetRevision** (*dict*) –
Information about the deployment group's target revision, including the revision's type and its location.
 - **revisionType** (*string*) –
The application revision's type:
 - S3: An application revision stored in Amazon S3.
 - GitHub: An application revision stored in GitHub.
 - **s3Location** (*dict*) –
Information about the location of application artifacts that are stored in Amazon S3.
 - **bucket** (*string*) –
The name of the Amazon S3 bucket where the application revision is stored.
 - **key** (*string*) –
The name of the Amazon S3 object that represents the bundled artifacts for the application revision.
 - **bundleType** (*string*) –
The file type of the application revision. Must be one of the following:
 - tar: A tar archive file.
 - tgz: A compressed tar archive file.
 - zip: A zip archive file.
 - **version** (*string*) –
A specific version of the Amazon S3 object that represents the bundled artifacts for the application revision.

If the version is not specified, the system will use the most recent version by default.
 - **eTag** (*string*) –

The ETag of the Amazon S3 object that represents the bundled artifacts for the application revision.

If the ETag is not specified as an input parameter, ETag validation of the object will be skipped.

- **gitHubLocation** (*dict*) –

Information about the location of application artifacts that are stored in GitHub.

- **repository** (*string*) –

The GitHub account and repository pair that stores a reference to the commit that represents the bundled artifacts for the application revision.

Specified as account/repository.

- **commitId** (*string*) –

The SHA1 commit ID of the GitHub commit that references the that represents the bundled artifacts for the application revision.

get_deployment_instance (***kwargs*)

Gets information about an instance as part of a deployment.

Request Syntax

```
response = client.get_deployment_instance(
    deploymentId='string',
    instanceId='string'
)
```

Parameters

- **deploymentId** (*string*) – **[REQUIRED]**

The unique ID of a deployment.

- **instanceId** (*string*) – **[REQUIRED]**

The unique ID of an instance in the deployment's deployment group.

Return type dict

Returns

Response Syntax

```
{
  'instanceSummary': {
    'deploymentId': 'string',
    'instanceId': 'string',
    'status': 'Pending'|'InProgress'|'Succeeded'|'Failed'|'Skipped'|'Unknown',
    'lastUpdatedAt': datetime(2015, 1, 1),
    'lifecycleEvents': [
      {
        'lifecycleEventName': 'string',
        'diagnostics': {
          'errorCode': 'Success'|'ScriptMissing'|'ScriptNotExecutable'|
          'scriptName': 'string',
          'message': 'string',
          'logTail': 'string'
        }
      },
      'startTime': datetime(2015, 1, 1),

```

```

        'endTime': datetime(2015, 1, 1),
        'status': 'Pending'|'InProgress'|'Succeeded'|'Failed'|'Skipped'|'Unknown'
    },
]
}

```

Response Structure

- (*dict*) –

Represents the output of a get deployment instance operation.

- **instanceSummary** (*dict*) –

Information about the instance.

- * **deploymentId** (*string*) –

The deployment ID.

- * **instanceId** (*string*) –

The instance ID.

- * **status** (*string*) –

The deployment status for this instance:

- Pending: The deployment is pending for this instance.
- In Progress: The deployment is in progress for this instance.
- Succeeded: The deployment has succeeded for this instance.
- Failed: The deployment has failed for this instance.
- Skipped: The deployment has been skipped for this instance.
- Unknown: The deployment status is unknown for this instance.

- * **lastUpdatedAt** (*datetime*) –

A timestamp indicating when the instance information was last updated.

- * **lifecycleEvents** (*list*) –

A list of lifecycle events for this instance.

- (*dict*) –

Information about a deployment lifecycle event.

- **lifecycleEventName** (*string*) –

The deployment lifecycle event name, such as Application-Stop, BeforeInstall, AfterInstall, ApplicationStart, or ValidateService.

- **diagnostics** (*dict*) –

Diagnostic information about the deployment lifecycle event.

- **errorCode** (*string*) –

The associated error code:

- Success: The specified script ran.
- ScriptMissing: The specified script was not found in the specified location.
- ScriptNotExecutable: The specified script is not a recognized executable file type.
- ScriptTimedOut: The specified script did not finish running in the specified time period.
- ScriptFailed: The specified script failed to run as expected.

- **UnknownError**: The specified script did not run for an unknown reason.
- **scriptName** (*string*) –
The name of the script.
- **message** (*string*) –
The message associated with the error.
- **logTail** (*string*) –
The last portion of the associated diagnostic log.
- **startTime** (*datetime*) –
A timestamp indicating when the deployment lifecycle event started.
- **endTime** (*datetime*) –
A timestamp indicating when the deployment lifecycle event ended.
- **status** (*string*) –
The deployment lifecycle event status:
 - Pending: The deployment lifecycle event is pending.
 - InProgress: The deployment lifecycle event is in progress.
 - Succeeded: The deployment lifecycle event has succeeded.
 - Failed: The deployment lifecycle event has failed.
 - Skipped: The deployment lifecycle event has been skipped.
 - Unknown: The deployment lifecycle event is unknown.

get_on_premises_instance (***kwargs*)

Gets information about an on-premises instance.

Request Syntax

```
response = client.get_on_premises_instance(  
    instanceName='string'  
)
```

Parameters **instanceName** (*string*) – **[REQUIRED]**

The name of the on-premises instance to get information about

Return type dict

Returns

Response Syntax

```
{  
    'instanceInfo': {  
        'instanceName': 'string',  
        'iamUserArn': 'string',  
        'instanceArn': 'string',  
        'registerTime': datetime(2015, 1, 1),  
        'deregisterTime': datetime(2015, 1, 1),  
        'tags': [  
            {  
                'Key': 'string',  
                'Value': 'string'  
            },  
        ],  
    },  
}
```

```
    }
}
```

Response Structure

- (*dict*) –

Represents the output of a get on-premises instance operation.

- **instanceInfo** (*dict*) –

Information about the on-premises instance.

- * **instanceName** (*string*) –

The name of the on-premises instance.

- * **iamUserArn** (*string*) –

The IAM user ARN associated with the on-premises instance.

- * **instanceArn** (*string*) –

The ARN of the on-premises instance.

- * **registerTime** (*datetime*) –

The time that the on-premises instance was registered.

- * **deregisterTime** (*datetime*) –

If the on-premises instance was deregistered, the time that the on-premises instance was deregistered.

- * **tags** (*list*) –

The tags that are currently associated with the on-premises instance.

- (*dict*) –

Information about a tag.

- **Key** (*string*) –

The tag's key.

- **Value** (*string*) –

The tag's value.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type L{botocore.paginate.Paginator}

Returns A paginator object.

get_waiter (*waiter_name*)

list_application_revisions (***kwargs*)

Lists information about revisions for an application.

Request Syntax

```
response = client.list_application_revisions(
    applicationName='string',
    sortBy='registerTime'|'firstUsedTime'|'lastUsedTime',
```

```
sortOrder='ascending'|'descending',
s3Bucket='string',
s3KeyPrefix='string',
deployed='include'|'exclude'|'ignore',
nextToken='string'
)
```

Parameters

- **applicationName** (*string*) – **[REQUIRED]**

The name of an existing AWS CodeDeploy application associated with the applicable IAM user or AWS account.

- **sortBy** (*string*) – The column name to sort the list results by:
 - registerTime: Sort the list results by when the revisions were registered with AWS CodeDeploy.
 - firstUsedTime: Sort the list results by when the revisions were first used by in a deployment.
 - lastUsedTime: Sort the list results by when the revisions were last used in a deployment.

If not specified or set to null, the results will be returned in an arbitrary order.

- **sortOrder** (*string*) – The order to sort the list results by:
 - ascending: Sort the list of results in ascending order.
 - descending: Sort the list of results in descending order.

If not specified, the results will be sorted in ascending order.

If set to null, the results will be sorted in an arbitrary order.

- **s3Bucket** (*string*) – A specific Amazon S3 bucket name to limit the search for revisions.

If set to null, then all of the user's buckets will be searched.

- **s3KeyPrefix** (*string*) – A specific key prefix for the set of Amazon S3 objects to limit the search for revisions.
- **deployed** (*string*) – Whether to list revisions based on whether the revision is the target revision of an deployment group:
 - include: List revisions that are target revisions of a deployment group.
 - exclude: Do not list revisions that are target revisions of a deployment group.
 - ignore: List all revisions, regardless of whether they are target revisions of a deployment group.
- **nextToken** (*string*) – An identifier that was returned from the previous list application revisions call, which can be used to return the next set of applications in the list.

Return type dict

Returns

Response Syntax

```
{
  'revisions': [
    {
      'revisionType': 'S3'|'GitHub',
      's3Location': {
        'bucket': 'string',
        'key': 'string',
        'bundleType': 'tar'|'tgz'|'zip',
        'version': 'string',
        'eTag': 'string'
      }
    }
  ]
}
```

```

    },
    'gitHubLocation': {
        'repository': 'string',
        'commitId': 'string'
    }
},
],
'nextToken': 'string'
}

```

Response Structure

- (*dict*) –

Represents the output of a list application revisions operation.

- **revisions** (*list*) –

A list of revision locations that contain the matching revisions.

- * (*dict*) –

Information about an application revision’s location.

- **revisionType** (*string*) –

The application revision’s type:

- S3: An application revision stored in Amazon S3.
- GitHub: An application revision stored in GitHub.

- **s3Location** (*dict*) –

Information about the location of application artifacts that are stored in Amazon S3.

- **bucket** (*string*) –

The name of the Amazon S3 bucket where the application revision is stored.

- **key** (*string*) –

The name of the Amazon S3 object that represents the bundled artifacts for the application revision.

- **bundleType** (*string*) –

The file type of the application revision. Must be one of the following:

- tar: A tar archive file.
- tgz: A compressed tar archive file.
- zip: A zip archive file.

- **version** (*string*) –

A specific version of the Amazon S3 object that represents the bundled artifacts for the application revision.

If the version is not specified, the system will use the most recent version by default.

- **eTag** (*string*) –

The ETag of the Amazon S3 object that represents the bundled artifacts for the application revision.

If the ETag is not specified as an input parameter, ETag validation of the object will be skipped.

- **gitHubLocation** (*dict*) –

Information about the location of application artifacts that are stored in GitHub.

- **repository** (*string*) –

The GitHub account and repository pair that stores a reference to the commit that represents the bundled artifacts for the application revision.

Specified as account/repository.

- **commitId** (*string*) –

The SHA1 commit ID of the GitHub commit that references the that represents the bundled artifacts for the application revision.

- **nextToken** (*string*) –

If the amount of information that is returned is significantly large, an identifier will also be returned, which can be used in a subsequent list application revisions call to return the next set of application revisions in the list.

list_applications (***kwargs*)

Lists the applications registered with the applicable IAM user or AWS account.

Request Syntax

```
response = client.list_applications(  
    nextToken='string'  
)
```

Parameters **nextToken** (*string*) – An identifier that was returned from the previous list applications call, which can be used to return the next set of applications in the list.

Return type dict

Returns

Response Syntax

```
{  
    'applications': [  
        'string',  
    ],  
    'nextToken': 'string'  
}
```

Response Structure

- (*dict*) –

Represents the output of a list applications operation.

- **applications** (*list*) –

A list of application names.

* (*string*) –

- **nextToken** (*string*) –

If the amount of information that is returned is significantly large, an identifier will also be returned, which can be used in a subsequent list applications call to return the next set of applications in the list.

list_deployment_configs (***kwargs*)

Lists the deployment configurations with the applicable IAM user or AWS account.

Request Syntax

```
response = client.list_deployment_configs(
    nextToken='string'
)
```

Parameters **nextToken** (*string*) – An identifier that was returned from the previous list deployment configurations call, which can be used to return the next set of deployment configurations in the list.

Return type dict

Returns

Response Syntax

```
{
    'deploymentConfigsList': [
        'string',
    ],
    'nextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a list deployment configurations operation.

- **deploymentConfigsList** (*list*) –

A list of deployment configurations, including the built-in configurations such as CodeDeployDefault.OneAtATime.

* (*string*) –

- **nextToken** (*string*) –

If the amount of information that is returned is significantly large, an identifier will also be returned, which can be used in a subsequent list deployment configurations call to return the next set of deployment configurations in the list.

list_deployment_groups (***kwargs*)

Lists the deployment groups for an application registered with the applicable IAM user or AWS account.

Request Syntax

```
response = client.list_deployment_groups(
    applicationName='string',
    nextToken='string'
)
```

Parameters

- **applicationName** (*string*) – **[REQUIRED]**

The name of an existing AWS CodeDeploy application associated with the applicable IAM user or AWS account.

- **nextToken** (*string*) – An identifier that was returned from the previous list deployment groups call, which can be used to return the next set of deployment groups in the list.

Return type dict

Returns

Response Syntax

```
{
    'applicationName': 'string',
    'deploymentGroups': [
        'string',
    ],
    'nextToken': 'string'
}
```

Response Structure

- (dict) –

Represents the output of a list deployment groups operation.

- **applicationName** (*string*) –

The application name.

- **deploymentGroups** (*list*) –

A list of corresponding deployment group names.

* (*string*) –

- **nextToken** (*string*) –

If the amount of information that is returned is significantly large, an identifier will also be returned, which can be used in a subsequent list deployment groups call to return the next set of deployment groups in the list.

list_deployment_instances (**kwargs)

Lists the instances for a deployment associated with the applicable IAM user or AWS account.

Request Syntax

```
response = client.list_deployment_instances(
    deploymentId='string',
    nextToken='string',
    instanceStatusFilter=[
        'Pending' | 'InProgress' | 'Succeeded' | 'Failed' | 'Skipped' | 'Unknown',
    ]
)
```

Parameters

- **deploymentId** (*string*) – [REQUIRED]

The unique ID of a deployment.

- **nextToken** (*string*) – An identifier that was returned from the previous list deployment instances call, which can be used to return the next set of deployment instances in the list.

- **instanceStatusFilter** (*list*) – A subset of instances to list, by status:

- Pending: Include in the resulting list those instances with pending deployments.
- InProgress: Include in the resulting list those instances with in-progress deployments.
- Succeeded: Include in the resulting list those instances with succeeded deployments.
- Failed: Include in the resulting list those instances with failed deployments.
- Skipped: Include in the resulting list those instances with skipped deployments.
- Unknown: Include in the resulting list those instances with deployments in an unknown state.

– (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'instancesList': [
        'string',
    ],
    'nextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a list deployment instances operation.

- **instancesList** (*list*) –

A list of instances IDs.

* (*string*) –

- **nextToken** (*string*) –

If the amount of information that is returned is significantly large, an identifier will also be returned, which can be used in a subsequent list deployment instances call to return the next set of deployment instances in the list.

list_deployments (***kwargs*)

Lists the deployments within a deployment group for an application registered with the applicable IAM user or AWS account.

Request Syntax

```
response = client.list_deployments(
    applicationName='string',
    deploymentGroupName='string',
    includeOnlyStatuses=[
        'Created'|'Queued'|'InProgress'|'Succeeded'|'Failed'|'Stopped',
    ],
    createTimeRange={
        'start': datetime(2015, 1, 1),
        'end': datetime(2015, 1, 1)
    },
    nextToken='string'
)
```

Parameters

- **applicationName** (*string*) – The name of an existing AWS CodeDeploy application associated with the applicable IAM user or AWS account.
- **deploymentGroupName** (*string*) – The name of an existing deployment group for the specified application.
- **includeOnlyStatuses** (*list*) – A subset of deployments to list, by status:
 - Created: Include in the resulting list created deployments.
 - Queued: Include in the resulting list queued deployments.
 - In Progress: Include in the resulting list in-progress deployments.
 - Succeeded: Include in the resulting list succeeded deployments.
 - Failed: Include in the resulting list failed deployments.
 - Aborted: Include in the resulting list aborted deployments.

- *(string)* –
- **createTimeRange** (*dict*) – A deployment creation start- and end-time range for returning a subset of the list of deployments.
 - **start** (*datetime*) –
The time range’s start time.

Note: Specify null to leave the time range’s start time open-ended.
 - **end** (*datetime*) –
The time range’s end time.

Note: Specify null to leave the time range’s end time open-ended.
- **nextToken** (*string*) – An identifier that was returned from the previous list deployments call, which can be used to return the next set of deployments in the list.

Return type dict

Returns

Response Syntax

```
{
  'deployments': [
    'string',
  ],
  'nextToken': 'string'
}
```

Response Structure

- (*dict*) –
Represents the output of a list deployments operation.
 - **deployments** (*list*) –
A list of deployment IDs.
 - * (*string*) –
 - **nextToken** (*string*) –
If the amount of information that is returned is significantly large, an identifier will also be returned, which can be used in a subsequent list deployments call to return the next set of deployments in the list.

list_on_premises_instances (***kwargs*)

Gets a list of one or more on-premises instance names.

Unless otherwise specified, both registered and deregistered on-premises instance names will be listed. To list only registered or deregistered on-premises instance names, use the registration status parameter.

Request Syntax

```
response = client.list_on_premises_instances(
    registrationStatus='Registered'|'Deregistered',
    tagFilters=[
        {
            'Key': 'string',
            'Value': 'string',
            'Type': 'KEY_ONLY'|'VALUE_ONLY'|'KEY_AND_VALUE'
        },
    ],
)
```

```
],
nextToken='string'
)
```

Parameters

- **registrationStatus** (*string*) – The on-premises instances registration status:
 - **Deregistered**: Include in the resulting list deregistered on-premises instances.
 - **Registered**: Include in the resulting list registered on-premises instances.
- **tagFilters** (*list*) – The on-premises instance tags that will be used to restrict the corresponding on-premises instance names that are returned.
 - (*dict*) –

Information about an on-premises instance tag filter.

* **Key** (*string*) –

The on-premises instance tag filter key.

* **Value** (*string*) –

The on-premises instance tag filter value.

* **Type** (*string*) –

The on-premises instance tag filter type:

- **KEY_ONLY**: Key only.
- **VALUE_ONLY**: Value only.
- **KEY_AND_VALUE**: Key and value.

- **nextToken** (*string*) – An identifier that was returned from the previous list on-premises instances call, which can be used to return the next set of on-premises instances in the list.

Return type dict

Returns

Response Syntax

```
{
  'instanceNames': [
    'string',
  ],
  'nextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of list on-premises instances operation.

- **instanceNames** (*list*) –

The list of matching on-premises instance names.

* (*string*) –

- **nextToken** (*string*) –

If the amount of information that is returned is significantly large, an identifier will also be returned, which can be used in a subsequent list on-premises instances call to return the next set of on-premises instances in the list.

register_application_revision (**kwargs)

Registers with AWS CodeDeploy a revision for the specified application.

Request Syntax

```
response = client.register_application_revision(
    applicationName='string',
    description='string',
    revision={
        'revisionType': 'S3'|'GitHub',
        's3Location': {
            'bucket': 'string',
            'key': 'string',
            'bundleType': 'tar'|'tgz'|'zip',
            'version': 'string',
            'eTag': 'string'
        },
        'gitHubLocation': {
            'repository': 'string',
            'commitId': 'string'
        }
    }
)
```

Parameters

- **applicationName** (*string*) – **[REQUIRED]**

The name of an existing AWS CodeDeploy application associated with the applicable IAM user or AWS account.

- **description** (*string*) – A comment about the revision.
- **revision** (*dict*) – **[REQUIRED]**

Information about the application revision to register, including the revision's type and its location.

- **revisionType** (*string*) –

The application revision's type:

- * **S3**: An application revision stored in Amazon S3.
- * **GitHub**: An application revision stored in GitHub.

- **s3Location** (*dict*) –

Information about the location of application artifacts that are stored in Amazon S3.

- * **bucket** (*string*) –

The name of the Amazon S3 bucket where the application revision is stored.

- * **key** (*string*) –

The name of the Amazon S3 object that represents the bundled artifacts for the application revision.

- * **bundleType** (*string*) –

The file type of the application revision. Must be one of the following:

- **tar**: A tar archive file.
- **tgz**: A compressed tar archive file.
- **zip**: A zip archive file.

- * **version** (*string*) –

A specific version of the Amazon S3 object that represents the bundled artifacts for the application revision.

If the version is not specified, the system will use the most recent version by default.

* **eTag** (*string*) –

The ETag of the Amazon S3 object that represents the bundled artifacts for the application revision.

If the ETag is not specified as an input parameter, ETag validation of the object will be skipped.

– **gitHubLocation** (*dict*) –

Information about the location of application artifacts that are stored in GitHub.

* **repository** (*string*) –

The GitHub account and repository pair that stores a reference to the commit that represents the bundled artifacts for the application revision.

Specified as account/repository.

* **commitId** (*string*) –

The SHA1 commit ID of the GitHub commit that references the that represents the bundled artifacts for the application revision.

Returns None

register_on_premises_instance (***kwargs*)

Registers an on-premises instance.

Request Syntax

```
response = client.register_on_premises_instance(
    instanceName='string',
    iamUserArn='string'
)
```

Parameters

- **instanceName** (*string*) – **[REQUIRED]**

The name of the on-premises instance to register.

- **iamUserArn** (*string*) – **[REQUIRED]**

The ARN of the IAM user to associate with the on-premises instance.

Returns None

remove_tags_from_on_premises_instances (***kwargs*)

Removes one or more tags from one or more on-premises instances.

Request Syntax

```
response = client.remove_tags_from_on_premises_instances(
    tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
    instanceNames=[
        'string',
    ]
)
```

Parameters

- **tags** (*list*) – [REQUIRED]

The tag key-value pairs to remove from the on-premises instances.

– (*dict*) –

Information about a tag.

* **Key** (*string*) –

The tag’s key.

* **Value** (*string*) –

The tag’s value.

- **instanceNames** (*list*) – [REQUIRED]

The names of the on-premises instances to remove tags from.

– (*string*) –

Returns None

stop_deployment (***kwargs*)

Attempts to stop an ongoing deployment.

Request Syntax

```
response = client.stop_deployment(
    deploymentId='string'
)
```

Parameters **deploymentId** (*string*) – [REQUIRED]

The unique ID of a deployment.

Return type dict

Returns

Response Syntax

```
{
    'status': 'Pending'|'Succeeded',
    'statusMessage': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a stop deployment operation.

– **status** (*string*) –

The status of the stop deployment operation:

* Pending: The stop operation is pending.

* Succeeded: The stop operation succeeded.

– **statusMessage** (*string*) –

An accompanying status message.

update_application (***kwargs*)

Changes an existing application’s name.

Request Syntax


```
response = client.update_application(
    applicationName='string',
    newApplicationName='string'
)
```

Parameters

- **applicationName** (*string*) – The current name of the application that you want to change.
- **newApplicationName** (*string*) – The new name that you want to change the application to.

Returns None

update_deployment_group (**kwargs)

Changes information about an existing deployment group.

Request Syntax

```
response = client.update_deployment_group(
    applicationName='string',
    currentDeploymentGroupName='string',
    newDeploymentGroupName='string',
    deploymentConfigName='string',
    ec2TagFilters=[
        {
            'Key': 'string',
            'Value': 'string',
            'Type': 'KEY_ONLY' | 'VALUE_ONLY' | 'KEY_AND_VALUE'
        },
    ],
    onPremisesInstanceTagFilters=[
        {
            'Key': 'string',
            'Value': 'string',
            'Type': 'KEY_ONLY' | 'VALUE_ONLY' | 'KEY_AND_VALUE'
        },
    ],
    autoScalingGroups=[
        'string',
    ],
    serviceRoleArn='string'
)
```

Parameters

- **applicationName** (*string*) – **[REQUIRED]**
The application name corresponding to the deployment group to update.
- **currentDeploymentGroupName** (*string*) – **[REQUIRED]**
The current name of the existing deployment group.
- **newDeploymentGroupName** (*string*) – The new name of the deployment group, if you want to change it.
- **deploymentConfigName** (*string*) – The replacement deployment configuration name to use, if you want to change it.
- **ec2TagFilters** (*list*) – The replacement set of Amazon EC2 tags to filter on, if you want to change them.
– (*dict*) –
Information about a tag filter.

- * **Key** (*string*) –
The tag filter key.
- * **Value** (*string*) –
The tag filter value.
- * **Type** (*string*) –
The tag filter type:
 - KEY_ONLY: Key only.
 - VALUE_ONLY: Value only.
 - KEY_AND_VALUE: Key and value.
- **onPremisesInstanceTagFilters** (*list*) – The replacement set of on-premises instance tags for filter on, if you want to change them.
 - (*dict*) –
Information about an on-premises instance tag filter.
 - * **Key** (*string*) –
The on-premises instance tag filter key.
 - * **Value** (*string*) –
The on-premises instance tag filter value.
 - * **Type** (*string*) –
The on-premises instance tag filter type:
 - KEY_ONLY: Key only.
 - VALUE_ONLY: Value only.
 - KEY_AND_VALUE: Key and value.
- **autoScalingGroups** (*list*) – The replacement list of Auto Scaling groups to be included in the deployment group, if you want to change them.
 - (*string*) –
- **serviceRoleArn** (*string*) – A replacement service role's ARN, if you want to change it.

Return type dict

Returns

Response Syntax

```
{
  'hooksNotCleanedUp': [
    {
      'name': 'string',
      'hook': 'string'
    },
  ]
}
```

Response Structure

- (*dict*) –

Represents the output of an update deployment group operation.

- **hooksNotCleanedUp** (*list*) –

If the output contains no data, and the corresponding deployment group contained at least one Auto Scaling group, AWS CodeDeploy successfully removed all corresponding Auto Scaling lifecycle event hooks from the AWS account. If the output does contain data, AWS CodeDeploy could not remove some Auto Scaling lifecycle event hooks from the AWS account.

* (*dict*) –

Information about an Auto Scaling group.

- **name** (*string*) –

The Auto Scaling group name.

- **hook** (*string*) –

An Auto Scaling lifecycle event hook name.

CodePipeline

Table of Contents

- CodePipeline
 - Client

Client

class CodePipeline.**Client**

A low-level client representing AWS CodePipeline:

```
import boto3
```

```
client = boto3.client('codepipeline')
```

These are the available methods:

- `acknowledge_job()`
- `acknowledge_third_party_job()`
- `can_paginate()`
- `create_custom_action_type()`
- `create_pipeline()`
- `delete_custom_action_type()`
- `delete_pipeline()`
- `disable_stage_transition()`
- `enable_stage_transition()`
- `generate_presigned_url()`
- `get_job_details()`
- `get_paginator()`
- `get_pipeline()`
- `get_pipeline_state()`
- `get_third_party_job_details()`
- `get_waiter()`
- `list_action_types()`
- `list_pipelines()`
- `poll_for_jobs()`
- `poll_for_third_party_jobs()`
- `put_action_revision()`
- `put_job_failure_result()`
- `put_job_success_result()`
- `put_third_party_job_failure_result()`
- `put_third_party_job_success_result()`
- `start_pipeline_execution()`

- `update_pipeline()`

acknowledge_job (***kwargs*)

Returns information about a specified job and whether that job has been received by the job worker. Only used for custom actions.

Request Syntax

```
response = client.acknowledge_job(  
    jobId='string',  
    nonce='string'  
)
```

Parameters

- **jobId** (*string*) – [REQUIRED]

The unique system-generated ID of the job for which you want to confirm receipt.

- **nonce** (*string*) – [REQUIRED]

A system-generated random number that AWS CodePipeline uses to ensure that the job is being worked on by only one job worker. This number must be returned in the response.

Return type dict**Returns****Response Syntax**

```
{  
    'status': 'Created'|'Queued'|'Dispatched'|'InProgress'|'TimedOut'|'Succeeded'|  
}
```

Response Structure

- (*dict*) –

Represents the output of an acknowledge job action.

- **status** (*string*) –

Whether the job worker has received the specified job.

acknowledge_third_party_job (***kwargs*)

Confirms a job worker has received the specified job. Only used for partner actions.

Request Syntax

```
response = client.acknowledge_third_party_job(  
    jobId='string',  
    nonce='string',  
    clientToken='string'  
)
```

Parameters

- **jobId** (*string*) – [REQUIRED]

The unique system-generated ID of the job.

- **nonce** (*string*) – [REQUIRED]

A system-generated random number that AWS CodePipeline uses to ensure that the job is being worked on by only one job worker. This number must be returned in the response.

- **clientToken** (*string*) – [REQUIRED]

The clientToken portion of the clientId and clientToken pair used to verify that the calling entity is allowed access to the job and its details.

Return type dict

Returns

Response Syntax

```
{
    'status': 'Created'|'Queued'|'Dispatched'|'InProgress'|'TimedOut'|'Succeeded'|
}
```

Response Structure

- (*dict*) –

Represents the output of an acknowledge third party job action.

- **status** (*string*) –

The status information for the third party job, if any.

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

create_custom_action_type (***kwargs*)

Creates a new custom action that can be used in all pipelines associated with the AWS account. Only used for custom actions.

Request Syntax

```
response = client.create_custom_action_type(
    category='Source'|'Build'|'Deploy'|'Test'|'Invoke',
    provider='string',
    version='string',
    settings={
        'thirdPartyConfigurationUrl': 'string',
        'entityUrlTemplate': 'string',
        'executionUrlTemplate': 'string',
        'revisionUrlTemplate': 'string'
    },
    configurationProperties=[
        {
            'name': 'string',
            'required': True|False,
            'key': True|False,
            'secret': True|False,
            'queryable': True|False,
            'description': 'string',
            'type': 'String'|'Number'|'Boolean'
        },
    ],
    inputArtifactDetails={
        'minimumCount': 123,
```

```
        'maximumCount': 123
    },
    outputArtifactDetails={
        'minimumCount': 123,
        'maximumCount': 123
    }
)
```

Parameters

- **category** (*string*) – [REQUIRED]

The category of the custom action, such as a source action or a build action.

- **provider** (*string*) – [REQUIRED]

The provider of the service used in the custom action, such as AWS CodeDeploy.

- **version** (*string*) – [REQUIRED]

The version number of the custom action.

Note: A newly-created custom action is always assigned a version number of 1. This is required.

- **settings** (*dict*) – Returns information about the settings for an action type.

- **thirdPartyConfigurationUrl** (*string*) –

The URL of a sign-up page where users can sign up for an external service and perform initial configuration of the action provided by that service.

- **entityUrlTemplate** (*string*) –

The URL returned to the AWS CodePipeline console that provides a deep link to the resources of the external system, such as the configuration page for an AWS CodeDeploy deployment group. This link is provided as part of the action display within the pipeline.

- **executionUrlTemplate** (*string*) –

The URL returned to the AWS CodePipeline console that contains a link to the top-level landing page for the external system, such as console page for AWS CodeDeploy. This link is shown on the pipeline view page in the AWS CodePipeline console and provides a link to the execution entity of the external action.

- **revisionUrlTemplate** (*string*) –

The URL returned to the AWS CodePipeline console that contains a link to the page where customers can update or change the configuration of the external action.

- **configurationProperties** (*list*) – The configuration properties for the custom action.

- (*dict*) –

Represents information about an action configuration property.

- * **name** (*string*) – [REQUIRED]

The name of the action configuration property.

- * **required** (*boolean*) – [REQUIRED]

Whether the configuration property is a required value.

- * **key** (*boolean*) – [REQUIRED]

Whether the configuration property is a key.

* **secret** (*boolean*) – [REQUIRED]

Whether the configuration property is secret. Secrets are hidden from all calls except for `GetJobDetails`, `GetThirdPartyJobDetails`, `PollForJobs`, and `PollForThirdPartyJobs`.

When updating a pipeline, passing `*****` without changing any other values of the action will preserve the prior value of the secret.

* **queryable** (*boolean*) –

Indicates that the property will be used in conjunction with `PollForJobs`. When creating a custom action, an action can have up to one queryable property. If it has one, that property must be both required and not secret.

If you create a pipeline with a custom action type, and that custom action contains a queryable property, the value for that configuration property is subject to additional restrictions. The value must be less than or equal to twenty (20) characters. The value can contain only alphanumeric characters, underscores, and hyphens.

* **description** (*string*) –

The description of the action configuration property that will be displayed to users.

* **type** (*string*) –

The type of the configuration property.

• **inputArtifactDetails** (*dict*) – [REQUIRED]

Returns information about the details of an artifact.

– **minimumCount** (*integer*) – [REQUIRED]

The minimum number of artifacts allowed for the action type.

– **maximumCount** (*integer*) – [REQUIRED]

The maximum number of artifacts allowed for the action type.

• **outputArtifactDetails** (*dict*) – [REQUIRED]

Returns information about the details of an artifact.

– **minimumCount** (*integer*) – [REQUIRED]

The minimum number of artifacts allowed for the action type.

– **maximumCount** (*integer*) – [REQUIRED]

The maximum number of artifacts allowed for the action type.

Return type dict

Returns

Response Syntax

```
{
  'actionType': {
    'id': {
      'category': 'Source'|'Build'|'Deploy'|'Test'|'Invoke',
      'owner': 'AWS'|'ThirdParty'|'Custom',
      'provider': 'string',
      'version': 'string'
    },
    'settings': {
      'thirdPartyConfigurationUrl': 'string',
      'entityUrlTemplate': 'string',
    }
  }
}
```

```
        'executionUrlTemplate': 'string',
        'revisionUrlTemplate': 'string'
    },
    'actionConfigurationProperties': [
        {
            'name': 'string',
            'required': True|False,
            'key': True|False,
            'secret': True|False,
            'queryable': True|False,
            'description': 'string',
            'type': 'String'|'Number'|'Boolean'
        },
    ],
    'inputArtifactDetails': {
        'minimumCount': 123,
        'maximumCount': 123
    },
    'outputArtifactDetails': {
        'minimumCount': 123,
        'maximumCount': 123
    }
}
```

Response Structure

- *(dict)* –

Represents the output of a create custom action operation.

- **actionType** *(dict)* –

Returns information about the details of an action type.

- * **id** *(dict)* –

Represents information about an action type.

- **category** *(string)* –

A category defines what kind of action can be taken in the stage, and constrains the provider type for the action. Valid categories are limited to one of the values below.

- **owner** *(string)* –

The creator of the action being called.

- **provider** *(string)* –

The provider of the service being called by the action. Valid providers are determined by the action category. For example, an action in the Deploy category type might have a provider of AWS CodeDeploy, which would be specified as CodeDeploy.

- **version** *(string)* –

A string that identifies the action type.

- * **settings** *(dict)* –

The settings for the action type.

- **thirdPartyConfigurationUrl** *(string)* –

The URL of a sign-up page where users can sign up for an external service and perform initial configuration of the action provided by that service.

- **entityUrlTemplate** (*string*) –

The URL returned to the AWS CodePipeline console that provides a deep link to the resources of the external system, such as the configuration page for an AWS CodeDeploy deployment group. This link is provided as part of the action display within the pipeline.

- **executionUrlTemplate** (*string*) –

The URL returned to the AWS CodePipeline console that contains a link to the top-level landing page for the external system, such as console page for AWS CodeDeploy. This link is shown on the pipeline view page in the AWS CodePipeline console and provides a link to the execution entity of the external action.

- **revisionUrlTemplate** (*string*) –

The URL returned to the AWS CodePipeline console that contains a link to the page where customers can update or change the configuration of the external action.

- * **actionConfigurationProperties** (*list*) –

The configuration properties for the action type.

- (*dict*) –

Represents information about an action configuration property.

- **name** (*string*) –

The name of the action configuration property.

- **required** (*boolean*) –

Whether the configuration property is a required value.

- **key** (*boolean*) –

Whether the configuration property is a key.

- **secret** (*boolean*) –

Whether the configuration property is secret. Secrets are hidden from all calls except for `GetJobDetails`, `GetThirdPartyJobDetails`, `PollForJobs`, and `PollForThirdPartyJobs`.

When updating a pipeline, passing `*****` without changing any other values of the action will preserve the prior value of the secret.

- **queryable** (*boolean*) –

Indicates that the property will be used in conjunction with `PollForJobs`. When creating a custom action, an action can have up to one queryable property. If it has one, that property must be both required and not secret.

If you create a pipeline with a custom action type, and that custom action contains a queryable property, the value for that configuration property is subject to additional restrictions. The value must be less than or equal to twenty (20) characters. The value can contain only alphanumeric characters, underscores, and hyphens.

- **description** (*string*) –

The description of the action configuration property that will be displayed to users.

- **type** (*string*) –

The type of the configuration property.

- * **inputArtifactDetails** (*dict*) –

The details of the input artifact for the action, such as its commit ID.

- **minimumCount** (*integer*) –

The minimum number of artifacts allowed for the action type.

- **maximumCount** (*integer*) –

The maximum number of artifacts allowed for the action type.

- * **outputArtifactDetails** (*dict*) –

The details of the output artifact of the action, such as its commit ID.

- **minimumCount** (*integer*) –

The minimum number of artifacts allowed for the action type.

- **maximumCount** (*integer*) –

The maximum number of artifacts allowed for the action type.

create_pipeline (***kwargs*)

Creates a pipeline.

Request Syntax

```
response = client.create_pipeline(
    pipeline={
        'name': 'string',
        'roleArn': 'string',
        'artifactStore': {
            'type': 'S3',
            'location': 'string',
            'encryptionKey': {
                'id': 'string',
                'type': 'KMS'
            }
        },
        'stages': [
            {
                'name': 'string',
                'blockers': [
                    {
                        'name': 'string',
                        'type': 'Schedule'
                    },
                ],
            },
        ],
        'actions': [
            {
                'name': 'string',
                'actionTypeId': {
                    'category': 'Source'|'Build'|'Deploy'|'Test'|'Invoke',
                    'owner': 'AWS'|'ThirdParty'|'Custom',
                    'provider': 'string',
                    'version': 'string'
                },
            },
        ],
        'runOrder': 123,
```

```

        'configuration': {
            'string': 'string'
        },
        'outputArtifacts': [
            {
                'name': 'string'
            },
        ],
        'inputArtifacts': [
            {
                'name': 'string'
            },
        ],
        'roleArn': 'string'
    },
]
},
],
'version': 123
}
)

```

Parameters `pipeline` (*dict*) – [REQUIRED]

Represents the structure of actions and stages to be performed in the pipeline.

- **name** (*string*) – [REQUIRED]

The name of the action to be performed.

- **roleArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) for AWS CodePipeline to use to either perform actions with no actionRoleArn, or to use to assume roles for actions with an actionRoleArn.

- **artifactStore** (*dict*) – [REQUIRED]

The Amazon S3 location where artifacts are stored for the pipeline. If this Amazon S3 bucket is created manually, it must meet the requirements for AWS CodePipeline. For more information, see the Concepts.

- **type** (*string*) – [REQUIRED]

The type of the artifact store, such as S3.

- **location** (*string*) – [REQUIRED]

The location for storing the artifacts for a pipeline, such as an S3 bucket or folder.

- **encryptionKey** (*dict*) –

The AWS Key Management Service (AWS KMS) key used to encrypt the data in the artifact store. If this is undefined, the default key for Amazon S3 is used.

- * **id** (*string*) – [REQUIRED]

The ID of the AWS KMS key.

- * **type** (*string*) – [REQUIRED]

The type of AWS KMS key, such as a customer master key.

- **stages** (*list*) – [REQUIRED]

The stage in which to perform the action.

- (*dict*) –

Represents information about a stage and its definition.

* **name** (*string*) – [REQUIRED]

The name of the stage.

* **blockers** (*list*) –

The gates included in a stage.

· (*dict*) –

Represents information about a gate declaration.

· **name** (*string*) – [REQUIRED]

The name of the gate declaration.

· **type** (*string*) – [REQUIRED]

The type of the gate declaration.

* **actions** (*list*) – [REQUIRED]

The actions included in a stage.

· (*dict*) –

Represents information about an action declaration.

· **name** (*string*) – [REQUIRED]

The action declaration's name.

· **actionTypeId** (*dict*) – [REQUIRED]

The configuration information for the action type.

· **category** (*string*) – [REQUIRED]

A category defines what kind of action can be taken in the stage, and constrains the provider type for the action. Valid categories are limited to one of the values below.

· **owner** (*string*) – [REQUIRED]

The creator of the action being called.

· **provider** (*string*) – [REQUIRED]

The provider of the service being called by the action. Valid providers are determined by the action category. For example, an action in the Deploy category type might have a provider of AWS CodeDeploy, which would be specified as CodeDeploy.

· **version** (*string*) – [REQUIRED]

A string that identifies the action type.

· **runOrder** (*integer*) –

The order in which actions are run.

· **configuration** (*dict*) –

The action declaration's configuration.

· (*string*) –

· (*string*) –

· **outputArtifacts** (*list*) –

The name or ID of the result of the action declaration, such as a test or build artifact.

· (*dict*) –

Represents information about the output of an action.

· **name** (*string*) – [REQUIRED]

The name of the output of an artifact, such as "My App".

The input artifact of an action must exactly match the output artifact declared in a preceding action, but the input artifact does not have to be the next action in strict sequence from the action that provided the output artifact. Actions in parallel can declare different output artifacts, which are in turn consumed by different following actions.

Output artifact names must be unique within a pipeline.

- **inputArtifacts** (*list*) –

The name or ID of the artifact consumed by the action, such as a test or build artifact.

- (*dict*) –

Represents information about an artifact to be worked on, such as a test or build artifact.

- **name** (*string*) – **[REQUIRED]**

The name of the artifact to be worked on, for example, “My App”.

The input artifact of an action must exactly match the output artifact declared in a preceding action, but the input artifact does not have to be the next action in strict sequence from the action that provided the output artifact. Actions in parallel can declare different output artifacts, which are in turn consumed by different following actions.

- **roleArn** (*string*) –

The ARN of the IAM service role that will perform the declared action. This is assumed through the roleArn for the pipeline.

- **version** (*integer*) –

The version number of the pipeline. A new pipeline always has a version number of 1. This number is automatically incremented when a pipeline is updated.

Return type dict

Returns

Response Syntax

```
{
  'pipeline': {
    'name': 'string',
    'roleArn': 'string',
    'artifactStore': {
      'type': 'S3',
      'location': 'string',
      'encryptionKey': {
        'id': 'string',
        'type': 'KMS'
      }
    }
  },
  'stages': [
    {
      'name': 'string',
      'blockers': [
        {
          'name': 'string',
```

```
        'type': 'Schedule'
    },
],
'actions': [
    {
        'name': 'string',
        'actionTypeId': {
            'category': 'Source'|'Build'|'Deploy'|'Test'|'Invoke',
            'owner': 'AWS'|'ThirdParty'|'Custom',
            'provider': 'string',
            'version': 'string'
        },
        'runOrder': 123,
        'configuration': {
            'string': 'string'
        },
        'outputArtifacts': [
            {
                'name': 'string'
            },
        ],
        'inputArtifacts': [
            {
                'name': 'string'
            },
        ],
        'roleArn': 'string'
    },
],
],
},
'version': 123
}
```

Response Structure

- (*dict*) –

Represents the output of a create pipeline action.

- **pipeline** (*dict*) –

Represents the structure of actions and stages to be performed in the pipeline.

- * **name** (*string*) –

The name of the action to be performed.

- * **roleArn** (*string*) –

The Amazon Resource Name (ARN) for AWS CodePipeline to use to either perform actions with no `actionRoleArn`, or to use to assume roles for actions with an `actionRoleArn`.

- * **artifactStore** (*dict*) –

The Amazon S3 location where artifacts are stored for the pipeline. If this Amazon S3 bucket is created manually, it must meet the requirements for AWS CodePipeline. For more information, see the Concepts.

- **type** (*string*) –

The type of the artifact store, such as S3.

- **location** (*string*) –

The location for storing the artifacts for a pipeline, such as an S3 bucket or folder.

- **encryptionKey** (*dict*) –

The AWS Key Management Service (AWS KMS) key used to encrypt the data in the artifact store. If this is undefined, the default key for Amazon S3 is used.

- **id** (*string*) –

The ID of the AWS KMS key.

- **type** (*string*) –

The type of AWS KMS key, such as a customer master key.

- * **stages** (*list*) –

The stage in which to perform the action.

- (*dict*) –

Represents information about a stage and its definition.

- **name** (*string*) –

The name of the stage.

- **blockers** (*list*) –

The gates included in a stage.

- (*dict*) –

Represents information about a gate declaration.

- **name** (*string*) –

The name of the gate declaration.

- **type** (*string*) –

The type of the gate declaration.

- **actions** (*list*) –

The actions included in a stage.

- (*dict*) –

Represents information about an action declaration.

- **name** (*string*) –

The action declaration's name.

- **actionTypeId** (*dict*) –

The configuration information for the action type.

- **category** (*string*) –

A category defines what kind of action can be taken in the stage, and constrains the provider type for the action. Valid categories are limited to one of the values below.

- **owner** (*string*) –

The creator of the action being called.

- **provider** (*string*) –

The provider of the service being called by the action. Valid providers are determined by the action category. For example, an action in the Deploy category type might have a provider of AWS CodeDeploy, which would be specified as CodeDeploy.

- **version** (*string*) –

A string that identifies the action type.

- **runOrder** (*integer*) –

The order in which actions are run.

- **configuration** (*dict*) –

The action declaration's configuration.

- (*string*) –

- (*string*) –

- **outputArtifacts** (*list*) –

The name or ID of the result of the action declaration, such as a test or build artifact.

- (*dict*) –

Represents information about the output of an action.

- **name** (*string*) –

The name of the output of an artifact, such as “My App”.

The input artifact of an action must exactly match the output artifact declared in a preceding action, but the input artifact does not have to be the next action in strict sequence from the action that provided the output artifact. Actions in parallel can declare different output artifacts, which are in turn consumed by different following actions.

Output artifact names must be unique within a pipeline.

- **inputArtifacts** (*list*) –

The name or ID of the artifact consumed by the action, such as a test or build artifact.

- (*dict*) –

Represents information about an artifact to be worked on, such as a test or build artifact.

- **name** (*string*) –

The name of the artifact to be worked on, for example, “My App”.

The input artifact of an action must exactly match the output artifact declared in a preceding action, but the input artifact does not have to be the next action in strict sequence from the action that provided the output artifact. Actions in parallel can declare different output artifacts, which are in turn consumed by different following actions.

- **roleArn** (*string*) –

The ARN of the IAM service role that will perform the declared action. This is assumed through the roleArn for the pipeline.

- * **version** (*integer*) –

The version number of the pipeline. A new pipeline always has a version number of 1. This number is automatically incremented when a pipeline is updated.

delete_custom_action_type (***kwargs*)

Marks a custom action as deleted. PollForJobs for the custom action will fail after the action is marked for deletion. Only used for custom actions.

Warning: You cannot recreate a custom action after it has been deleted unless you increase the version number of the action.

Request Syntax

```
response = client.delete_custom_action_type(
    category='Source' | 'Build' | 'Deploy' | 'Test' | 'Invoke',
    provider='string',
    version='string'
)
```

Parameters

- **category** (*string*) – [REQUIRED]

The category of the custom action that you want to delete, such as source or deploy.

- **provider** (*string*) – [REQUIRED]

The provider of the service used in the custom action, such as AWS CodeDeploy.

- **version** (*string*) – [REQUIRED]

The version of the custom action to delete.

Returns None

delete_pipeline (***kwargs*)

Deletes the specified pipeline.

Request Syntax

```
response = client.delete_pipeline(
    name='string'
)
```

Parameters **name** (*string*) – [REQUIRED]

The name of the pipeline to be deleted.

Returns None

disable_stage_transition (***kwargs*)

Prevents artifacts in a pipeline from transitioning to the next stage in the pipeline.

Request Syntax

```
response = client.disable_stage_transition(
    pipelineName='string',
    stageName='string',
    transitionType='Inbound' | 'Outbound',
    reason='string'
)
```

Parameters

- **pipelineName** (*string*) – [REQUIRED]

The name of the pipeline in which you want to disable the flow of artifacts from one stage to another.

- **stageName** (*string*) – [REQUIRED]

The name of the stage where you want to disable the inbound or outbound transition of artifacts.

- **transitionType** (*string*) – [REQUIRED]

Specifies whether artifacts will be prevented from transitioning into the stage and being processed by the actions in that stage (inbound), or prevented from transitioning from the stage after they have been processed by the actions in that stage (outbound).

- **reason** (*string*) – [REQUIRED]

The reason given to the user why a stage is disabled, such as waiting for manual approval or manual tests. This message is displayed in the pipeline console UI.

Returns None

enable_stage_transition (***kwargs*)

Enables artifacts in a pipeline to transition to a stage in a pipeline.

Request Syntax

```
response = client.enable_stage_transition(
    pipelineName='string',
    stageName='string',
    transitionType='Inbound'|'Outbound'
)
```

Parameters

- **pipelineName** (*string*) – [REQUIRED]

The name of the pipeline in which you want to enable the flow of artifacts from one stage to another.

- **stageName** (*string*) – [REQUIRED]

The name of the stage where you want to enable the transition of artifacts, either into the stage (inbound) or from that stage to the next stage (outbound).

- **transitionType** (*string*) – [REQUIRED]

Specifies whether artifacts will be allowed to enter the stage and be processed by the actions in that stage (inbound) or whether already-processed artifacts will be allowed to transition to the next stage (outbound).

Returns None

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_job_details (***kwargs*)

Returns information about a job. Only used for custom actions.

Warning: When this API is called, AWS CodePipeline returns temporary credentials for the Amazon S3 bucket used to store artifacts for the pipeline, if the action requires access to that Amazon S3 bucket for input or output artifacts. Additionally, this API returns any secret values defined for the action.

Request Syntax

```
response = client.get_job_details(
    jobId='string'
)
```

Parameters `jobId` (*string*) – [REQUIRED]

The unique system-generated ID for the job.

Return type dict

Returns

Response Syntax

```
{
    'jobDetails': {
        'id': 'string',
        'data': {
            'actionTypeId': {
                'category': 'Source'|'Build'|'Deploy'|'Test'|'Invoke',
                'owner': 'AWS'|'ThirdParty'|'Custom',
                'provider': 'string',
                'version': 'string'
            },
            'actionConfiguration': {
                'configuration': {
                    'string': 'string'
                }
            },
            'pipelineContext': {
                'pipelineName': 'string',
                'stage': {
                    'name': 'string'
                },
                'action': {
                    'name': 'string'
                }
            },
            'inputArtifacts': [
                {
                    'name': 'string',
                    'revision': 'string',
                    'location': {
                        'type': 'S3',
                        's3Location': {
                            'bucketName': 'string',
                            'objectKey': 'string'
                        }
                    }
                }
            ],
            'outputArtifacts': [
                {
```

```
        'name': 'string',
        'revision': 'string',
        'location': {
            'type': 'S3',
            's3Location': {
                'bucketName': 'string',
                'objectKey': 'string'
            }
        },
    },
    ],
    'artifactCredentials': {
        'accessKeyId': 'string',
        'secretAccessKey': 'string',
        'sessionToken': 'string'
    },
    'continuationToken': 'string',
    'encryptionKey': {
        'id': 'string',
        'type': 'KMS'
    }
},
'accountId': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a get job details action.

- **jobDetails** (*dict*) –

The details of the job.

Note: If `AWSSessionCredentials` is used, a long-running job can call `GetJobDetails` again to obtain new credentials.

- * **id** (*string*) –

The unique system-generated ID of the job.

- * **data** (*dict*) –

Represents additional information about a job required for a job worker to complete the job.

- **actionTypeId** (*dict*) –

Represents information about an action type.

- **category** (*string*) –

A category defines what kind of action can be taken in the stage, and constrains the provider type for the action. Valid categories are limited to one of the values below.

- **owner** (*string*) –

The creator of the action being called.

- **provider** (*string*) –

The provider of the service being called by the action. Valid providers are determined by the action category. For example,

an action in the Deploy category type might have a provider of AWS CodeDeploy, which would be specified as CodeDeploy.

- **version** (*string*) –

A string that identifies the action type.

- **actionConfiguration** (*dict*) –

Represents information about an action configuration.

- **configuration** (*dict*) –

The configuration data for the action.

- (*string*) –

- (*string*) –

- **pipelineContext** (*dict*) –

Represents information about a pipeline to a job worker.

- **pipelineName** (*string*) –

The name of the pipeline. This is a user-specified value. Pipeline names must be unique across all pipeline names under an Amazon Web Services account.

- **stage** (*dict*) –

The stage of the pipeline.

- **name** (*string*) –

The name of the stage.

- **action** (*dict*) –

Represents the context of an action within the stage of a pipeline to a job worker.

- **name** (*string*) –

The name of the action within the context of a job.

- **inputArtifacts** (*list*) –

The artifact supplied to the job.

- (*dict*) –

Represents information about an artifact that will be worked upon by actions in the pipeline.

- **name** (*string*) –

The artifact's name.

- **revision** (*string*) –

The artifact's revision ID. Depending on the type of object, this could be a commit ID (GitHub) or a revision ID (Amazon S3).

- **location** (*dict*) –

The location of an artifact.

- **type** (*string*) –

The type of artifact in the location.

- **s3Location** (*dict*) –

The Amazon S3 bucket that contains the artifact.

- **bucketName** (*string*) –

The name of the Amazon S3 bucket.

- **objectKey** (*string*) –
The key of the object in the Amazon S3 bucket, which uniquely identifies the object in the bucket.
- **outputArtifacts** (*list*) –
The output of the job.
- (*dict*) –
Represents information about an artifact that will be worked upon by actions in the pipeline.
- **name** (*string*) –
The artifact's name.
- **revision** (*string*) –
The artifact's revision ID. Depending on the type of object, this could be a commit ID (GitHub) or a revision ID (Amazon S3).
- **location** (*dict*) –
The location of an artifact.
- **type** (*string*) –
The type of artifact in the location.
- **s3Location** (*dict*) –
The Amazon S3 bucket that contains the artifact.
- **bucketName** (*string*) –
The name of the Amazon S3 bucket.
- **objectKey** (*string*) –
The key of the object in the Amazon S3 bucket, which uniquely identifies the object in the bucket.
- **artifactCredentials** (*dict*) –
Represents an AWS session credentials object. These credentials are temporary credentials that are issued by AWS Secure Token Service (STS). They can be used to access input and output artifacts in the Amazon S3 bucket used to store artifact for the pipeline in AWS CodePipeline.
- **accessKeyId** (*string*) –
The access key for the session.
- **secretAccessKey** (*string*) –
The secret access key for the session.
- **sessionToken** (*string*) –
The token for the session.
- **continuationToken** (*string*) –
A system-generated token, such as a AWS CodeDeploy deployment ID, that a job requires in order to continue the job asynchronously.
- **encryptionKey** (*dict*) –
Represents information about the AWS Key Management Service (AWS KMS) key used to encrypt data in the artifact store.

- **id** (*string*) –
The ID of the AWS KMS key.
- **type** (*string*) –
The type of AWS KMS key, such as a customer master key.
- * **accountId** (*string*) –
The AWS account ID associated with the job.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type L{botocore.paginate.Paginator}

Returns A paginator object.

get_pipeline (***kwargs*)

Returns the metadata, structure, stages, and actions of a pipeline. Can be used to return the entire structure of a pipeline in JSON format, which can then be modified and used to update the pipeline structure with `UpdatePipeline`.

Request Syntax

```
response = client.get_pipeline(
    name='string',
    version=123
)
```

Parameters

- **name** (*string*) – **[REQUIRED]**

The name of the pipeline for which you want to get information. Pipeline names must be unique under an Amazon Web Services (AWS) user account.

- **version** (*integer*) – The version number of the pipeline. If you do not specify a version, defaults to the most current version.

Return type dict

Returns

Response Syntax

```
{
  'pipeline': {
    'name': 'string',
    'roleArn': 'string',
    'artifactStore': {
      'type': 'S3',
      'location': 'string',
      'encryptionKey': {
        'id': 'string',
        'type': 'KMS'
      }
    },
  },
  'stages': [
```

```
{
  'name': 'string',
  'blockers': [
    {
      'name': 'string',
      'type': 'Schedule'
    },
  ],
  'actions': [
    {
      'name': 'string',
      'actionTypeId': {
        'category': 'Source'|'Build'|'Deploy'|'Test'|'Invoke',
        'owner': 'AWS'|'ThirdParty'|'Custom',
        'provider': 'string',
        'version': 'string'
      },
      'runOrder': 123,
      'configuration': {
        'string': 'string'
      },
      'outputArtifacts': [
        {
          'name': 'string'
        },
      ],
      'inputArtifacts': [
        {
          'name': 'string'
        },
      ],
      'roleArn': 'string'
    },
  ],
},
'version': 123
}
```

Response Structure

- (*dict*) –

Represents the output of a get pipeline action.

- **pipeline** (*dict*) –

Represents the structure of actions and stages to be performed in the pipeline.

- * **name** (*string*) –

The name of the action to be performed.

- * **roleArn** (*string*) –

The Amazon Resource Name (ARN) for AWS CodePipeline to use to either perform actions with no `actionRoleArn`, or to use to assume roles for actions with an `actionRoleArn`.

- * **artifactStore** (*dict*) –

The Amazon S3 location where artifacts are stored for the pipeline.

If this Amazon S3 bucket is created manually, it must meet the requirements for AWS CodePipeline. For more information, see the Concepts.

- **type** (*string*) –
The type of the artifact store, such as S3.
- **location** (*string*) –
The location for storing the artifacts for a pipeline, such as an S3 bucket or folder.
- **encryptionKey** (*dict*) –
The AWS Key Management Service (AWS KMS) key used to encrypt the data in the artifact store. If this is undefined, the default key for Amazon S3 is used.
- **id** (*string*) –
The ID of the AWS KMS key.
- **type** (*string*) –
The type of AWS KMS key, such as a customer master key.
- * **stages** (*list*) –
The stage in which to perform the action.
 - (*dict*) –
Represents information about a stage and its definition.
 - **name** (*string*) –
The name of the stage.
 - **blockers** (*list*) –
The gates included in a stage.
 - (*dict*) –
Represents information about a gate declaration.
 - **name** (*string*) –
The name of the gate declaration.
 - **type** (*string*) –
The type of the gate declaration.
 - **actions** (*list*) –
The actions included in a stage.
 - (*dict*) –
Represents information about an action declaration.
 - **name** (*string*) –
The action declaration's name.
 - **actionTypeId** (*dict*) –
The configuration information for the action type.
 - **category** (*string*) –
A category defines what kind of action can be taken in the stage, and constrains the provider type for the action. Valid categories are limited to one of the values below.
 - **owner** (*string*) –
The creator of the action being called.

- **provider** (*string*) –

The provider of the service being called by the action. Valid providers are determined by the action category. For example, an action in the Deploy category type might have a provider of AWS CodeDeploy, which would be specified as CodeDeploy.

- **version** (*string*) –

A string that identifies the action type.

- **runOrder** (*integer*) –

The order in which actions are run.

- **configuration** (*dict*) –

The action declaration's configuration.

- (*string*) –

- (*string*) –

- **outputArtifacts** (*list*) –

The name or ID of the result of the action declaration, such as a test or build artifact.

- (*dict*) –

Represents information about the output of an action.

- **name** (*string*) –

The name of the output of an artifact, such as “My App”.

The input artifact of an action must exactly match the output artifact declared in a preceding action, but the input artifact does not have to be the next action in strict sequence from the action that provided the output artifact. Actions in parallel can declare different output artifacts, which are in turn consumed by different following actions.

Output artifact names must be unique within a pipeline.

- **inputArtifacts** (*list*) –

The name or ID of the artifact consumed by the action, such as a test or build artifact.

- (*dict*) –

Represents information about an artifact to be worked on, such as a test or build artifact.

- **name** (*string*) –

The name of the artifact to be worked on, for example, “My App”.

The input artifact of an action must exactly match the output artifact declared in a preceding action, but the input artifact does not have to be the next action in strict sequence from the action that provided the output artifact. Actions in parallel can declare different output artifacts, which are in turn consumed by different following actions.

- **roleArn** (*string*) –

The ARN of the IAM service role that will perform the declared action. This is assumed through the roleArn for the pipeline.

* **version** (*integer*) –

The version number of the pipeline. A new pipeline always has a version number of 1. This number is automatically incremented when a pipeline is updated.

get_pipeline_state (***kwargs*)

Returns information about the state of a pipeline, including the stages, actions, and details about the last run of the pipeline.

Request Syntax

```
response = client.get_pipeline_state(
    name='string'
)
```

Parameters *name* (*string*) – **[REQUIRED]**

The name of the pipeline about which you want to get information.

Return type dict

Returns

Response Syntax

```
{
    'pipelineName': 'string',
    'pipelineVersion': 123,
    'stageStates': [
        {
            'stageName': 'string',
            'inboundTransitionState': {
                'enabled': True|False,
                'lastChangedBy': 'string',
                'lastChangedAt': datetime(2015, 1, 1),
                'disabledReason': 'string'
            },
            'actionStates': [
                {
                    'actionName': 'string',
                    'currentRevision': {
                        'revisionId': 'string',
                        'revisionChangeId': 'string',
                        'created': datetime(2015, 1, 1)
                    },
                    'latestExecution': {
                        'status': 'InProgress'|'Succeeded'|'Failed',
                        'summary': 'string',
                        'lastStatusChange': datetime(2015, 1, 1),
                        'externalExecutionId': 'string',
                        'externalExecutionUrl': 'string',
                        'percentComplete': 123,
                        'errorDetails': {
                            'code': 'string',
                            'message': 'string'
                        }
                    },
                    'entityUrl': 'string',
                    'revisionUrl': 'string'
                },
            ],
        },
    ],
}
```

```
    ],
    },
],
'created': datetime(2015, 1, 1),
'updated': datetime(2015, 1, 1)
}
```

Response Structure

- (*dict*) –

Represents the output of a get pipeline state action.

- **pipelineName** (*string*) –

The name of the pipeline for which you want to get the state.

- **pipelineVersion** (*integer*) –

The version number of the pipeline.

Note: A newly-created pipeline is always assigned a version number of 1.

- **stageStates** (*list*) –

A list of the pipeline stage output information, including stage name, state, most recent run details, whether the stage is disabled, and other data.

- * (*dict*) –

Represents information about the state of the stage.

- **stageName** (*string*) –

The name of the stage.

- **inboundTransitionState** (*dict*) –

The state of the inbound transition, which is either enabled or disabled.

- **enabled** (*boolean*) –

Whether the transition between stages is enabled (true) or disabled (false).

- **lastChangedBy** (*string*) –

The ID of the user who last changed the transition state.

- **lastChangedAt** (*datetime*) –

The timestamp when the transition state was last changed.

- **disabledReason** (*string*) –

The user-specified reason why the transition between two stages of a pipeline was disabled.

- **actionStates** (*list*) –

The state of the stage.

- (*dict*) –

Represents information about the state of an action.

- **actionName** (*string*) –

The name of the action.

- **currentRevision** (*dict*) –

Represents information about the version (or revision) of an action.

- **revisionId** (*string*) –
The system-generated unique ID that identifies the revision number of the action.
- **revisionChangeId** (*string*) –
The unique identifier of the change that set the state to this revision, for example a deployment ID or timestamp.
- **created** (*datetime*) –
The date and time when the most recent version of the action was created, in timestamp format.
- **latestExecution** (*dict*) –
Represents information about how an action runs.
- **status** (*string*) –
The status of the action, or for a completed action, the last status of the action.
- **summary** (*string*) –
A summary of the run of the action.
- **lastStatusChange** (*datetime*) –
The last status change of the action.
- **externalExecutionId** (*string*) –
The external ID of the run of the action.
- **externalExecutionUrl** (*string*) –
The URL of a resource external to AWS that will be used when running the action, for example an external repository URL.
- **percentComplete** (*integer*) –
A percentage of completeness of the action as it runs.
- **errorDetails** (*dict*) –
The details of an error returned by a URL external to AWS.
- **code** (*string*) –
The system ID or error number code of the error.
- **message** (*string*) –
The text of the error message.
- **entityUrl** (*string*) –
A URL link for more information about the state of the action, such as a deployment group details page.
- **revisionUrl** (*string*) –
A URL link for more information about the revision, such as a commit details page.

– **created** (*datetime*) –

The date and time the pipeline was created, in timestamp format.

– **updated** (*datetime*) –

The date and time the pipeline was last updated, in timestamp format.

get_third_party_job_details (***kwargs*)

Requests the details of a job for a third party action. Only used for partner actions.

Warning: When this API is called, AWS CodePipeline returns temporary credentials for the Amazon S3 bucket used to store artifacts for the pipeline, if the action requires access to that Amazon S3 bucket for input or output artifacts. Additionally, this API returns any secret values defined for the action.

Request Syntax

```
response = client.get_third_party_job_details(
    jobId='string',
    clientToken='string'
)
```

Parameters

- **jobId** (*string*) – [REQUIRED]

The unique system-generated ID used for identifying the job.

- **clientToken** (*string*) – [REQUIRED]

The clientToken portion of the clientId and clientToken pair used to verify that the calling entity is allowed access to the job and its details.

Return type dict

Returns

Response Syntax

```
{
    'jobDetails': {
        'id': 'string',
        'data': {
            'actionTypeId': {
                'category': 'Source'|'Build'|'Deploy'|'Test'|'Invoke',
                'owner': 'AWS'|'ThirdParty'|'Custom',
                'provider': 'string',
                'version': 'string'
            },
            'actionConfiguration': {
                'configuration': {
                    'string': 'string'
                }
            },
            'pipelineContext': {
                'pipelineName': 'string',
                'stage': {
                    'name': 'string'
                },
                'action': {
                    'name': 'string'
                }
            },
            'inputArtifacts': [
                {
                    'name': 'string',
                    'revision': 'string',
                    'location': {
                        'type': 'S3',
                        's3Location': {
                            'bucketName': 'string',
                            'objectKey': 'string'
                        }
                    }
                }
            ]
        }
    }
}
```

```

    },
    ],
    'outputArtifacts': [
        {
            'name': 'string',
            'revision': 'string',
            'location': {
                'type': 'S3',
                's3Location': {
                    'bucketName': 'string',
                    'objectKey': 'string'
                }
            }
        }
    ],
    ],
    'artifactCredentials': {
        'accessKeyId': 'string',
        'secretAccessKey': 'string',
        'sessionToken': 'string'
    },
    'continuationToken': 'string',
    'encryptionKey': {
        'id': 'string',
        'type': 'KMS'
    }
},
'nonce': 'string'
}

```

Response Structure

- (*dict*) –

Represents the output of a get third party job details action.

- **jobDetails** (*dict*) –

The details of the job, including any protected values defined for the job.

- * **id** (*string*) –

The identifier used to identify the job details in AWS CodePipeline.

- * **data** (*dict*) –

The data to be returned by the third party job worker.

- **actionTypeId** (*dict*) –

Represents information about an action type.

- **category** (*string*) –

A category defines what kind of action can be taken in the stage, and constrains the provider type for the action. Valid categories are limited to one of the values below.

- **owner** (*string*) –

The creator of the action being called.

- **provider** (*string*) –

The provider of the service being called by the action. Valid providers are determined by the action category. For example,

an action in the Deploy category type might have a provider of AWS CodeDeploy, which would be specified as CodeDeploy.

- **version** (*string*) –

A string that identifies the action type.

- **actionConfiguration** (*dict*) –

Represents information about an action configuration.

- **configuration** (*dict*) –

The configuration data for the action.

- (*string*) –

- (*string*) –

- **pipelineContext** (*dict*) –

Represents information about a pipeline to a job worker.

- **pipelineName** (*string*) –

The name of the pipeline. This is a user-specified value. Pipeline names must be unique across all pipeline names under an Amazon Web Services account.

- **stage** (*dict*) –

The stage of the pipeline.

- **name** (*string*) –

The name of the stage.

- **action** (*dict*) –

Represents the context of an action within the stage of a pipeline to a job worker.

- **name** (*string*) –

The name of the action within the context of a job.

- **inputArtifacts** (*list*) –

The name of the artifact that will be worked upon by the action, if any. This name might be system-generated, such as “MyApp”, or might be defined by the user when the action is created. The input artifact name must match the name of an output artifact generated by an action in an earlier action or stage of the pipeline.

- (*dict*) –

Represents information about an artifact that will be worked upon by actions in the pipeline.

- **name** (*string*) –

The artifact’s name.

- **revision** (*string*) –

The artifact’s revision ID. Depending on the type of object, this could be a commit ID (GitHub) or a revision ID (Amazon S3).

- **location** (*dict*) –

The location of an artifact.

- **type** (*string*) –

The type of artifact in the location.

- **s3Location** (*dict*) –
The Amazon S3 bucket that contains the artifact.
- **bucketName** (*string*) –
The name of the Amazon S3 bucket.
- **objectKey** (*string*) –
The key of the object in the Amazon S3 bucket, which uniquely identifies the object in the bucket.
- **outputArtifacts** (*list*) –
The name of the artifact that will be the result of the action, if any. This name might be system-generated, such as “My-BuiltApp”, or might be defined by the user when the action is created.
- (*dict*) –
Represents information about an artifact that will be worked upon by actions in the pipeline.
- **name** (*string*) –
The artifact’s name.
- **revision** (*string*) –
The artifact’s revision ID. Depending on the type of object, this could be a commit ID (GitHub) or a revision ID (Amazon S3).
- **location** (*dict*) –
The location of an artifact.
- **type** (*string*) –
The type of artifact in the location.
- **s3Location** (*dict*) –
The Amazon S3 bucket that contains the artifact.
- **bucketName** (*string*) –
The name of the Amazon S3 bucket.
- **objectKey** (*string*) –
The key of the object in the Amazon S3 bucket, which uniquely identifies the object in the bucket.
- **artifactCredentials** (*dict*) –
Represents an AWS session credentials object. These credentials are temporary credentials that are issued by AWS Secure Token Service (STS). They can be used to access input and output artifacts in the Amazon S3 bucket used to store artifact for the pipeline in AWS CodePipeline.
- **accessKeyId** (*string*) –
The access key for the session.
- **secretAccessKey** (*string*) –
The secret access key for the session.
- **sessionToken** (*string*) –
The token for the session.
- **continuationToken** (*string*) –

A system-generated token, such as a AWS CodeDeploy deployment ID, that a job requires in order to continue the job asynchronously.

- **encryptionKey** (*dict*) –

The AWS Key Management Service (AWS KMS) key used to encrypt and decrypt data in the artifact store for the pipeline.

- **id** (*string*) –

The ID of the AWS KMS key.

- **type** (*string*) –

The type of AWS KMS key, such as a customer master key.

- * **nonce** (*string*) –

A system-generated random number that AWS CodePipeline uses to ensure that the job is being worked on by only one job worker. This number must be returned in the response.

get_waiter (*waiter_name*)

list_action_types (**kwargs)

Gets a summary of all AWS CodePipeline action types associated with your account.

Request Syntax

```
response = client.list_action_types(
    actionOwnerFilter='AWS'|'ThirdParty'|'Custom',
    nextToken='string'
)
```

Parameters

- **actionOwnerFilter** (*string*) – Filters the list of action types to those created by a specified entity.
- **nextToken** (*string*) – An identifier that was returned from the previous list action types call, which can be used to return the next set of action types in the list.

Return type dict

Returns

Response Syntax

```
{
    'actionTypes': [
        {
            'id': {
                'category': 'Source'|'Build'|'Deploy'|'Test'|'Invoke',
                'owner': 'AWS'|'ThirdParty'|'Custom',
                'provider': 'string',
                'version': 'string'
            },
            'settings': {
                'thirdPartyConfigurationUrl': 'string',
                'entityUrlTemplate': 'string',
                'executionUrlTemplate': 'string',
                'revisionUrlTemplate': 'string'
            },
            'actionConfigurationProperties': [
                {
                    'name': 'string',
```

```

        'required': True|False,
        'key': True|False,
        'secret': True|False,
        'queryable': True|False,
        'description': 'string',
        'type': 'String'|'Number'|'Boolean'
    },
],
'inputArtifactDetails': {
    'minimumCount': 123,
    'maximumCount': 123
},
'outputArtifactDetails': {
    'minimumCount': 123,
    'maximumCount': 123
}
},
],
'nextToken': 'string'
}

```

Response Structure

- (*dict*) –

Represents the output of a list action types action.

- **actionTypes** (*list*) –

Provides details of the action types.

- * (*dict*) –

Returns information about the details of an action type.

- **id** (*dict*) –

Represents information about an action type.

- **category** (*string*) –

A category defines what kind of action can be taken in the stage, and constrains the provider type for the action. Valid categories are limited to one of the values below.

- **owner** (*string*) –

The creator of the action being called.

- **provider** (*string*) –

The provider of the service being called by the action. Valid providers are determined by the action category. For example, an action in the Deploy category type might have a provider of AWS CodeDeploy, which would be specified as CodeDeploy.

- **version** (*string*) –

A string that identifies the action type.

- **settings** (*dict*) –

The settings for the action type.

- **thirdPartyConfigurationUrl** (*string*) –

The URL of a sign-up page where users can sign up for an external service and perform initial configuration of the action provided by that service.

- **entityUrlTemplate** (*string*) –

The URL returned to the AWS CodePipeline console that provides a deep link to the resources of the external system, such as the configuration page for an AWS CodeDeploy deployment group. This link is provided as part of the action display within the pipeline.

- **executionUrlTemplate** (*string*) –

The URL returned to the AWS CodePipeline console that contains a link to the top-level landing page for the external system, such as console page for AWS CodeDeploy. This link is shown on the pipeline view page in the AWS CodePipeline console and provides a link to the execution entity of the external action.

- **revisionUrlTemplate** (*string*) –

The URL returned to the AWS CodePipeline console that contains a link to the page where customers can update or change the configuration of the external action.

- **actionConfigurationProperties** (*list*) –

The configuration properties for the action type.

- (*dict*) –

Represents information about an action configuration property.

- **name** (*string*) –

The name of the action configuration property.

- **required** (*boolean*) –

Whether the configuration property is a required value.

- **key** (*boolean*) –

Whether the configuration property is a key.

- **secret** (*boolean*) –

Whether the configuration property is secret. Secrets are hidden from all calls except for `GetJobDetails`, `GetThirdPartyJobDetails`, `PollForJobs`, and `PollForThirdPartyJobs`.

When updating a pipeline, passing `*****` without changing any other values of the action will preserve the prior value of the secret.

- **queryable** (*boolean*) –

Indicates that the property will be used in conjunction with `PollForJobs`. When creating a custom action, an action can have up to one queryable property. If it has one, that property must be both required and not secret.

If you create a pipeline with a custom action type, and that custom action contains a queryable property, the value for that configuration property is subject to additional restrictions. The value must be less than or equal to twenty (20) characters. The value can contain only alphanumeric characters, underscores, and hyphens.

- **description** (*string*) –

The description of the action configuration property that will be displayed to users.

- **type** (*string*) –
The type of the configuration property.
- **inputArtifactDetails** (*dict*) –
The details of the input artifact for the action, such as its commit ID.
- **minimumCount** (*integer*) –
The minimum number of artifacts allowed for the action type.
- **maximumCount** (*integer*) –
The maximum number of artifacts allowed for the action type.
- **outputArtifactDetails** (*dict*) –
The details of the output artifact of the action, such as its commit ID.
- **minimumCount** (*integer*) –
The minimum number of artifacts allowed for the action type.
- **maximumCount** (*integer*) –
The maximum number of artifacts allowed for the action type.
- **nextToken** (*string*) –
If the amount of returned information is significantly large, an identifier is also returned which can be used in a subsequent list action types call to return the next set of action types in the list.

list_pipelines (***kwargs*)

Gets a summary of all of the pipelines associated with your account.

Request Syntax

```
response = client.list_pipelines(
    nextToken='string'
)
```

Parameters **nextToken** (*string*) – An identifier that was returned from the previous list pipelines call, which can be used to return the next set of pipelines in the list.

Return type dict

Returns

Response Syntax

```
{
    'pipelines': [
        {
            'name': 'string',
            'version': 123,
            'created': datetime(2015, 1, 1),
            'updated': datetime(2015, 1, 1)
        },
    ],
    'nextToken': 'string'
}
```

Response Structure

- (*dict*) –
Represents the output of a list pipelines action.

- **pipelines** (*list*) –

The list of pipelines.

- * (*dict*) –

Returns a summary of a pipeline.

- **name** (*string*) –

The name of the pipeline.

- **version** (*integer*) –

The version number of the pipeline.

- **created** (*datetime*) –

The date and time the pipeline was created, in timestamp format.

- **updated** (*datetime*) –

The date and time of the last update to the pipeline, in timestamp format.

- **nextToken** (*string*) –

If the amount of returned information is significantly large, an identifier is also returned which can be used in a subsequent list pipelines call to return the next set of pipelines in the list.

poll_for_jobs (***kwargs*)

Returns information about any jobs for AWS CodePipeline to act upon.

Warning: When this API is called, AWS CodePipeline returns temporary credentials for the Amazon S3 bucket used to store artifacts for the pipeline, if the action requires access to that Amazon S3 bucket for input or output artifacts. Additionally, this API returns any secret values defined for the action.

Request Syntax

```
response = client.poll_for_jobs(  
    actionTypes={  
        'category': 'Source'|'Build'|'Deploy'|'Test'|'Invoke',  
        'owner': 'AWS'|'ThirdParty'|'Custom',  
        'provider': 'string',  
        'version': 'string'  
    },  
    maxBatchSize=123,  
    queryParam={  
        'string': 'string'  
    }  
)
```

Parameters

- **actionTypeId** (*dict*) – **[REQUIRED]**

Represents information about an action type.

- **category** (*string*) – **[REQUIRED]**

A category defines what kind of action can be taken in the stage, and constrains the provider type for the action. Valid categories are limited to one of the values below.

- **owner** (*string*) – **[REQUIRED]**

The creator of the action being called.

– **provider** (*string*) – [REQUIRED]

The provider of the service being called by the action. Valid providers are determined by the action category. For example, an action in the Deploy category type might have a provider of AWS CodeDeploy, which would be specified as CodeDeploy.

– **version** (*string*) – [REQUIRED]

A string that identifies the action type.

- **maxBatchSize** (*integer*) – The maximum number of jobs to return in a poll for jobs call.
 - **queryParam** (*dict*) – A map of property names and values. For an action type with no queryable properties, this value must be null or an empty map. For an action type with a queryable property, you must supply that property as a key in the map. Only jobs whose action configuration matches the mapped value will be returned.
- (*string*) –
* (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'jobs': [
    {
      'id': 'string',
      'data': {
        'actionTypeId': {
          'category': 'Source'|'Build'|'Deploy'|'Test'|'Invoke',
          'owner': 'AWS'|'ThirdParty'|'Custom',
          'provider': 'string',
          'version': 'string'
        },
        'actionConfiguration': {
          'configuration': {
            'string': 'string'
          }
        },
        'pipelineContext': {
          'pipelineName': 'string',
          'stage': {
            'name': 'string'
          },
          'action': {
            'name': 'string'
          }
        },
        'inputArtifacts': [
          {
            'name': 'string',
            'revision': 'string',
            'location': {
              'type': 'S3',
              's3Location': {
                'bucketName': 'string',
                'objectKey': 'string'
              }
            }
          }
        ]
      }
    }
  ]
}
```

```
        },
    ],
    'outputArtifacts': [
        {
            'name': 'string',
            'revision': 'string',
            'location': {
                'type': 'S3',
                's3Location': {
                    'bucketName': 'string',
                    'objectKey': 'string'
                }
            }
        }
    ],
    'artifactCredentials': {
        'accessKeyId': 'string',
        'secretAccessKey': 'string',
        'sessionToken': 'string'
    },
    'continuationToken': 'string',
    'encryptionKey': {
        'id': 'string',
        'type': 'KMS'
    }
},
'nonce': 'string',
'accountId': 'string'
},
]
```

Response Structure

- (*dict*) –

Represents the output of a poll for jobs action.

- **jobs** (*list*) –

Information about the jobs to take action on.

- * (*dict*) –

Represents information about a job.

- **id** (*string*) –

The unique system-generated ID of the job.

- **data** (*dict*) –

Additional data about a job.

- **actionTypeId** (*dict*) –

Represents information about an action type.

- **category** (*string*) –

A category defines what kind of action can be taken in the stage, and constrains the provider type for the action. Valid categories are limited to one of the values below.

- **owner** (*string*) –

The creator of the action being called.

- **provider** (*string*) –
The provider of the service being called by the action. Valid providers are determined by the action category. For example, an action in the Deploy category type might have a provider of AWS CodeDeploy, which would be specified as CodeDeploy.
- **version** (*string*) –
A string that identifies the action type.
- **actionConfiguration** (*dict*) –
Represents information about an action configuration.
- **configuration** (*dict*) –
The configuration data for the action.
- (*string*) –
- (*string*) –
- **pipelineContext** (*dict*) –
Represents information about a pipeline to a job worker.
- **pipelineName** (*string*) –
The name of the pipeline. This is a user-specified value. Pipeline names must be unique across all pipeline names under an Amazon Web Services account.
- **stage** (*dict*) –
The stage of the pipeline.
- **name** (*string*) –
The name of the stage.
- **action** (*dict*) –
Represents the context of an action within the stage of a pipeline to a job worker.
- **name** (*string*) –
The name of the action within the context of a job.
- **inputArtifacts** (*list*) –
The artifact supplied to the job.
- (*dict*) –
Represents information about an artifact that will be worked upon by actions in the pipeline.
- **name** (*string*) –
The artifact's name.
- **revision** (*string*) –
The artifact's revision ID. Depending on the type of object, this could be a commit ID (GitHub) or a revision ID (Amazon S3).
- **location** (*dict*) –
The location of an artifact.
- **type** (*string*) –
The type of artifact in the location.
- **s3Location** (*dict*) –
The Amazon S3 bucket that contains the artifact.

- **bucketName** (*string*) –
The name of the Amazon S3 bucket.
- **objectKey** (*string*) –
The key of the object in the Amazon S3 bucket, which uniquely identifies the object in the bucket.
- **outputArtifacts** (*list*) –
The output of the job.
- (*dict*) –
Represents information about an artifact that will be worked upon by actions in the pipeline.
- **name** (*string*) –
The artifact's name.
- **revision** (*string*) –
The artifact's revision ID. Depending on the type of object, this could be a commit ID (GitHub) or a revision ID (Amazon S3).
- **location** (*dict*) –
The location of an artifact.
- **type** (*string*) –
The type of artifact in the location.
- **s3Location** (*dict*) –
The Amazon S3 bucket that contains the artifact.
- **bucketName** (*string*) –
The name of the Amazon S3 bucket.
- **objectKey** (*string*) –
The key of the object in the Amazon S3 bucket, which uniquely identifies the object in the bucket.
- **artifactCredentials** (*dict*) –
Represents an AWS session credentials object. These credentials are temporary credentials that are issued by AWS Secure Token Service (STS). They can be used to access input and output artifacts in the Amazon S3 bucket used to store artifact for the pipeline in AWS CodePipeline.
- **accessKeyId** (*string*) –
The access key for the session.
- **secretAccessKey** (*string*) –
The secret access key for the session.
- **sessionToken** (*string*) –
The token for the session.
- **continuationToken** (*string*) –
A system-generated token, such as a AWS CodeDeploy deployment ID, that a job requires in order to continue the job asynchronously.
- **encryptionKey** (*dict*) –

Represents information about the AWS Key Management Service (AWS KMS) key used to encrypt data in the artifact store.

- **id** (*string*) –

The ID of the AWS KMS key.

- **type** (*string*) –

The type of AWS KMS key, such as a customer master key.

- **nonce** (*string*) –

A system-generated random number that AWS CodePipeline uses to ensure that the job is being worked on by only one job worker. This number must be returned in the response.

- **accountId** (*string*) –

The ID of the AWS account to use when performing the job.

poll_for_third_party_jobs (***kwargs*)

Determines whether there are any third party jobs for a job worker to act on. Only used for partner actions.

Warning: When this API is called, AWS CodePipeline returns temporary credentials for the Amazon S3 bucket used to store artifacts for the pipeline, if the action requires access to that Amazon S3 bucket for input or output artifacts.

Request Syntax

```
response = client.poll_for_third_party_jobs(
    actionTypeId={
        'category': 'Source'|'Build'|'Deploy'|'Test'|'Invoke',
        'owner': 'AWS'|'ThirdParty'|'Custom',
        'provider': 'string',
        'version': 'string'
    },
    maxBatchSize=123
)
```

Parameters

- **actionTypeId** (*dict*) – **[REQUIRED]**

Represents information about an action type.

- **category** (*string*) – **[REQUIRED]**

A category defines what kind of action can be taken in the stage, and constrains the provider type for the action. Valid categories are limited to one of the values below.

- **owner** (*string*) – **[REQUIRED]**

The creator of the action being called.

- **provider** (*string*) – **[REQUIRED]**

The provider of the service being called by the action. Valid providers are determined by the action category. For example, an action in the Deploy category type might have a provider of AWS CodeDeploy, which would be specified as CodeDeploy.

- **version** (*string*) – **[REQUIRED]**

A string that identifies the action type.

- **maxBatchSize** (*integer*) – The maximum number of jobs to return in a poll for jobs call.

Return type dict

Returns

Response Syntax

```
{
  'jobs': [
    {
      'clientId': 'string',
      'jobId': 'string'
    },
  ]
}
```

Response Structure

- (*dict*) –

Represents the output of a poll for third party jobs action.

- **jobs** (*list*) –

Information about the jobs to take action on.

- * (*dict*) –

A response to a PollForThirdPartyJobs request returned by AWS CodePipeline when there is a job to be worked upon by a partner action.

- **clientId** (*string*) –

The clientToken portion of the clientId and clientToken pair used to verify that the calling entity is allowed access to the job and its details.

- **jobId** (*string*) –

The identifier used to identify the job in AWS CodePipeline.

put_action_revision (***kwargs*)

Provides information to AWS CodePipeline about new revisions to a source.

Request Syntax

```
response = client.put_action_revision(
    pipelineName='string',
    stageName='string',
    actionName='string',
    actionRevision={
        'revisionId': 'string',
        'revisionChangeId': 'string',
        'created': datetime(2015, 1, 1)
    }
)
```

Parameters

- **pipelineName** (*string*) – [REQUIRED]

The name of the pipeline that will start processing the revision to the source.

- **stageName** (*string*) – [REQUIRED]

The name of the stage that contains the action that will act upon the revision.

- **actionName** (*string*) – [REQUIRED]

The name of the action that will process the revision.

- **actionRevision** (*dict*) – [REQUIRED]

Represents information about the version (or revision) of an action.

- **revisionId** (*string*) – [REQUIRED]

The system-generated unique ID that identifies the revision number of the action.

- **revisionChangeId** (*string*) –

The unique identifier of the change that set the state to this revision, for example a deployment ID or timestamp.

- **created** (*datetime*) – [REQUIRED]

The date and time when the most recent version of the action was created, in timestamp format.

Return type dict

Returns

Response Syntax

```
{
    'newRevision': True|False,
    'pipelineExecutionId': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a put action revision action.

- **newRevision** (*boolean*) –

The new revision number or ID for the revision after the action completes.

- **pipelineExecutionId** (*string*) –

The ID of the current workflow state of the pipeline.

put_job_failure_result (***kwargs*)

Represents the failure of a job as returned to the pipeline by a job worker. Only used for custom actions.

Request Syntax

```
response = client.put_job_failure_result(
    jobId='string',
    failureDetails={
        'type': 'JobFailed'|'ConfigurationError'|'PermissionError'|'RevisionOutOfSync'|'Rev
        'message': 'string',
        'externalExecutionId': 'string'
    }
)
```

Parameters

- **jobId** (*string*) – [REQUIRED]

The unique system-generated ID of the job that failed. This is the same ID returned from PollForJobs.

- **failureDetails** (*dict*) – [REQUIRED]

The details about the failure of a job.

- **type** (*string*) – [REQUIRED]
The type of the failure.
- **message** (*string*) – [REQUIRED]
The message about the failure.
- **externalExecutionId** (*string*) –
The external ID of the run of the action that failed.

Returns None

put_job_success_result (***kwargs*)

Represents the success of a job as returned to the pipeline by a job worker. Only used for custom actions.

Request Syntax

```
response = client.put_job_success_result(  
    jobId='string',  
    currentRevision={  
        'revision': 'string',  
        'changeIdentifier': 'string'  
    },  
    continuationToken='string',  
    executionDetails={  
        'summary': 'string',  
        'externalExecutionId': 'string',  
        'percentComplete': 123  
    }  
)
```

Parameters

- **jobId** (*string*) – [REQUIRED]
The unique system-generated ID of the job that succeeded. This is the same ID returned from PollForJobs.
- **currentRevision** (*dict*) – The ID of the current revision of the artifact successfully worked upon by the job.
 - **revision** (*string*) – [REQUIRED]
The revision ID of the current version of an artifact.
 - **changeIdentifier** (*string*) – [REQUIRED]
The change identifier for the current revision.
- **continuationToken** (*string*) – A system-generated token, such as a AWS CodeDeploy deployment ID, that the successful job used to complete a job asynchronously.
- **executionDetails** (*dict*) – The execution details of the successful job, such as the actions taken by the job worker.
 - **summary** (*string*) –
The summary of the current status of the actions.
 - **externalExecutionId** (*string*) –
The system-generated unique ID of this action used to identify this job worker in any external systems, such as AWS CodeDeploy.
 - **percentComplete** (*integer*) –
The percentage of work completed on the action, represented on a scale of zero to one hundred percent.

Returns None

put_third_party_job_failure_result (**kwargs)

Represents the failure of a third party job as returned to the pipeline by a job worker. Only used for partner actions.

Request Syntax

```
response = client.put_third_party_job_failure_result(
    jobId='string',
    clientToken='string',
    failureDetails={
        'type': 'JobFailed'|'ConfigurationError'|'PermissionError'|'RevisionOutOfSync'|'Rev
        'message': 'string',
        'externalExecutionId': 'string'
    }
)
```

Parameters

- **jobId** (*string*) – **[REQUIRED]**

The ID of the job that failed. This is the same ID returned from PollForThirdPartyJobs.

- **clientToken** (*string*) – **[REQUIRED]**

The clientToken portion of the clientId and clientToken pair used to verify that the calling entity is allowed access to the job and its details.

- **failureDetails** (*dict*) – **[REQUIRED]**

Represents information about failure details.

- **type** (*string*) – **[REQUIRED]**

The type of the failure.

- **message** (*string*) – **[REQUIRED]**

The message about the failure.

- **externalExecutionId** (*string*) –

The external ID of the run of the action that failed.

Returns None

put_third_party_job_success_result (**kwargs)

Represents the success of a third party job as returned to the pipeline by a job worker. Only used for partner actions.

Request Syntax

```
response = client.put_third_party_job_success_result(
    jobId='string',
    clientToken='string',
    currentRevision={
        'revision': 'string',
        'changeIdentifier': 'string'
    },
    continuationToken='string',
    executionDetails={
        'summary': 'string',
        'externalExecutionId': 'string',
        'percentComplete': 123
    }
)
```

Parameters

- **jobId** (*string*) – [REQUIRED]

The ID of the job that successfully completed. This is the same ID returned from `PollForThirdPartyJobs`.

- **clientToken** (*string*) – [REQUIRED]

The `clientToken` portion of the `clientId` and `clientToken` pair used to verify that the calling entity is allowed access to the job and its details.

- **currentRevision** (*dict*) – Represents information about a current revision.
 - **revision** (*string*) – [REQUIRED]

The revision ID of the current version of an artifact.

- **changeIdentifier** (*string*) – [REQUIRED]

The change identifier for the current revision.

- **continuationToken** (*string*) – A system-generated token, such as a AWS CodeDeploy deployment ID, that a job uses in order to continue the job asynchronously.

- **executionDetails** (*dict*) – The details of the actions taken and results produced on an artifact as it passes through stages in the pipeline.

- **summary** (*string*) –

The summary of the current status of the actions.

- **externalExecutionId** (*string*) –

The system-generated unique ID of this action used to identify this job worker in any external systems, such as AWS CodeDeploy.

- **percentComplete** (*integer*) –

The percentage of work completed on the action, represented on a scale of zero to one hundred percent.

Returns None

start_pipeline_execution (***kwargs*)

Starts the specified pipeline. Specifically, it begins processing the latest commit to the source location specified as part of the pipeline.

Request Syntax

```
response = client.start_pipeline_execution(
    name='string'
)
```

Parameters **name** (*string*) – [REQUIRED]

The name of the pipeline to start.

Return type dict

Returns**Response Syntax**

```
{
    'pipelineExecutionId': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a start pipeline execution action.

– **pipelineExecutionId** (*string*) –

The unique system-generated ID of the pipeline that was started.

update_pipeline (***kwargs*)

Updates a specified pipeline with edits or changes to its structure. Use a JSON file with the pipeline structure in conjunction with UpdatePipeline to provide the full structure of the pipeline. Updating the pipeline increases the version number of the pipeline by 1.

Request Syntax

```
response = client.update_pipeline(
    pipeline={
        'name': 'string',
        'roleArn': 'string',
        'artifactStore': {
            'type': 'S3',
            'location': 'string',
            'encryptionKey': {
                'id': 'string',
                'type': 'KMS'
            }
        },
    },
    'stages': [
        {
            'name': 'string',
            'blockers': [
                {
                    'name': 'string',
                    'type': 'Schedule'
                },
            ],
        },
    ],
    'actions': [
        {
            'name': 'string',
            'actionTypeId': {
                'category': 'Source|Build|Deploy|Test|Invoke',
                'owner': 'AWS|ThirdParty|Custom',
                'provider': 'string',
                'version': 'string'
            },
            'runOrder': 123,
            'configuration': {
                'string': 'string'
            },
            'outputArtifacts': [
                {
                    'name': 'string'
                },
            ],
            'inputArtifacts': [
                {
                    'name': 'string'
                },
            ],
            'roleArn': 'string'
        },
    ],
)
```

```
    ],  
    'version': 123  
  }  
)
```

Parameters `pipeline` (*dict*) – [REQUIRED]

The name of the pipeline to be updated.

- **name** (*string*) – [REQUIRED]

The name of the action to be performed.

- **roleArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) for AWS CodePipeline to use to either perform actions with no `actionRoleArn`, or to use to assume roles for actions with an `actionRoleArn`.

- **artifactStore** (*dict*) – [REQUIRED]

The Amazon S3 location where artifacts are stored for the pipeline. If this Amazon S3 bucket is created manually, it must meet the requirements for AWS CodePipeline. For more information, see the Concepts.

- **type** (*string*) – [REQUIRED]

The type of the artifact store, such as S3.

- **location** (*string*) – [REQUIRED]

The location for storing the artifacts for a pipeline, such as an S3 bucket or folder.

- **encryptionKey** (*dict*) –

The AWS Key Management Service (AWS KMS) key used to encrypt the data in the artifact store. If this is undefined, the default key for Amazon S3 is used.

- * **id** (*string*) – [REQUIRED]

The ID of the AWS KMS key.

- * **type** (*string*) – [REQUIRED]

The type of AWS KMS key, such as a customer master key.

- **stages** (*list*) – [REQUIRED]

The stage in which to perform the action.

- (*dict*) –

Represents information about a stage and its definition.

- * **name** (*string*) – [REQUIRED]

The name of the stage.

- * **blockers** (*list*) –

The gates included in a stage.

- (*dict*) –

Represents information about a gate declaration.

- **name** (*string*) – [REQUIRED]

The name of the gate declaration.

- **type** (*string*) – [REQUIRED]

The type of the gate declaration.

* **actions** (*list*) – [REQUIRED]

The actions included in a stage.

· (*dict*) –

Represents information about an action declaration.

· **name** (*string*) – [REQUIRED]

The action declaration's name.

· **actionTypeId** (*dict*) – [REQUIRED]

The configuration information for the action type.

· **category** (*string*) – [REQUIRED]

A category defines what kind of action can be taken in the stage, and constrains the provider type for the action. Valid categories are limited to one of the values below.

· **owner** (*string*) – [REQUIRED]

The creator of the action being called.

· **provider** (*string*) – [REQUIRED]

The provider of the service being called by the action. Valid providers are determined by the action category. For example, an action in the Deploy category type might have a provider of AWS CodeDeploy, which would be specified as CodeDeploy.

· **version** (*string*) – [REQUIRED]

A string that identifies the action type.

· **runOrder** (*integer*) –

The order in which actions are run.

· **configuration** (*dict*) –

The action declaration's configuration.

· (*string*) –

· (*string*) –

· **outputArtifacts** (*list*) –

The name or ID of the result of the action declaration, such as a test or build artifact.

· (*dict*) –

Represents information about the output of an action.

· **name** (*string*) – [REQUIRED]

The name of the output of an artifact, such as "My App".

The input artifact of an action must exactly match the output artifact declared in a preceding action, but the input artifact does not have to be the next action in strict sequence from the action that provided the output artifact. Actions in parallel can declare different output artifacts, which are in turn consumed by different following actions.

Output artifact names must be unique within a pipeline.

· **inputArtifacts** (*list*) –

The name or ID of the artifact consumed by the action, such as a test or build artifact.

· (*dict*) –

Represents information about an artifact to be worked on, such as a test or build artifact.

- **name** (*string*) – [REQUIRED]

The name of the artifact to be worked on, for example, “My App”.

The input artifact of an action must exactly match the output artifact declared in a preceding action, but the input artifact does not have to be the next action in strict sequence from the action that provided the output artifact. Actions in parallel can declare different output artifacts, which are in turn consumed by different following actions.

- **roleArn** (*string*) –

The ARN of the IAM service role that will perform the declared action. This is assumed through the roleArn for the pipeline.

- **version** (*integer*) –

The version number of the pipeline. A new pipeline always has a version number of 1. This number is automatically incremented when a pipeline is updated.

Return type dict

Returns

Response Syntax

```
{
  'pipeline': {
    'name': 'string',
    'roleArn': 'string',
    'artifactStore': {
      'type': 'S3',
      'location': 'string',
      'encryptionKey': {
        'id': 'string',
        'type': 'KMS'
      }
    }
  },
  'stages': [
    {
      'name': 'string',
      'blockers': [
        {
          'name': 'string',
          'type': 'Schedule'
        }
      ],
      'actions': [
        {
          'name': 'string',
          'actionTypeId': {
            'category': 'Source'|'Build'|'Deploy'|'Test'|'Invoke',
            'owner': 'AWS'|'ThirdParty'|'Custom',
            'provider': 'string',
            'version': 'string'
          },
          'runOrder': 123,
          'configuration': {
```

```

        'string': 'string'
    },
    'outputArtifacts': [
        {
            'name': 'string'
        },
    ],
    'inputArtifacts': [
        {
            'name': 'string'
        },
    ],
    'roleArn': 'string'
    },
]
},
],
'version': 123
}

```

Response Structure

- (*dict*) –

Represents the output of an update pipeline action.

- **pipeline** (*dict*) –

The structure of the updated pipeline.

- * **name** (*string*) –

The name of the action to be performed.

- * **roleArn** (*string*) –

The Amazon Resource Name (ARN) for AWS CodePipeline to use to either perform actions with no `actionRoleArn`, or to use to assume roles for actions with an `actionRoleArn`.

- * **artifactStore** (*dict*) –

The Amazon S3 location where artifacts are stored for the pipeline. If this Amazon S3 bucket is created manually, it must meet the requirements for AWS CodePipeline. For more information, see the Concepts.

- **type** (*string*) –

The type of the artifact store, such as S3.

- **location** (*string*) –

The location for storing the artifacts for a pipeline, such as an S3 bucket or folder.

- **encryptionKey** (*dict*) –

The AWS Key Management Service (AWS KMS) key used to encrypt the data in the artifact store. If this is undefined, the default key for Amazon S3 is used.

- **id** (*string*) –

The ID of the AWS KMS key.

- **type** (*string*) –

The type of AWS KMS key, such as a customer master key.

* **stages** (*list*) –

The stage in which to perform the action.

· (*dict*) –

Represents information about a stage and its definition.

· **name** (*string*) –

The name of the stage.

· **blockers** (*list*) –

The gates included in a stage.

· (*dict*) –

Represents information about a gate declaration.

· **name** (*string*) –

The name of the gate declaration.

· **type** (*string*) –

The type of the gate declaration.

· **actions** (*list*) –

The actions included in a stage.

· (*dict*) –

Represents information about an action declaration.

· **name** (*string*) –

The action declaration's name.

· **actionTypeId** (*dict*) –

The configuration information for the action type.

· **category** (*string*) –

A category defines what kind of action can be taken in the stage, and constrains the provider type for the action. Valid categories are limited to one of the values below.

· **owner** (*string*) –

The creator of the action being called.

· **provider** (*string*) –

The provider of the service being called by the action. Valid providers are determined by the action category. For example, an action in the Deploy category type might have a provider of AWS CodeDeploy, which would be specified as CodeDeploy.

· **version** (*string*) –

A string that identifies the action type.

· **runOrder** (*integer*) –

The order in which actions are run.

· **configuration** (*dict*) –

The action declaration's configuration.

· (*string*) –

· (*string*) –

· **outputArtifacts** (*list*) –

The name or ID of the result of the action declaration, such as a test or build artifact.

- *(dict)* –

Represents information about the output of an action.

- **name** (*string*) –

The name of the output of an artifact, such as “My App”.

The input artifact of an action must exactly match the output artifact declared in a preceding action, but the input artifact does not have to be the next action in strict sequence from the action that provided the output artifact. Actions in parallel can declare different output artifacts, which are in turn consumed by different following actions.

Output artifact names must be unique within a pipeline.

- **inputArtifacts** (*list*) –

The name or ID of the artifact consumed by the action, such as a test or build artifact.

- *(dict)* –

Represents information about an artifact to be worked on, such as a test or build artifact.

- **name** (*string*) –

The name of the artifact to be worked on, for example, “My App”.

The input artifact of an action must exactly match the output artifact declared in a preceding action, but the input artifact does not have to be the next action in strict sequence from the action that provided the output artifact. Actions in parallel can declare different output artifacts, which are in turn consumed by different following actions.

- **roleArn** (*string*) –

The ARN of the IAM service role that will perform the declared action. This is assumed through the roleArn for the pipeline.

- * **version** (*integer*) –

The version number of the pipeline. A new pipeline always has a version number of 1. This number is automatically incremented when a pipeline is updated.

CognitoIdentity

Table of Contents

- CognitoIdentity
 - Client

Client

class CognitoIdentity.**Client**

A low-level client representing Amazon Cognito Identity:

```
import boto3
```

```
client = boto3.client('cognito-identity')
```

These are the available methods:

- `can_paginate()`
- `create_identity_pool()`
- `delete_identities()`
- `delete_identity_pool()`
- `describe_identity()`
- `describe_identity_pool()`
- `generate_presigned_url()`
- `get_credentials_for_identity()`
- `get_id()`
- `get_identity_pool_roles()`
- `get_open_id_token()`
- `get_open_id_token_for_developer_identity()`
- `get_paginator()`
- `get_waiter()`
- `list_identities()`
- `list_identity_pools()`
- `lookup_developer_identity()`
- `merge_developer_identities()`
- `set_identity_pool_roles()`
- `unlink_developer_identity()`
- `unlink_identity()`
- `update_identity_pool()`

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you’d normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

create_identity_pool (***kwargs*)

Creates a new identity pool. The identity pool is a store of user identity information that is specific to your AWS account. The limit on identity pools is 60 per account. The keys for `SupportedLoginProviders` are as follows:

- Facebook: `graph.facebook.com`
- Google: `accounts.google.com`
- Amazon: `www.amazon.com`
- Twitter: `api.twitter.com`
- Digits: `www.digits.com`

You must use AWS Developer credentials to call this API.

Request Syntax

```
response = client.create_identity_pool(
    IdentityPoolName='string',
    AllowUnauthenticatedIdentities=True|False,
    SupportedLoginProviders={
        'string': 'string'
    },

```



```

DeveloperProviderName='string',
OpenIdConnectProviderARNs=[
    'string',
]
)

```

Parameters

- **IdentityPoolName** (*string*) – [REQUIRED]
A string that you provide.
- **AllowUnauthenticatedIdentities** (*boolean*) – [REQUIRED]
TRUE if the identity pool supports unauthenticated logins.
- **SupportedLoginProviders** (*dict*) – Optional key:value pairs mapping provider names to provider app IDs.
 - (*string*) –
 - * (*string*) –
- **DeveloperProviderName** (*string*) – The “domain” by which Cognito will refer to your users. This name acts as a placeholder that allows your backend and the Cognito service to communicate about the developer provider. For the `DeveloperProviderName`, you can use letters as well as period (.), underscore (_), and dash (-).

Once you have set a developer provider name, you cannot change it. Please take care in setting this parameter.
- **OpenIdConnectProviderARNs** (*list*) – A list of OpenID Connect provider ARNs.
 - (*string*) –

Return type dict

Returns

Response Syntax

```

{
    'IdentityPoolId': 'string',
    'IdentityPoolName': 'string',
    'AllowUnauthenticatedIdentities': True|False,
    'SupportedLoginProviders': {
        'string': 'string'
    },
    'DeveloperProviderName': 'string',
    'OpenIdConnectProviderARNs': [
        'string',
    ]
}

```

Response Structure

- (*dict*) – An object representing a Cognito identity pool.
 - **IdentityPoolId** (*string*) – An identity pool ID in the format REGION:GUID.
 - **IdentityPoolName** (*string*) –
A string that you provide.
 - **AllowUnauthenticatedIdentities** (*boolean*) – TRUE if the identity pool supports unauthenticated logins.
 - **SupportedLoginProviders** (*dict*) –
Optional key:value pairs mapping provider names to provider app IDs.

- * (*string*) –
- (*string*) –
- **DeveloperProviderName** (*string*) –
The “domain” by which Cognito will refer to your users.
- **OpenIdConnectProviderARNs** (*list*) –
A list of OpenID Connect provider ARNs.
- * (*string*) –

delete_identities (***kwargs*)

Deletes identities from an identity pool. You can specify a list of 1-60 identities that you want to delete.

You must use AWS Developer credentials to call this API.

Request Syntax

```
response = client.delete_identities(  
    IdentityIdsToDelete=[  
        'string',  
    ]  
)
```

Parameters **IdentityIdsToDelete** (*list*) – [REQUIRED]

A list of 1-60 identities that you want to delete.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{  
    'UnprocessedIdentityIds': [  
        {  
            'IdentityId': 'string',  
            'ErrorCode': 'AccessDenied'|'InternalServerError'  
        },  
    ]  
}
```

Response Structure

- (*dict*) –

Returned in response to a successful DeleteIdentities operation.

– **UnprocessedIdentityIds** (*list*) –

An array of UnprocessedIdentityId objects, each of which contains an ErrorCode and IdentityId.

- * (*dict*) –

An array of UnprocessedIdentityId objects, each of which contains an ErrorCode and IdentityId.

- **IdentityId** (*string*) –

A unique identifier in the format REGION:GUID.

- **ErrorCode** (*string*) –

The error code indicating the type of error that occurred.

delete_identity_pool (**kwargs)

Deletes a user pool. Once a pool is deleted, users will not be able to authenticate with the pool.

You must use AWS Developer credentials to call this API.

Request Syntax

```
response = client.delete_identity_pool(
    IdentityPoolId='string'
)
```

Parameters **IdentityPoolId** (*string*) – **[REQUIRED]** An identity pool ID in the format REGION:GUID.

Returns None

describe_identity (**kwargs)

Returns metadata related to the given identity, including when the identity was created and any associated linked logins.

You must use AWS Developer credentials to call this API.

Request Syntax

```
response = client.describe_identity(
    IdentityId='string'
)
```

Parameters **IdentityId** (*string*) – **[REQUIRED]**

A unique identifier in the format REGION:GUID.

Return type dict

Returns

Response Syntax

```
{
    'IdentityId': 'string',
    'Logins': [
        'string',
    ],
    'CreationDate': datetime(2015, 1, 1),
    'LastModifiedDate': datetime(2015, 1, 1)
}
```

Response Structure

- (*dict*) – A description of the identity.
 - **IdentityId** (*string*) – A unique identifier in the format REGION:GUID.
 - **Logins** (*list*) – A set of optional name-value pairs that map provider names to provider tokens.
 - * (*string*) –
 - **CreationDate** (*datetime*) –
 - Date on which the identity was created.
 - **LastModifiedDate** (*datetime*) –
 - Date on which the identity was last modified.

describe_identity_pool (**kwargs)

Gets details about a particular identity pool, including the pool name, ID description, creation date, and current number of users.

You must use AWS Developer credentials to call this API.

Request Syntax

```
response = client.describe_identity_pool(
    IdentityPoolId='string'
)
```

Parameters **IdentityPoolId** (*string*) – **[REQUIRED]** An identity pool ID in the format REGION:GUID.

Return type dict

Returns

Response Syntax

```
{
    'IdentityPoolId': 'string',
    'IdentityPoolName': 'string',
    'AllowUnauthenticatedIdentities': True|False,
    'SupportedLoginProviders': {
        'string': 'string'
    },
    'DeveloperProviderName': 'string',
    'OpenIdConnectProviderARNs': [
        'string',
    ]
}
```

Response Structure

- (*dict*) – An object representing a Cognito identity pool.
 - **IdentityPoolId** (*string*) – An identity pool ID in the format REGION:GUID.
 - **IdentityPoolName** (*string*) –
A string that you provide.
 - **AllowUnauthenticatedIdentities** (*boolean*) – TRUE if the identity pool supports unauthenticated logins.
 - **SupportedLoginProviders** (*dict*) –
Optional key:value pairs mapping provider names to provider app IDs.
 - * (*string*) –
 - (*string*) –
 - **DeveloperProviderName** (*string*) –
The “domain” by which Cognito will refer to your users.
 - **OpenIdConnectProviderARNs** (*list*) –
A list of OpenID Connect provider ARNs.
 - * (*string*) –

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to *ClientMethod*.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method’s model.

Returns The presigned url

get_credentials_for_identity (**kwargs)

Returns credentials for the provided identity ID. Any provided logins will be validated against supported login providers. If the token is for cognito-identity.amazonaws.com, it will be passed through to AWS Security Token Service with the appropriate role for the token.

This is a public API. You do not need any credentials to call this API.

Request Syntax

```
response = client.get_credentials_for_identity(
    IdentityId='string',
    Logins={
        'string': 'string'
    }
)
```

Parameters

- **IdentityId** (*string*) – [REQUIRED]
A unique identifier in the format REGION:GUID.
- **Logins** (*dict*) – A set of optional name-value pairs that map provider names to provider tokens.
 - (*string*) –
 - * (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'IdentityId': 'string',
    'Credentials': {
        'AccessKeyId': 'string',
        'SecretKey': 'string',
        'SessionToken': 'string',
        'Expiration': datetime(2015, 1, 1)
    }
}
```

Response Structure

- (*dict*) –
Returned in response to a successful GetCredentialsForIdentity operation.
 - **IdentityId** (*string*) –
A unique identifier in the format REGION:GUID.
 - **Credentials** (*dict*) –
Credentials for the provided identity ID.
 - * **AccessKeyId** (*string*) –
The Access Key portion of the credentials.
 - * **SecretKey** (*string*) –
The Secret Access Key portion of the credentials
 - * **SessionToken** (*string*) –
The Session Token portion of the credentials

* **Expiration** (*datetime*) –

The date at which these credentials will expire.

get_id (***kwargs*)

Generates (or retrieves) a Cognito ID. Supplying multiple logins will create an implicit linked account.

This is a public API. You do not need any credentials to call this API.

Request Syntax

```
response = client.get_id(
    AccountId='string',
    IdentityPoolId='string',
    Logins={
        'string': 'string'
    }
)
```

Parameters

- **AccountId** (*string*) – A standard AWS account ID (9+ digits).
- **IdentityPoolId** (*string*) – **[REQUIRED]** An identity pool ID in the format REGION:GUID.
- **Logins** (*dict*) – A set of optional name-value pairs that map provider names to provider tokens.

The available provider names for `Logins` are as follows:

- Facebook: `graph.facebook.com`
- Google: `accounts.google.com`
- Amazon: `www.amazon.com`
- Twitter: `api.twitter.com`
- Digits: `www.digits.com`
- (*string*) –
 - * (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'IdentityId': 'string'
}
```

Response Structure

- (*dict*) – Returned in response to a `GetId` request.
 - **IdentityId** (*string*) – A unique identifier in the format REGION:GUID.

get_identity_pool_roles (***kwargs*)

Gets the roles for an identity pool.

You must use AWS Developer credentials to call this API.

Request Syntax

```
response = client.get_identity_pool_roles(
    IdentityPoolId='string'
)
```

Parameters `IdentityPoolId` (*string*) – [REQUIRED]

An identity pool ID in the format REGION:GUID.

Return type dict**Returns****Response Syntax**

```
{
  'IdentityPoolId': 'string',
  'Roles': {
    'string': 'string'
  }
}
```

Response Structure

- (*dict*) –

Returned in response to a successful `GetIdentityPoolRoles` operation.

- **IdentityPoolId** (*string*) –

An identity pool ID in the format REGION:GUID.

- **Roles** (*dict*) –

The map of roles associated with this pool. Currently only authenticated and unauthenticated roles are supported.

- * (*string*) –
- (*string*) –

get_open_id_token (***kwargs*)

Gets an OpenID token, using a known Cognito ID. This known Cognito ID is returned by `GetId`. You can optionally add additional logins for the identity. Supplying multiple logins creates an implicit link.

The OpenId token is valid for 15 minutes.

This is a public API. You do not need any credentials to call this API.

Request Syntax

```
response = client.get_open_id_token(
    IdentityId='string',
    Logins={
        'string': 'string'
    }
)
```

Parameters

- **IdentityId** (*string*) – [REQUIRED] A unique identifier in the format REGION:GUID.
- **Logins** (*dict*) – A set of optional name-value pairs that map provider names to provider tokens. When using `graph.facebook.com` and `www.amazon.com`, supply the `access_token` returned from the provider's authflow. For `accounts.google.com` or any other OpenId Connect provider, always include the `id_token`.
 - (*string*) –
 - * (*string*) –

Return type dict**Returns****Response Syntax**

```
{
    'IdentityId': 'string',
    'Token': 'string'
}
```

Response Structure

- (*dict*) – Returned in response to a successful `GetOpenIdToken` request.
 - **IdentityId** (*string*) – A unique identifier in the format `REGION:GUID`. Note that the `IdentityId` returned may not match the one passed on input.
 - **Token** (*string*) – An OpenID token, valid for 15 minutes.

`get_open_id_token_for_developer_identity` (***kwargs*)

Registers (or retrieves) a Cognito `IdentityId` and an OpenID Connect token for a user authenticated by your backend authentication process. Supplying multiple logins will create an implicit linked account. You can only specify one developer provider as part of the `Logins` map, which is linked to the identity pool. The developer provider is the “domain” by which Cognito will refer to your users.

You can use `GetOpenIdTokenForDeveloperIdentity` to create a new identity and to link new logins (that is, user credentials issued by a public provider or developer provider) to an existing identity. When you want to create a new identity, the `IdentityId` should be null. When you want to associate a new login with an existing authenticated/unauthenticated identity, you can do so by providing the existing `IdentityId`. This API will create the identity in the specified `IdentityPoolId`.

You must use AWS Developer credentials to call this API.

Request Syntax

```
response = client.get_open_id_token_for_developer_identity(
    IdentityPoolId='string',
    IdentityId='string',
    Logins={
        'string': 'string'
    },
    TokenDuration=123
)
```

Parameters

- **IdentityPoolId** (*string*) – [REQUIRED]

An identity pool ID in the format `REGION:GUID`.

- **IdentityId** (*string*) – A unique identifier in the format `REGION:GUID`.
- **Logins** (*dict*) – [REQUIRED]

A set of optional name-value pairs that map provider names to provider tokens. Each name-value pair represents a user from a public provider or developer provider. If the user is from a developer provider, the name-value pair will follow the syntax `"developer_provider_name": "developer_user_identifier"`. The developer provider is the “domain” by which Cognito will refer to your users; you provided this domain while creating/updating the identity pool. The developer user identifier is an identifier from your backend that uniquely identifies a user. When you create an identity pool, you can specify the supported logins.

- (*string*) –
 - * (*string*) –
- **TokenDuration** (*integer*) – The expiration time of the token, in seconds. You can specify a custom expiration time for the token so that you can cache it. If you don’t provide an expiration time, the token is valid for 15 minutes. You can

exchange the token with Amazon STS for temporary AWS credentials, which are valid for a maximum of one hour. The maximum token duration you can set is 24 hours. You should take care in setting the expiration time for a token, as there are significant security implications: an attacker could use a leaked token to access your AWS resources for the token's duration.

Return type dict

Returns

Response Syntax

```
{
    'IdentityId': 'string',
    'Token': 'string'
}
```

Response Structure

- (*dict*) –

Returned in response to a successful `GetOpenIdTokenForDeveloperIdentity` request.

- **IdentityId** (*string*) –

A unique identifier in the format REGION:GUID.

- **Token** (*string*) –

An OpenID token.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type L[botocore.paginate.Paginator]

Returns A paginator object.

get_waiter (*waiter_name*)

list_identities (***kwargs*)

Lists the identities in a pool.

You must use AWS Developer credentials to call this API.

Request Syntax

```
response = client.list_identities(
    IdentityPoolId='string',
    MaxResults=123,
    NextToken='string',
    HideDisabled=True|False
)
```

Parameters

- **IdentityPoolId** (*string*) – [REQUIRED] An identity pool ID in the format REGION:GUID.

- **MaxResults** (*integer*) – **[REQUIRED]** The maximum number of identities to return.
- **NextToken** (*string*) – A pagination token.
- **HideDisabled** (*boolean*) – An optional boolean parameter that allows you to hide disabled identities. If omitted, the ListIdentities API will include disabled identities in the response.

Return type dict

Returns

Response Syntax

```
{
  'IdentityPoolId': 'string',
  'Identities': [
    {
      'IdentityId': 'string',
      'Logins': [
        'string',
      ],
      'CreationDate': datetime(2015, 1, 1),
      'LastModifiedDate': datetime(2015, 1, 1)
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) – The response to a ListIdentities request.
 - **IdentityPoolId** (*string*) – An identity pool ID in the format REGION:GUID.
 - **Identities** (*list*) – An object containing a set of identities and associated mappings.
 - * (*dict*) – A description of the identity.
 - **IdentityId** (*string*) – A unique identifier in the format REGION:GUID.
 - **Logins** (*list*) – A set of optional name-value pairs that map provider names to provider tokens.
 - (*string*) –
 - **CreationDate** (*datetime*) –
Date on which the identity was created.
 - **LastModifiedDate** (*datetime*) –
Date on which the identity was last modified.
 - **NextToken** (*string*) – A pagination token.

list_identity_pools (***kwargs*)

Lists all of the Cognito identity pools registered for your account.

You must use AWS Developer credentials to call this API.

Request Syntax

```
response = client.list_identity_pools(
    MaxResults=123,
    NextToken='string'
)
```

Parameters

- **MaxResults** (*integer*) – [REQUIRED] The maximum number of identities to return.
- **NextToken** (*string*) – A pagination token.

Return type dict

Returns

Response Syntax

```
{
  'IdentityPools': [
    {
      'IdentityPoolId': 'string',
      'IdentityPoolName': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) – The result of a successful ListIdentityPools action.
 - **IdentityPools** (*list*) – The identity pools returned by the ListIdentityPools action.
 - * (*dict*) – A description of the identity pool.
 - **IdentityPoolId** (*string*) – An identity pool ID in the format REGION:GUID.
 - **IdentityPoolName** (*string*) – A string that you provide.
 - **NextToken** (*string*) – A pagination token.

lookup_developer_identity (**kwargs)

Retrieves the IdentityID associated with a DeveloperUserIdentifier or the list of DeveloperUserIdentifier s associated with an IdentityId for an existing identity. Either IdentityID or DeveloperUserIdentifier must not be null. If you supply only one of these values, the other value will be searched in the database and returned as a part of the response. If you supply both, DeveloperUserIdentifier will be matched against IdentityID . If the values are verified against the database, the response returns both values and is the same as the request. Otherwise a ResourceConflictException is thrown.

You must use AWS Developer credentials to call this API.

Request Syntax

```
response = client.lookup_developer_identity(
    IdentityPoolId='string',
    IdentityId='string',
    DeveloperUserIdentifier='string',
    MaxResults=123,
    NextToken='string'
)
```

Parameters

- **IdentityPoolId** (*string*) – [REQUIRED]
An identity pool ID in the format REGION:GUID.
- **IdentityId** (*string*) – A unique identifier in the format REGION:GUID.
- **DeveloperUserIdentifier** (*string*) – A unique ID used by your backend authentication process to identify a user. Typically, a developer identity provider would issue many developer user identifiers, in keeping with the number of users.
- **MaxResults** (*integer*) – The maximum number of identities to return.

- **NextToken** (*string*) – A pagination token. The first call you make will have `NextToken` set to null. After that the service will return `NextToken` values as needed. For example, let's say you make a request with `MaxResults` set to 10, and there are 20 matches in the database. The service will return a pagination token as a part of the response. This token can be used to call the API again and get results starting from the 11th match.

Return type dict

Returns

Response Syntax

```
{
    'IdentityId': 'string',
    'DeveloperUserIdentifierList': [
        'string',
    ],
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Returned in response to a successful `LookupDeveloperIdentity` action.

- **IdentityId** (*string*) –

A unique identifier in the format `REGION:GUID`.

- **DeveloperUserIdentifierList** (*list*) –

This is the list of developer user identifiers associated with an identity ID. Cognito supports the association of multiple developer user identifiers with an identity ID.

* (*string*) –

- **NextToken** (*string*) –

A pagination token. The first call you make will have `NextToken` set to null. After that the service will return `NextToken` values as needed. For example, let's say you make a request with `MaxResults` set to 10, and there are 20 matches in the database. The service will return a pagination token as a part of the response. This token can be used to call the API again and get results starting from the 11th match.

`merge_developer_identities` (***kwargs*)

Merges two users having different `IdentityId`s, existing in the same identity pool, and identified by the same developer provider. You can use this action to request that discrete users be merged and identified as a single user in the Cognito environment. Cognito associates the given source user (`SourceUserIdentifier`) with the `IdentityId` of the `DestinationUserIdentifier`. Only developer-authenticated users can be merged. If the users to be merged are associated with the same public provider, but as two different users, an exception will be thrown.

You must use AWS Developer credentials to call this API.

Request Syntax

```
response = client.merge_developer_identities(
    SourceUserIdentifier='string',
    DestinationUserIdentifier='string',
    DeveloperProviderName='string',
    IdentityPoolId='string'
)
```

Parameters

- **SourceUserIdentifier** (*string*) – [REQUIRED]
User identifier for the source user. The value should be a `DeveloperUserIdentifier`.
- **DestinationUserIdentifier** (*string*) – [REQUIRED]
User identifier for the destination user. The value should be a `DeveloperUserIdentifier`.
- **DeveloperProviderName** (*string*) – [REQUIRED]
The “domain” by which Cognito will refer to your users. This is a (pseudo) domain name that you provide while creating an identity pool. This name acts as a placeholder that allows your backend and the Cognito service to communicate about the developer provider. For the `DeveloperProviderName`, you can use letters as well as period (.), underscore (_), and dash (-).
- **IdentityPoolId** (*string*) – [REQUIRED]
An identity pool ID in the format REGION:GUID.

Return type dict**Returns****Response Syntax**

```
{
    'IdentityId': 'string'
}
```

Response Structure

- (*dict*) –
Returned in response to a successful `MergeDeveloperIdentities` action.
 - **IdentityId** (*string*) –
A unique identifier in the format REGION:GUID.

set_identity_pool_roles (***kwargs*)

Sets the roles for an identity pool. These roles are used when making calls to `GetCredentialsForIdentity` action.

You must use AWS Developer credentials to call this API.

Request Syntax

```
response = client.set_identity_pool_roles(
    IdentityPoolId='string',
    Roles={
        'string': 'string'
    }
)
```

Parameters

- **IdentityPoolId** (*string*) – [REQUIRED]
An identity pool ID in the format REGION:GUID.
- **Roles** (*dict*) – [REQUIRED]
The map of roles associated with this pool. For a given role, the key will be either “authenticated” or “unauthenticated” and the value will be the Role ARN.
 - (*string*) –
 - * (*string*) –

Returns None

unlink_developer_identity (**kwargs)

Unlinks a `DeveloperUserIdentifier` from an existing identity. Unlinked developer users will be considered new identities next time they are seen. If, for a given Cognito identity, you remove all federated identities as well as the developer user identifier, the Cognito identity becomes inaccessible.

You must use AWS Developer credentials to call this API.

Request Syntax

```
response = client.unlink_developer_identity(
    IdentityId='string',
    IdentityPoolId='string',
    DeveloperProviderName='string',
    DeveloperUserIdentifier='string'
)
```

Parameters

- **IdentityId** (*string*) – [REQUIRED]
A unique identifier in the format REGION:GUID.
- **IdentityPoolId** (*string*) – [REQUIRED]
An identity pool ID in the format REGION:GUID.
- **DeveloperProviderName** (*string*) – [REQUIRED]
The “domain” by which Cognito will refer to your users.
- **DeveloperUserIdentifier** (*string*) – [REQUIRED] A unique ID used by your backend authentication process to identify a user.

Returns None

unlink_identity (**kwargs)

Unlinks a federated identity from an existing account. Unlinked logins will be considered new identities next time they are seen. Removing the last linked login will make this identity inaccessible.

This is a public API. You do not need any credentials to call this API.

Request Syntax

```
response = client.unlink_identity(
    IdentityId='string',
    Logins={
        'string': 'string'
    },
    LoginsToRemove=[
        'string',
    ]
)
```

Parameters

- **IdentityId** (*string*) – [REQUIRED] A unique identifier in the format REGION:GUID.
- **Logins** (*dict*) – [REQUIRED] A set of optional name-value pairs that map provider names to provider tokens.
 - (*string*) –
 - * (*string*) –
- **LoginsToRemove** (*list*) – [REQUIRED] Provider names to unlink from this identity.
 - (*string*) –

Returns None**update_identity_pool** (**kwargs)

Updates a user pool.

You must use AWS Developer credentials to call this API.

Request Syntax

```
response = client.update_identity_pool(
    IdentityPoolId='string',
    IdentityPoolName='string',
    AllowUnauthenticatedIdentities=True|False,
    SupportedLoginProviders={
        'string': 'string'
    },
    DeveloperProviderName='string',
    OpenIdConnectProviderARNs=[
        'string',
    ]
)
```

Parameters

- **IdentityPoolId** (*string*) – **[REQUIRED]** An identity pool ID in the format REGION:GUID.
- **IdentityPoolName** (*string*) – **[REQUIRED]**
A string that you provide.
- **AllowUnauthenticatedIdentities** (*boolean*) – **[REQUIRED]** TRUE if the identity pool supports unauthenticated logins.
- **SupportedLoginProviders** (*dict*) – Optional key:value pairs mapping provider names to provider app IDs.
– (*string*) –
 * (*string*) –
- **DeveloperProviderName** (*string*) – The “domain” by which Cognito will refer to your users.
- **OpenIdConnectProviderARNs** (*list*) – A list of OpenID Connect provider ARNs.
– (*string*) –

Return type dict

Returns**Response Syntax**

```
{
    'IdentityPoolId': 'string',
    'IdentityPoolName': 'string',
    'AllowUnauthenticatedIdentities': True|False,
    'SupportedLoginProviders': {
        'string': 'string'
    },
    'DeveloperProviderName': 'string',
    'OpenIdConnectProviderARNs': [
        'string',
    ]
}
```

Response Structure

- (*dict*) – An object representing a Cognito identity pool.

- **IdentityPoolId** (*string*) – An identity pool ID in the format REGION:GUID.
- **IdentityPoolName** (*string*) –
A string that you provide.
- **AllowUnauthenticatedIdentities** (*boolean*) – TRUE if the identity pool supports unauthenticated logins.
- **SupportedLoginProviders** (*dict*) –
Optional key:value pairs mapping provider names to provider app IDs.
 - * (*string*) –
 - (*string*) –
- **DeveloperProviderName** (*string*) –
The “domain” by which Cognito will refer to your users.
- **OpenIdConnectProviderARNs** (*list*) –
A list of OpenID Connect provider ARNs.
 - * (*string*) –

CognitoSync

Table of Contents

- CognitoSync
 - Client

Client

class CognitoSync.Client

A low-level client representing Amazon Cognito Sync:

```
import boto3
```

```
client = boto3.client('cognito-sync')
```

These are the available methods:

- `bulk_publish()`
- `can_paginate()`
- `delete_dataset()`
- `describe_dataset()`
- `describe_identity_pool_usage()`
- `describe_identity_usage()`
- `generate_presigned_url()`
- `get_bulk_publish_details()`
- `get_cognito_events()`
- `get_identity_pool_configuration()`
- `get_paginator()`
- `get_waiter()`
- `list_datasets()`
- `list_identity_pool_usage()`
- `list_records()`
- `register_device()`

- `set_cognito_events()`
- `set_identity_pool_configuration()`
- `subscribe_to_dataset()`
- `unsubscribe_from_dataset()`
- `update_records()`

bulk_publish (***kwargs*)

Initiates a bulk publish of all existing datasets for an Identity Pool to the configured stream. Customers are limited to one successful bulk publish per 24 hours. Bulk publish is an asynchronous request, customers can see the status of the request via the `GetBulkPublishDetails` operation.

This API can only be called with developer credentials. You cannot call this API with the temporary user credentials provided by Cognito Identity.

Request Syntax

```
response = client.bulk_publish(
    IdentityPoolId='string'
)
```

Parameters `IdentityPoolId` (*string*) – **[REQUIRED]** A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.

Return type dict

Returns**Response Syntax**

```
{
    'IdentityPoolId': 'string'
}
```

Response Structure

- (*dict*) – The output for the BulkPublish operation.
 - **IdentityPoolId** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters `operation_name` (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

delete_dataset (***kwargs*)

Deletes the specific dataset. The dataset will be deleted permanently, and the action can't be undone. Datasets that this dataset was merged with will no longer report the merge. Any subsequent operation on this dataset will result in a `ResourceNotFoundException`.

This API can be called with temporary user credentials provided by Cognito Identity or with developer credentials.

Request Syntax

```
response = client.delete_dataset(
    IdentityPoolId='string',
    IdentityId='string',
    DatasetName='string'
)
```

Parameters

- **IdentityPoolId** (*string*) – **[REQUIRED]** A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
- **IdentityId** (*string*) – **[REQUIRED]** A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
- **DatasetName** (*string*) – **[REQUIRED]** A string of up to 128 characters. Allowed characters are a-z, A-Z, 0-9, '_' (underscore), '-' (dash), and '.' (dot).

Return type dict

Returns

Response Syntax

```
{
  'Dataset': {
    'IdentityId': 'string',
    'DatasetName': 'string',
    'CreationDate': datetime(2015, 1, 1),
    'LastModifiedDate': datetime(2015, 1, 1),
    'LastModifiedBy': 'string',
    'DataStorage': 123,
    'NumRecords': 123
  }
}
```

Response Structure

- (*dict*) – Response to a successful DeleteDataset request.
 - **Dataset** (*dict*) – A collection of data for an identity pool. An identity pool can have multiple datasets. A dataset is per identity and can be general or associated with a particular entity in an application (like a saved game). Datasets are automatically created if they don't exist. Data is synced by dataset, and a dataset can hold up to 1MB of key-value pairs.
 - * **IdentityId** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
 - * **DatasetName** (*string*) – A string of up to 128 characters. Allowed characters are a-z, A-Z, 0-9, '_' (underscore), '-' (dash), and '.' (dot).
 - * **CreationDate** (*datetime*) – Date on which the dataset was created.
 - * **LastModifiedDate** (*datetime*) – Date when the dataset was last modified.
 - * **LastModifiedBy** (*string*) – The device that made the last change to this dataset.
 - * **DataStorage** (*integer*) – Total size in bytes of the records in this dataset.
 - * **NumRecords** (*integer*) – Number of records in this dataset.

describe_dataset (***kwargs*)

Gets meta data about a dataset by identity and dataset name. With Amazon Cognito Sync, each identity

has access only to its own data. Thus, the credentials used to make this API call need to have access to the identity data.

This API can be called with temporary user credentials provided by Cognito Identity or with developer credentials. You should use Cognito Identity credentials to make this API call.

Request Syntax

```
response = client.describe_dataset(
    IdentityPoolId='string',
    IdentityId='string',
    DatasetName='string'
)
```

Parameters

- **IdentityPoolId** (*string*) – **[REQUIRED]** A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
- **IdentityId** (*string*) – **[REQUIRED]** A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
- **DatasetName** (*string*) – **[REQUIRED]** A string of up to 128 characters. Allowed characters are a-z, A-Z, 0-9, ‘_’ (underscore), ‘-’ (dash), and ‘.’ (dot).

Return type dict

Returns

Response Syntax

```
{
  'Dataset': {
    'IdentityId': 'string',
    'DatasetName': 'string',
    'CreationDate': datetime(2015, 1, 1),
    'LastModifiedDate': datetime(2015, 1, 1),
    'LastModifiedBy': 'string',
    'DataStorage': 123,
    'NumRecords': 123
  }
}
```

Response Structure

- (*dict*) – Response to a successful DescribeDataset request.
 - **Dataset** (*dict*) – Meta data for a collection of data for an identity. An identity can have multiple datasets. A dataset can be general or associated with a particular entity in an application (like a saved game). Datasets are automatically created if they don’t exist. Data is synced by dataset, and a dataset can hold up to 1MB of key-value pairs.
 - * **IdentityId** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
 - * **DatasetName** (*string*) – A string of up to 128 characters. Allowed characters are a-z, A-Z, 0-9, ‘_’ (underscore), ‘-’ (dash), and ‘.’ (dot).
 - * **CreationDate** (*datetime*) – Date on which the dataset was created.
 - * **LastModifiedDate** (*datetime*) – Date when the dataset was last modified.

- * **LastModifiedBy** (*string*) – The device that made the last change to this dataset.
- * **DataStorage** (*integer*) – Total size in bytes of the records in this dataset.
- * **NumRecords** (*integer*) – Number of records in this dataset.

describe_identity_pool_usage (***kwargs*)

Gets usage details (for example, data storage) about a particular identity pool.

This API can only be called with developer credentials. You cannot call this API with the temporary user credentials provided by Cognito Identity.

Request Syntax

```
response = client.describe_identity_pool_usage(
    IdentityPoolId='string'
)
```

Parameters **IdentityPoolId** (*string*) – [REQUIRED] A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.

Return type dict

Returns

Response Syntax

```
{
  'IdentityPoolUsage': {
    'IdentityPoolId': 'string',
    'SyncSessionsCount': 123,
    'DataStorage': 123,
    'LastModifiedDate': datetime(2015, 1, 1)
  }
}
```

Response Structure

- (*dict*) – Response to a successful DescribeIdentityPoolUsage request.
 - **IdentityPoolUsage** (*dict*) – Information about the usage of the identity pool.
 - * **IdentityPoolId** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
 - * **SyncSessionsCount** (*integer*) – Number of sync sessions for the identity pool.
 - * **DataStorage** (*integer*) – Data storage information for the identity pool.
 - * **LastModifiedDate** (*datetime*) – Date on which the identity pool was last modified.

describe_identity_usage (***kwargs*)

Gets usage information for an identity, including number of datasets and data usage.

This API can be called with temporary user credentials provided by Cognito Identity or with developer credentials.

Request Syntax

```
response = client.describe_identity_usage(
    IdentityPoolId='string',
    IdentityId='string'
)
```

Parameters

- **IdentityPoolId** (*string*) – **[REQUIRED]** A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
- **IdentityId** (*string*) – **[REQUIRED]** A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.

Return type dict

Returns

Response Syntax

```
{
  'IdentityUsage': {
    'IdentityId': 'string',
    'IdentityPoolId': 'string',
    'LastModifiedDate': datetime(2015, 1, 1),
    'DatasetCount': 123,
    'DataStorage': 123
  }
}
```

Response Structure

- (*dict*) – The response to a successful DescribeIdentityUsage request.
 - **IdentityUsage** (*dict*) – Usage information for the identity.
 - * **IdentityId** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
 - * **IdentityPoolId** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
 - * **LastModifiedDate** (*datetime*) – Date on which the identity was last modified.
 - * **DatasetCount** (*integer*) – Number of datasets for the identity.
 - * **DataStorage** (*integer*) – Total data storage for this identity.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to *ClientMethod*.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_bulk_publish_details (***kwargs*)

Get the status of the last BulkPublish operation for an identity pool.

This API can only be called with developer credentials. You cannot call this API with the temporary user credentials provided by Cognito Identity.

Request Syntax

```
response = client.get_bulk_publish_details(  
    IdentityPoolId='string'  
)
```

Parameters **IdentityPoolId** (*string*) – **[REQUIRED]** A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.

Return type dict

Returns

Response Syntax

```
{  
    'IdentityPoolId': 'string',  
    'BulkPublishStartTime': datetime(2015, 1, 1),  
    'BulkPublishCompleteTime': datetime(2015, 1, 1),  
    'BulkPublishStatus': 'NOT_STARTED' | 'IN_PROGRESS' | 'FAILED' | 'SUCCEEDED',  
    'FailureMessage': 'string'  
}
```

Response Structure

- (*dict*) – The output for the GetBulkPublishDetails operation.
 - **IdentityPoolId** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
 - **BulkPublishStartTime** (*datetime*) – The date/time at which the last bulk publish was initiated.
 - **BulkPublishCompleteTime** (*datetime*) – If BulkPublishStatus is SUCCEEDED, the time the last bulk publish operation completed.
 - **BulkPublishStatus** (*string*) – Status of the last bulk publish operation, valid values are:
 - NOT_STARTED - No bulk publish has been requested for this identity pool
 - IN_PROGRESS - Data is being published to the configured stream
 - SUCCEEDED - All data for the identity pool has been published to the configured stream
 - FAILED - Some portion of the data has failed to publish, check FailureMessage for the cause.
 - **FailureMessage** (*string*) – If BulkPublishStatus is FAILED this field will contain the error message that caused the bulk publish to fail.

get_cognito_events (***kwargs*)

Gets the events and the corresponding Lambda functions associated with an identity pool.

This API can only be called with developer credentials. You cannot call this API with the temporary user credentials provided by Cognito Identity.

Request Syntax

```
response = client.get_cognito_events(  
    IdentityPoolId='string'  
)
```

Parameters `IdentityPoolId` (*string*) – [REQUIRED]

The Cognito Identity Pool ID for the request

Return type dict

Returns

Response Syntax

```
{
  'Events': {
    'string': 'string'
  }
}
```

Response Structure

- (*dict*) –

The response from the GetCognitoEvents request

– **Events** (*dict*) –

The Cognito Events returned from the GetCognitoEvents request

* (*string*) –

· (*string*) –

get_identity_pool_configuration (***kwargs*)

Gets the configuration settings of an identity pool.

This API can only be called with developer credentials. You cannot call this API with the temporary user credentials provided by Cognito Identity.

Request Syntax

```
response = client.get_identity_pool_configuration(
    IdentityPoolId='string'
)
```

Parameters `IdentityPoolId` (*string*) – [REQUIRED]

A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. This is the ID of the pool for which to return a configuration.

Return type dict

Returns

Response Syntax

```
{
  'IdentityPoolId': 'string',
  'PushSync': {
    'ApplicationArns': [
      'string',
    ],
    'RoleArn': 'string'
  },
  'CognitoStreams': {
    'StreamName': 'string',
    'RoleArn': 'string',
    'StreamingStatus': 'ENABLED'|'DISABLED'
  }
}
```

Response Structure

- (*dict*) –

The output for the `GetIdentityPoolConfiguration` operation.

- **IdentityPoolId** (*string*) –

A name-spaced GUID (for example, `us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE`) created by Amazon Cognito.

- **PushSync** (*dict*) –

Options to apply to this identity pool for push synchronization.

- * **ApplicationArns** (*list*) –

List of SNS platform application ARNs that could be used by clients.

- (*string*) –

- * **RoleArn** (*string*) –

A role configured to allow Cognito to call SNS on behalf of the developer.

- **CognitoStreams** (*dict*) – Options to apply to this identity pool for Amazon Cognito streams.

- * **StreamName** (*string*) – The name of the Cognito stream to receive updates. This stream must be in the developers account and in the same region as the identity pool.

- * **RoleArn** (*string*) – The ARN of the role Amazon Cognito can assume in order to publish to the stream. This role must grant access to Amazon Cognito (`cognito-sync`) to invoke `PutRecord` on your Cognito stream.

- * **StreamingStatus** (*string*) – Status of the Cognito streams. Valid values are:

ENABLED - Streaming of updates to identity pool is enabled.

DISABLED - Streaming of updates to identity pool is disabled. Bulk publish will also fail if `StreamingStatus` is `DISABLED`.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_waiter (*waiter_name*)**list_datasets** (***kwargs*)

Lists datasets for an identity. With Amazon Cognito Sync, each identity has access only to its own data. Thus, the credentials used to make this API call need to have access to the identity data.

`ListDatasets` can be called with temporary user credentials provided by Cognito Identity or with developer credentials. You should use the Cognito Identity credentials to make this API call.

Request Syntax


```
response = client.list_datasets(
    IdentityPoolId='string',
    IdentityId='string',
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **IdentityPoolId** (*string*) – **[REQUIRED]** A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
- **IdentityId** (*string*) – **[REQUIRED]** A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
- **NextToken** (*string*) – A pagination token for obtaining the next page of results.
- **MaxResults** (*integer*) – The maximum number of results to be returned.

Return type dict

Returns

Response Syntax

```
{
  'Datasets': [
    {
      'IdentityId': 'string',
      'DatasetName': 'string',
      'CreationDate': datetime(2015, 1, 1),
      'LastModifiedDate': datetime(2015, 1, 1),
      'LastModifiedBy': 'string',
      'DataStorage': 123,
      'NumRecords': 123
    },
  ],
  'Count': 123,
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) – Returned for a successful ListDatasets request.
 - **Datasets** (*list*) – A set of datasets.
 - * (*dict*) – A collection of data for an identity pool. An identity pool can have multiple datasets. A dataset is per identity and can be general or associated with a particular entity in an application (like a saved game). Datasets are automatically created if they don't exist. Data is synced by dataset, and a dataset can hold up to 1MB of key-value pairs.
 - **IdentityId** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
 - **DatasetName** (*string*) – A string of up to 128 characters. Allowed characters are a-z, A-Z, 0-9, '_' (underscore), '-' (dash), and '.' (dot).
 - **CreationDate** (*datetime*) – Date on which the dataset was created.

- **LastModifiedDate** (*datetime*) – Date when the dataset was last modified.
- **LastModifiedBy** (*string*) – The device that made the last change to this dataset.
- **DataStorage** (*integer*) – Total size in bytes of the records in this dataset.
- **NumRecords** (*integer*) – Number of records in this dataset.
- **Count** (*integer*) – Number of datasets returned.
- **NextToken** (*string*) – A pagination token for obtaining the next page of results.

list_identity_pool_usage (***kwargs*)

Gets a list of identity pools registered with Cognito.

ListIdentityPoolUsage can only be called with developer credentials. You cannot make this API call with the temporary user credentials provided by Cognito Identity.

Request Syntax

```
response = client.list_identity_pool_usage(
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **NextToken** (*string*) – A pagination token for obtaining the next page of results.
- **MaxResults** (*integer*) – The maximum number of results to be returned.

Return type dict

Returns

Response Syntax

```
{
  'IdentityPoolUsages': [
    {
      'IdentityPoolId': 'string',
      'SyncSessionsCount': 123,
      'DataStorage': 123,
      'LastModifiedDate': datetime(2015, 1, 1)
    },
  ],
  'MaxResults': 123,
  'Count': 123,
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) – Returned for a successful ListIdentityPoolUsage request.
 - **IdentityPoolUsages** (*list*) – Usage information for the identity pools.
 - * (*dict*) – Usage information for the identity pool.
 - **IdentityPoolId** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
 - **SyncSessionsCount** (*integer*) – Number of sync sessions for the identity pool.
 - **DataStorage** (*integer*) – Data storage information for the identity pool.

- **LastModifiedDate** (*datetime*) – Date on which the identity pool was last modified.
- **MaxResults** (*integer*) – The maximum number of results to be returned.
- **Count** (*integer*) – Total number of identities for the identity pool.
- **NextToken** (*string*) – A pagination token for obtaining the next page of results.

`list_records` (**kwargs)

Gets paginated records, optionally changed after a particular sync count for a dataset and identity. With Amazon Cognito Sync, each identity has access only to its own data. Thus, the credentials used to make this API call need to have access to the identity data.

ListRecords can be called with temporary user credentials provided by Cognito Identity or with developer credentials. You should use Cognito Identity credentials to make this API call.

Request Syntax

```
response = client.list_records(
    IdentityPoolId='string',
    IdentityId='string',
    DatasetName='string',
    LastSyncCount=123,
    NextToken='string',
    MaxResults=123,
    SyncSessionToken='string'
)
```

Parameters

- **IdentityPoolId** (*string*) – **[REQUIRED]** A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
- **IdentityId** (*string*) – **[REQUIRED]** A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
- **DatasetName** (*string*) – **[REQUIRED]** A string of up to 128 characters. Allowed characters are a-z, A-Z, 0-9, '_' (underscore), '-' (dash), and '.' (dot).
- **LastSyncCount** (*integer*) – The last server sync count for this record.
- **NextToken** (*string*) – A pagination token for obtaining the next page of results.
- **MaxResults** (*integer*) – The maximum number of results to be returned.
- **SyncSessionToken** (*string*) – A token containing a session ID, identity ID, and expiration.

Return type dict

Returns

Response Syntax

```
{
  'Records': [
    {
      'Key': 'string',
      'Value': 'string',
      'SyncCount': 123,
      'LastModifiedDate': datetime(2015, 1, 1),
      'LastModifiedBy': 'string',
      'DeviceLastModifiedDate': datetime(2015, 1, 1)
    },
  ],
}
```

```
'NextToken': 'string',
'Count': 123,
'DatasetSyncCount': 123,
'LastModifiedBy': 'string',
'MergedDatasetNames': [
    'string',
],
'DatasetExists': True|False,
'DatasetDeletedAfterRequestedSyncCount': True|False,
'SyncSessionToken': 'string'
}
```

Response Structure

- (*dict*) – Returned for a successful ListRecordsRequest.
 - **Records** (*list*) – A list of all records.
 - * (*dict*) – The basic data structure of a dataset.
 - **Key** (*string*) – The key for the record.
 - **Value** (*string*) – The value for the record.
 - **SyncCount** (*integer*) – The server sync count for this record.
 - **LastModifiedDate** (*datetime*) – The date on which the record was last modified.
 - **LastModifiedBy** (*string*) – The user/device that made the last change to this record.
 - **DeviceLastModifiedDate** (*datetime*) – The last modified date of the client device.
 - **NextToken** (*string*) – A pagination token for obtaining the next page of results.
 - **Count** (*integer*) – Total number of records.
 - **DatasetSyncCount** (*integer*) – Server sync count for this dataset.
 - **LastModifiedBy** (*string*) – The user/device that made the last change to this record.
 - **MergedDatasetNames** (*list*) – Names of merged datasets.
 - * (*string*) –
 - **DatasetExists** (*boolean*) – Indicates whether the dataset exists.
 - **DatasetDeletedAfterRequestedSyncCount** (*boolean*) – A boolean value specifying whether to delete the dataset locally.
 - **SyncSessionToken** (*string*) – A token containing a session ID, identity ID, and expiration.

register_device (**kwargs)

Registers a device to receive push sync notifications.

This API can only be called with temporary credentials provided by Cognito Identity. You cannot call this API with developer credentials.

Request Syntax

```
response = client.register_device(
    IdentityPoolId='string',
    IdentityId='string',
    Platform='APNS'|'APNS_SANDBOX'|'GCM'|'ADM',
    Token='string'
)
```

Parameters

- **IdentityPoolId** (*string*) – [REQUIRED]

A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. Here, the ID of the pool that the identity belongs to.

- **IdentityId** (*string*) – [REQUIRED]

The unique ID for this identity.

- **Platform** (*string*) – [REQUIRED]

The SNS platform type (e.g. GCM, SDM, APNS, APNS_SANDBOX).

- **Token** (*string*) – [REQUIRED]

The push token.

Return type dict

Returns

Response Syntax

```
{
    'DeviceId': 'string'
}
```

Response Structure

- (*dict*) –

Response to a RegisterDevice request.

- **DeviceId** (*string*) –

The unique ID generated for this device by Cognito.

set_cognito_events (***kwargs*)

Sets the AWS Lambda function for a given event type for an identity pool. This request only updates the key/value pair specified. Other key/values pairs are not updated. To remove a key value pair, pass a empty value for the particular key.

This API can only be called with developer credentials. You cannot call this API with the temporary user credentials provided by Cognito Identity.

Request Syntax

```
response = client.set_cognito_events(
    IdentityPoolId='string',
    Events={
        'string': 'string'
    }
)
```

Parameters

- **IdentityPoolId** (*string*) – [REQUIRED]

The Cognito Identity Pool to use when configuring Cognito Events

- **Events** (*dict*) – [REQUIRED]

The events to configure

- (*string*) –
- * (*string*) –

Returns None

set_identity_pool_configuration (***kwargs*)

Sets the necessary configuration for push sync.

This API can only be called with developer credentials. You cannot call this API with the temporary user credentials provided by Cognito Identity.

Request Syntax

```
response = client.set_identity_pool_configuration(
    IdentityPoolId='string',
    PushSync={
        'ApplicationArns': [
            'string',
        ],
        'RoleArn': 'string'
    },
    CognitoStreams={
        'StreamName': 'string',
        'RoleArn': 'string',
        'StreamingStatus': 'ENABLED'|'DISABLED'
    }
)
```

Parameters

- **IdentityPoolId** (*string*) – [REQUIRED]

A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. This is the ID of the pool to modify.

- **PushSync** (*dict*) – Options to apply to this identity pool for push synchronization.

- **ApplicationArns** (*list*) –

List of SNS platform application ARNs that could be used by clients.

* (*string*) –

- **RoleArn** (*string*) –

A role configured to allow Cognito to call SNS on behalf of the developer.

- **CognitoStreams** (*dict*) – Options to apply to this identity pool for Amazon Cognito streams.

- **StreamName** (*string*) – The name of the Cognito stream to receive updates. This stream must be in the developers account and in the same region as the identity pool.

- **RoleArn** (*string*) – The ARN of the role Amazon Cognito can assume in order to publish to the stream. This role must grant access to Amazon Cognito (cognito-sync) to invoke PutRecord on your Cognito stream.

- **StreamingStatus** (*string*) – Status of the Cognito streams. Valid values are:

ENABLED - Streaming of updates to identity pool is enabled.

DISABLED - Streaming of updates to identity pool is disabled. Bulk publish will also fail if StreamingStatus is DISABLED.

Return type dict

Returns

Response Syntax

```
{
    'IdentityPoolId': 'string',
    'PushSync': {
        'ApplicationArns': [
```

```

        'string',
    ],
    'RoleArn': 'string'
},
'CognitoStreams': {
    'StreamName': 'string',
    'RoleArn': 'string',
    'StreamingStatus': 'ENABLED'|'DISABLED'
}
}

```

Response Structure

- (*dict*) –

The output for the SetIdentityPoolConfiguration operation

- **IdentityPoolId** (*string*) –

A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito.

- **PushSync** (*dict*) –

Options to apply to this identity pool for push synchronization.

- * **ApplicationArns** (*list*) –

List of SNS platform application ARNs that could be used by clients.

· (*string*) –

- * **RoleArn** (*string*) –

A role configured to allow Cognito to call SNS on behalf of the developer.

- **CognitoStreams** (*dict*) – Options to apply to this identity pool for Amazon Cognito streams.

- * **StreamName** (*string*) – The name of the Cognito stream to receive updates. This stream must be in the developers account and in the same region as the identity pool.

- * **RoleArn** (*string*) – The ARN of the role Amazon Cognito can assume in order to publish to the stream. This role must grant access to Amazon Cognito (cognito-sync) to invoke PutRecord on your Cognito stream.

- * **StreamingStatus** (*string*) – Status of the Cognito streams. Valid values are:

ENABLED - Streaming of updates to identity pool is enabled.

DISABLED - Streaming of updates to identity pool is disabled. Bulk publish will also fail if StreamingStatus is DISABLED.

subscribe_to_dataset (***kwargs*)

Subscribes to receive notifications when a dataset is modified by another device.

This API can only be called with temporary credentials provided by Cognito Identity. You cannot call this API with developer credentials.

Request Syntax

```

response = client.subscribe_to_dataset(
    IdentityPoolId='string',
    IdentityId='string',

```

```
DatasetName='string',  
DeviceId='string'  
)
```

Parameters

- **IdentityPoolId** (*string*) – [REQUIRED]

A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. The ID of the pool to which the identity belongs.

- **IdentityId** (*string*) – [REQUIRED]

Unique ID for this identity.

- **DatasetName** (*string*) – [REQUIRED]

The name of the dataset to subscribe to.

- **DeviceId** (*string*) – [REQUIRED]

The unique ID generated for this device by Cognito.

Return type dict

Returns**Response Syntax**

```
{}
```

Response Structure

- (*dict*) –

Response to a SubscribeToDataset request.

unsubscribe_from_dataset (***kwargs*)

Unsubscribes from receiving notifications when a dataset is modified by another device.

This API can only be called with temporary credentials provided by Cognito Identity. You cannot call this API with developer credentials.

Request Syntax

```
response = client.unsubscribe_from_dataset(  
    IdentityPoolId='string',  
    IdentityId='string',  
    DatasetName='string',  
    DeviceId='string'  
)
```

Parameters

- **IdentityPoolId** (*string*) – [REQUIRED]

A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. The ID of the pool to which this identity belongs.

- **IdentityId** (*string*) – [REQUIRED]

Unique ID for this identity.

- **DatasetName** (*string*) – [REQUIRED]

The name of the dataset from which to unsubscribe.

- **DeviceId** (*string*) – [REQUIRED]

The unique ID generated for this device by Cognito.

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –

Response to an UnsubscribeFromDataset request.

update_records (**kwargs)

Posts updates to records and adds and deletes records for a dataset and user.

The sync count in the record patch is your last known sync count for that record. The server will reject an UpdateRecords request with a ResourceConflictException if you try to patch a record with a new value but a stale sync count.

For example, if the sync count on the server is 5 for a key called highScore and you try and submit a new highScore with sync count of 4, the request will be rejected. To obtain the current sync count for a record, call ListRecords. On a successful update of the record, the response returns the new sync count for that record. You should present that sync count the next time you try to update that same record. When the record does not exist, specify the sync count as 0.

This API can be called with temporary user credentials provided by Cognito Identity or with developer credentials.

Request Syntax

```
response = client.update_records(
    IdentityPoolId='string',
    IdentityId='string',
    DatasetName='string',
    DeviceId='string',
    RecordPatches=[
        {
            'Op': 'replace'|'remove',
            'Key': 'string',
            'Value': 'string',
            'SyncCount': 123,
            'DeviceLastModifiedDate': datetime(2015, 1, 1)
        },
    ],
    SyncSessionToken='string',
    ClientContext='string'
)
```

Parameters

- **IdentityPoolId** (*string*) – **[REQUIRED]** A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
- **IdentityId** (*string*) – **[REQUIRED]** A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
- **DatasetName** (*string*) – **[REQUIRED]** A string of up to 128 characters. Allowed characters are a-z, A-Z, 0-9, '_' (underscore), '-' (dash), and '.' (dot).
- **DeviceId** (*string*) – The unique ID generated for this device by Cognito.
- **RecordPatches** (*list*) – A list of patch operations.

- (*dict*) – An update operation for a record.
 - * **Op** (*string*) – **[REQUIRED]** An operation, either replace or remove.
 - * **Key** (*string*) – **[REQUIRED]** The key associated with the record patch.
 - * **Value** (*string*) – The value associated with the record patch.
 - * **SyncCount** (*integer*) – **[REQUIRED]** Last known server sync count for this record. Set to 0 if unknown.
 - * **DeviceLastModifiedDate** (*datetime*) – The last modified date of the client device.
- **SyncSessionToken** (*string*) – **[REQUIRED]** The SyncSessionToken returned by a previous call to ListRecords for this dataset and identity.
- **ClientContext** (*string*) – Intended to supply a device ID that will populate the lastModifiedBy field referenced in other methods. The ClientContext field is not yet implemented.

Return type dict

Returns

Response Syntax

```
{
  'Records': [
    {
      'Key': 'string',
      'Value': 'string',
      'SyncCount': 123,
      'LastModifiedDate': datetime(2015, 1, 1),
      'LastModifiedBy': 'string',
      'DeviceLastModifiedDate': datetime(2015, 1, 1)
    },
  ]
}
```

Response Structure

- (*dict*) – Returned for a successful UpdateRecordsRequest.
 - **Records** (*list*) – A list of records that have been updated.
 - * (*dict*) – The basic data structure of a dataset.
 - **Key** (*string*) – The key for the record.
 - **Value** (*string*) – The value for the record.
 - **SyncCount** (*integer*) – The server sync count for this record.
 - **LastModifiedDate** (*datetime*) – The date on which the record was last modified.
 - **LastModifiedBy** (*string*) – The user/device that made the last change to this record.
 - **DeviceLastModifiedDate** (*datetime*) – The last modified date of the client device.

ConfigService

Table of Contents

- ConfigService
 - Client

Client

class ConfigService.**Client**

A low-level client representing AWS Config (Config Service):

```
import boto3

client = boto3.client('config')
```

These are the available methods:

- `can_paginate()`
- `delete_delivery_channel()`
- `deliver_config_snapshot()`
- `describe_configuration_recorder_status()`
- `describe_configuration_recorders()`
- `describe_delivery_channel_status()`
- `describe_delivery_channels()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_resource_config_history()`
- `get_waiter()`
- `list_discovered_resources()`
- `put_configuration_recorder()`
- `put_delivery_channel()`
- `start_configuration_recorder()`
- `stop_configuration_recorder()`

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

delete_delivery_channel (***kwargs*)

Deletes the specified delivery channel.

The delivery channel cannot be deleted if it is the only delivery channel and the configuration recorder is still running. To delete the delivery channel, stop the running configuration recorder using the `StopConfigurationRecorder` action.

Request Syntax

```
response = client.delete_delivery_channel(
    DeliveryChannelName='string'
)
```

Parameters *DeliveryChannelName* (*string*) – [REQUIRED]

The name of the delivery channel to delete.

Returns None

deliver_config_snapshot (***kwargs*)

Schedules delivery of a configuration snapshot to the Amazon S3 bucket in the specified delivery channel. After the delivery has started, AWS Config sends following notifications using an Amazon SNS topic that you have specified.

- Notification of starting the delivery.
- Notification of delivery completed, if the delivery was successfully completed.
- Notification of delivery failure, if the delivery failed to complete.

Request Syntax

```
response = client.deliver_config_snapshot(  
    deliveryChannelName='string'  
)
```

Parameters `deliveryChannelName` (*string*) – **[REQUIRED]**

The name of the delivery channel through which the snapshot is delivered.

Return type dict

Returns**Response Syntax**

```
{  
    'configSnapshotId': 'string'  
}
```

Response Structure

- (*dict*) –

The output for the DeliverConfigSnapshot action in JSON format.

– **configSnapshotId** (*string*) –

The ID of the snapshot that is being created.

describe_configuration_recorder_status (***kwargs*)

Returns the current status of the specified configuration recorder. If a configuration recorder is not specified, this action returns the status of all configuration recorder associated with the account.

Note: Currently, you can specify only one configuration recorder per account.

Request Syntax

```
response = client.describe_configuration_recorder_status(  
    ConfigurationRecorderNames=[  
        'string',  
    ]  
)
```

Parameters **ConfigurationRecorderNames** (*list*) – The name(s) of the configuration recorder. If the name is not specified, the action returns the current status of all the configuration recorders associated with the account.

- (*string*) –

Return type dict

Returns**Response Syntax**

```
{  
    'ConfigurationRecordersStatus': [  
        {  
            'name': 'string',  
            'lastStartTime': datetime(2015, 1, 1),  
            'lastStopTime': datetime(2015, 1, 1),  
        }  
    ]  
}
```

```

        'recording': True|False,
        'lastStatus': 'Pending'|'Success'|'Failure',
        'lastErrorCode': 'string',
        'lastErrorMessage': 'string',
        'lastStatusChangeTime': datetime(2015, 1, 1)
    },
]
}

```

Response Structure

- (*dict*) –

The output for the DescribeConfigurationRecorderStatus action in JSON format.

– ConfigurationRecordersStatus (*list*) –

A list that contains status of the specified recorders.

- * (*dict*) –

The current status of the configuration recorder.

- **name** (*string*) –

The name of the configuration recorder.

- **lastStartTime** (*datetime*) –

The time the recorder was last started.

- **lastStopTime** (*datetime*) –

The time the recorder was last stopped.

- **recording** (*boolean*) –

Specifies whether the recorder is currently recording or not.

- **lastStatus** (*string*) –

The last (previous) status of the recorder.

- **lastErrorCode** (*string*) –

The error code indicating that the recording failed.

- **lastErrorMessage** (*string*) –

The message indicating that the recording failed due to an error.

- **lastStatusChangeTime** (*datetime*) –

The time when the status was last changed.

describe_configuration_recorders (***kwargs*)

Returns the name of one or more specified configuration recorders. If the recorder name is not specified, this action returns the names of all the configuration recorders associated with the account.

Note: Currently, you can specify only one configuration recorder per account.

Request Syntax

```

response = client.describe_configuration_recorders(
    ConfigurationRecorderNames=[
        'string',
    ]
)

```

Parameters **ConfigurationRecorderNames** (*list*) – A list of configuration recorder names.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'ConfigurationRecorders': [
    {
      'name': 'string',
      'roleARN': 'string',
      'recordingGroup': {
        'allSupported': True|False,
        'resourceTypes': [
          'AWS::EC2::CustomerGateway' | 'AWS::EC2::EIP' | 'AWS::EC2::Instance'
        ]
      }
    },
  ]
}
```

Response Structure

- (*dict*) –

The output for the DescribeConfigurationRecorders action.

- **ConfigurationRecorders** (*list*) –

A list that contains the descriptions of the specified configuration recorders.

- * (*dict*) –

An object that represents the recording of configuration changes of an AWS resource.

- **name** (*string*) –

The name of the recorder. By default, AWS Config automatically assigns the name “default” when creating the configuration recorder. You cannot change the assigned name.

- **roleARN** (*string*) –

Amazon Resource Name (ARN) of the IAM role used to describe the AWS resources associated with the account.

- **recordingGroup** (*dict*) –

The recording group specifies either to record configurations for all supported resources or to provide a list of resource types to record. The list of resource types must be a subset of supported resource types.

- **allSupported** (*boolean*) –

Records all supported resource types in the recording group. For a list of supported resource types, see [Supported resource types](#). If you specify **allSupported**, you cannot enumerate a list of **resourceTypes**.

- **resourceTypes** (*list*) –

A comma-separated list of strings representing valid AWS resource types (for example, `AWS::EC2::Instance` or `AWS::CloudTrail::Trail`). **resourceTypes** is only valid if you have chosen not to select **allSupported**. For a list of valid **resourceTypes** values, see the **resourceType**

Value column in the following topic: [Supported AWS Resource Types](#).

• (string) –

describe_delivery_channel_status (**kwargs)

Returns the current status of the specified delivery channel. If a delivery channel is not specified, this action returns the current status of all delivery channels associated with the account.

Note: Currently, you can specify only one delivery channel per account.

Request Syntax

```
response = client.describe_delivery_channel_status(
    DeliveryChannelNames=[
        'string',
    ]
)
```

Parameters **DeliveryChannelNames** (list) – A list of delivery channel names.

• (string) –

Return type dict

Returns

Response Syntax

```
{
    'DeliveryChannelsStatus': [
        {
            'name': 'string',
            'configSnapshotDeliveryInfo': {
                'lastStatus': 'Success'|'Failure'|'Not_Applicable',
                'lastErrorCode': 'string',
                'lastErrorMessage': 'string',
                'lastAttemptTime': datetime(2015, 1, 1),
                'lastSuccessfulTime': datetime(2015, 1, 1)
            },
            'configHistoryDeliveryInfo': {
                'lastStatus': 'Success'|'Failure'|'Not_Applicable',
                'lastErrorCode': 'string',
                'lastErrorMessage': 'string',
                'lastAttemptTime': datetime(2015, 1, 1),
                'lastSuccessfulTime': datetime(2015, 1, 1)
            },
            'configStreamDeliveryInfo': {
                'lastStatus': 'Success'|'Failure'|'Not_Applicable',
                'lastErrorCode': 'string',
                'lastErrorMessage': 'string',
                'lastStatusChangeTime': datetime(2015, 1, 1)
            }
        },
    ]
}
```

Response Structure

• (dict) –

The output for the DescribeDeliveryChannelStatus action.

– **DeliveryChannelsStatus** (*list*) –

A list that contains the status of a specified delivery channel.

* (*dict*) –

The status of a specified delivery channel.

Valid values: `Success` | `Failure`

- **name** (*string*) –

The name of the delivery channel.

- **configSnapshotDeliveryInfo** (*dict*) –

A list containing the status of the delivery of the snapshot to the specified Amazon S3 bucket.

- **lastStatus** (*string*) –

Status of the last attempted delivery.

- **lastErrorCode** (*string*) –

The error code from the last attempted delivery.

- **lastErrorMessage** (*string*) –

The error message from the last attempted delivery.

- **lastAttemptTime** (*datetime*) –

The time of the last attempted delivery.

- **lastSuccessfulTime** (*datetime*) –

The time of the last successful delivery.

- **configHistoryDeliveryInfo** (*dict*) –

A list that contains the status of the delivery of the configuration history to the specified Amazon S3 bucket.

- **lastStatus** (*string*) –

Status of the last attempted delivery.

- **lastErrorCode** (*string*) –

The error code from the last attempted delivery.

- **lastErrorMessage** (*string*) –

The error message from the last attempted delivery.

- **lastAttemptTime** (*datetime*) –

The time of the last attempted delivery.

- **lastSuccessfulTime** (*datetime*) –

The time of the last successful delivery.

- **configStreamDeliveryInfo** (*dict*) –

A list containing the status of the delivery of the configuration stream notification to the specified Amazon SNS topic.

- **lastStatus** (*string*) –

Status of the last attempted delivery.

Note Providing an SNS topic on a [DeliveryChannel](#) for AWS Config is optional. If the SNS delivery is turned off, the last status will be **Not_Applicable** .

- **lastErrorCode** (*string*) –

The error code from the last attempted delivery.

- **lastErrorMessage** (*string*) –
The error message from the last attempted delivery.
- **lastStatusChangeTime** (*datetime*) –
The time from the last status change.

describe_delivery_channels (***kwargs*)

Returns details about the specified delivery channel. If a delivery channel is not specified, this action returns the details of all delivery channels associated with the account.

Note: Currently, you can specify only one delivery channel per account.

Request Syntax

```
response = client.describe_delivery_channels(
    DeliveryChannelNames=[
        'string',
    ]
)
```

Parameters **DeliveryChannelNames** (*list*) – A list of delivery channel names.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'DeliveryChannels': [
        {
            'name': 'string',
            's3BucketName': 'string',
            's3KeyPrefix': 'string',
            'snsTopicARN': 'string'
        },
    ]
}
```

Response Structure

- (*dict*) –

The output for the DescribeDeliveryChannels action.

– **DeliveryChannels** (*list*) –

A list that contains the descriptions of the specified delivery channel.

- * (*dict*) –

A logical container used for storing the configuration changes of an AWS resource.

- **name** (*string*) –

The name of the delivery channel. By default, AWS Config automatically assigns the name “default” when creating the delivery channel. You cannot change the assigned name.

- **s3BucketName** (*string*) –

The name of the Amazon S3 bucket used to store configuration history for the delivery channel.

- **s3KeyPrefix** (*string*) –
The prefix for the specified Amazon S3 bucket.
- **snsTopicARN** (*string*) –
The Amazon Resource Name (ARN) of the SNS topic that AWS Config delivers notifications to.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to *ClientMethod*.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_resource_config_history (***kwargs*)

Returns a list of configuration items for the specified resource. The list contains details about each state of the resource during the specified time interval.

The response is paginated, and by default, AWS Config returns a limit of 10 configuration items per page. You can customize this number with the `limit` parameter. The response includes a `nextToken` string, and to get the next page of results, run the request again and enter this string for the `nextToken` parameter.

Note: Each call to the API is limited to span a duration of seven days. It is likely that the number of records returned is smaller than the specified `limit`. In such cases, you can make another call, using the `nextToken`.

Request Syntax

```
response = client.get_resource_config_history(
    resourceType='AWS::EC2::CustomerGateway'|'AWS::EC2::EIP'|'AWS::EC2::Instance'|'AWS::EC2::
    resourceId='string',
    laterTime=datetime(2015, 1, 1),
    earlierTime=datetime(2015, 1, 1),
    chronologicalOrder='Reverse'|'Forward',
    limit=123,
    nextToken='string'
)
```

Parameters

- **resourceType** (*string*) – [REQUIRED]
The resource type.
- **resourceId** (*string*) – [REQUIRED]
The ID of the resource (for example., sg-xxxxxx).
- **laterTime** (*datetime*) – The time stamp that indicates a later time. If not specified, current time is taken.
- **earlierTime** (*datetime*) – The time stamp that indicates an earlier time. If not specified, the action returns paginated results that contain configuration items that start from when the first configuration item was recorded.
- **chronologicalOrder** (*string*) – The chronological order for configuration items listed. By default the results are listed in reverse chronological order.
- **limit** (*integer*) – The maximum number of configuration items returned on each page. The default is 10. You cannot specify a limit greater than 100. If you specify 0, AWS Config uses the default.
- **nextToken** (*string*) – The nextToken string returned on a previous page that you use to get the next page of results in a paginated response.

Return type dict

Returns

Response Syntax

```
{
  'configurationItems': [
    {
      'version': 'string',
      'accountId': 'string',
      'configurationItemCaptureTime': datetime(2015, 1, 1),
      'configurationItemStatus': 'Ok'|'Failed'|'Discovered'|'Deleted',
      'configurationStateId': 'string',
      'configurationItemMD5Hash': 'string',
      'arn': 'string',
      'resourceType': 'AWS::EC2::CustomerGateway'|'AWS::EC2::EIP'|'AWS::EC2:',
      'resourceId': 'string',
      'resourceName': 'string',
      'awsRegion': 'string',
      'availabilityZone': 'string',
      'resourceCreationTime': datetime(2015, 1, 1),
      'tags': {
        'string': 'string'
      },
      'relatedEvents': [
        'string',
      ],
      'relationships': [
        {
          'resourceType': 'AWS::EC2::CustomerGateway'|'AWS::EC2::EIP'|'A',
          'resourceId': 'string',
          'resourceName': 'string',
          'relationshipName': 'string'
        },
      ],
      'configuration': 'string'
    },
  ],
  'nextToken': 'string'
}
```

Response Structure

- (*dict*) –

The output for the `GetResourceConfigHistory` action.

- **configurationItems** (*list*) –

A list that contains the configuration history of one or more resources.

- * (*dict*) –

A list that contains detailed configurations of a specified resource.

Note: Currently, the list does not contain information about non-AWS components (for example, applications on your Amazon EC2 instances).

- **version** (*string*) –

The version number of the resource configuration.

- **accountId** (*string*) –

The 12 digit AWS account ID associated with the resource.

- **configurationItemCaptureTime** (*datetime*) –

The time when the configuration recording was initiated.

- **configurationItemStatus** (*string*) –

The configuration item status.

- **configurationStateId** (*string*) –

An identifier that indicates the ordering of the configuration items of a resource.

- **configurationItemMD5Hash** (*string*) –

Unique MD5 hash that represents the configuration item's state.

You can use MD5 hash to compare the states of two or more configuration items that are associated with the same resource.

- **arn** (*string*) –

The Amazon Resource Name (ARN) of the resource.

- **resourceType** (*string*) –

The type of AWS resource.

- **resourceId** (*string*) –

The ID of the resource (for example., `sg-xxxxxxx`).

- **resourceName** (*string*) –

The custom name of the resource, if available.

- **awsRegion** (*string*) –

The region where the resource resides.

- **availabilityZone** (*string*) –

The Availability Zone associated with the resource.

- **resourceCreationTime** (*datetime*) –

The time stamp when the resource was created.

- **tags** (*dict*) –

A mapping of key value tags associated with the resource.

- *(string)* –
- *(string)* –
- **relatedEvents** (*list*) –

A list of CloudTrail event IDs.

A populated field indicates that the current configuration was initiated by the events recorded in the CloudTrail log. For more information about CloudTrail, see [What is AWS CloudTrail?](#) .

An empty field indicates that the current configuration was not initiated by any event.

- *(string)* –
- **relationships** (*list*) –

A list of related AWS resources.

- *(dict)* –

The relationship of the related resource to the main resource.

- **resourceType** (*string*) –

The resource type of the related resource.

- **resourceId** (*string*) –

The ID of the related resource (for example, `sg-xxxxxx`).

- **resourceName** (*string*) –

The custom name of the related resource, if available.

- **relationshipName** (*string*) –

The type of relationship with the related resource.

- **configuration** (*string*) –

The description of the resource configuration.

- **nextToken** (*string*) –

The string that you use in a subsequent request to get the next page of results in a paginated response.

get_waiter (*waiter_name*)

list_discovered_resources (***kwargs*)

Accepts a resource type and returns a list of resource identifiers for the resources of that type. A resource identifier includes the resource type, ID, and (if available) the custom resource name. The results consist of resources that AWS Config has discovered, including those that AWS Config is not currently recording. You can narrow the results to include only resources that have specific resource IDs or a resource name.

Note: You can specify either resource IDs or a resource name but not both in the same request.

The response is paginated, and by default AWS Config lists 100 resource identifiers on each page. You can customize this number with the `limit` parameter. The response includes a `nextToken` string, and to get the next page of results, run the request again and enter this string for the `nextToken` parameter.

Request Syntax

```
response = client.list_discovered_resources(
    resourceType='AWS::EC2::CustomerGateway' | 'AWS::EC2::EIP' | 'AWS::EC2::Instance' | 'AWS::EC2::',
    resourceIds=[
        'string',
    ],
)
```

```
resourceName='string',
limit=123,
includeDeletedResources=True|False,
nextToken='string'
)
```

Parameters

- **resourceType** (*string*) – [REQUIRED]
The type of resources that you want AWS Config to list in the response.
- **resourceIds** (*list*) – The IDs of only those resources that you want AWS Config to list in the response. If you do not specify this parameter, AWS Config lists all resources of the specified type that it has discovered.
 - (*string*) –
- **resourceName** (*string*) – The custom name of only those resources that you want AWS Config to list in the response. If you do not specify this parameter, AWS Config lists all resources of the specified type that it has discovered.
- **limit** (*integer*) – The maximum number of resource identifiers returned on each page. The default is 100. You cannot specify a limit greater than 100. If you specify 0, AWS Config uses the default.
- **includeDeletedResources** (*boolean*) – Specifies whether AWS Config includes deleted resources in the results. By default, deleted resources are not included.
- **nextToken** (*string*) – The nextToken string returned on a previous page that you use to get the next page of results in a paginated response.

Return type dict

Returns

Response Syntax

```
{
  'resourceIdentifiers': [
    {
      'resourceType': 'AWS::EC2::CustomerGateway'|'AWS::EC2::EIP'|'AWS::EC2::',
      'resourceId': 'string',
      'resourceName': 'string',
      'resourceDeletionTime': datetime(2015, 1, 1)
    },
  ],
  'nextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **resourceIdentifiers** (*list*) –
The details that identify a resource that is discovered by AWS Config, including the resource type, ID, and (if available) the custom resource name.
 - * (*dict*) –
The details that identify a resource that is discovered by AWS Config, including the resource type, ID, and (if available) the custom resource name.
 - **resourceType** (*string*) –
The type of resource.
 - **resourceId** (*string*) –

- The ID of the resource (for example., sg-xxxxxx).
- **resourceName** (*string*) –
 - The custom name of the resource (if available).
- **resourceDeletionTime** (*datetime*) –
 - The time that the resource was deleted.
- **nextToken** (*string*) –
 - The string that you use in a subsequent request to get the next page of results in a paginated response.

put_configuration_recorder (***kwargs*)

Creates a new configuration recorder to record the selected resource configurations.

You can use this action to change the role `roleARN` and/or the `recordingGroup` of an existing recorder. To change the role, call the action on the existing configuration recorder and specify a role.

Note: Currently, you can specify only one configuration recorder per account.

If `ConfigurationRecorder` does not have the **recordingGroup** parameter specified, the default is to record all supported resource types.

Request Syntax

```
response = client.put_configuration_recorder(
    ConfigurationRecorder={
        'name': 'string',
        'roleARN': 'string',
        'recordingGroup': {
            'allSupported': True|False,
            'resourceTypes': [
                'AWS::EC2::CustomerGateway' | 'AWS::EC2::EIP' | 'AWS::EC2::Instance' | 'AWS::EC2::'
            ]
        }
    }
)
```

Parameters ConfigurationRecorder (*dict*) – [REQUIRED]

The configuration recorder object that records each configuration change made to the resources.

- **name** (*string*) –

The name of the recorder. By default, AWS Config automatically assigns the name “default” when creating the configuration recorder. You cannot change the assigned name.

- **roleARN** (*string*) –

Amazon Resource Name (ARN) of the IAM role used to describe the AWS resources associated with the account.

- **recordingGroup** (*dict*) –

The recording group specifies either to record configurations for all supported resources or to provide a list of resource types to record. The list of resource types must be a subset of supported resource types.

- **allSupported** (*boolean*) –

Records all supported resource types in the recording group. For a list of supported resource types, see [Supported resource types](#). If you specify

allSupported , you cannot enumerate a list of **resourceTypes** .

– **resourceTypes** (*list*) –

A comma-separated list of strings representing valid AWS resource types (for example, `AWS::EC2::Instance` or `AWS::CloudTrail::Trail`). **resourceTypes** is only valid if you have chosen not to select **allSupported** . For a list of valid **resourceTypes** values, see the **resourceType Value** column in the following topic: [Supported AWS Resource Types](#) .

* (*string*) –

Returns None

put_delivery_channel (***kwargs*)

Creates a new delivery channel object to deliver the configuration information to an Amazon S3 bucket, and to an Amazon SNS topic.

You can use this action to change the Amazon S3 bucket or an Amazon SNS topic of the existing delivery channel. To change the Amazon S3 bucket or an Amazon SNS topic, call this action and specify the changed values for the S3 bucket and the SNS topic. If you specify a different value for either the S3 bucket or the SNS topic, this action will keep the existing value for the parameter that is not changed.

Note: Currently, you can specify only one delivery channel per account.

Request Syntax

```
response = client.put_delivery_channel(
    DeliveryChannel={
        'name': 'string',
        's3BucketName': 'string',
        's3KeyPrefix': 'string',
        'snsTopicARN': 'string'
    }
)
```

Parameters **DeliveryChannel** (*dict*) – [REQUIRED]

The configuration delivery channel object that delivers the configuration information to an Amazon S3 bucket, and to an Amazon SNS topic.

- **name** (*string*) –

The name of the delivery channel. By default, AWS Config automatically assigns the name “default” when creating the delivery channel. You cannot change the assigned name.

- **s3BucketName** (*string*) –

The name of the Amazon S3 bucket used to store configuration history for the delivery channel.

- **s3KeyPrefix** (*string*) –

The prefix for the specified Amazon S3 bucket.

- **snsTopicARN** (*string*) –

The Amazon Resource Name (ARN) of the SNS topic that AWS Config delivers notifications to.

Returns None

start_configuration_recorder (***kwargs*)

Starts recording configurations of the AWS resources you have selected to record in your AWS account.

You must have created at least one delivery channel to successfully start the configuration recorder.

Request Syntax

```
response = client.start_configuration_recorder(
    ConfigurationRecorderName='string'
)
```

Parameters **ConfigurationRecorderName** (*string*) – [REQUIRED]

The name of the recorder object that records each configuration change made to the resources.

Returns None

stop_configuration_recorder (***kwargs*)

Stops recording configurations of the AWS resources you have selected to record in your AWS account.

Request Syntax

```
response = client.stop_configuration_recorder(
    ConfigurationRecorderName='string'
)
```

Parameters **ConfigurationRecorderName** (*string*) – [REQUIRED]

The name of the recorder object that records each configuration change made to the resources.

Returns None

DataPipeline

Table of Contents

- DataPipeline
 - Client
 - Paginators

Client

class DataPipeline.**Client**

A low-level client representing AWS Data Pipeline:

```
import boto3

client = boto3.client('datapipeline')
```

These are the available methods:

- `activate_pipeline()`
- `add_tags()`
- `can_paginate()`
- `create_pipeline()`
- `deactivate_pipeline()`
- `delete_pipeline()`
- `describe_objects()`

- `describe_pipelines()`
- `evaluate_expression()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_pipeline_definition()`
- `get_waiter()`
- `list_pipelines()`
- `poll_for_task()`
- `put_pipeline_definition()`
- `query_objects()`
- `remove_tags()`
- `report_task_progress()`
- `report_task_runner_heartbeat()`
- `set_status()`
- `set_task_status()`
- `validate_pipeline_definition()`

activate_pipeline (***kwargs*)

Validates the specified pipeline and starts processing pipeline tasks. If the pipeline does not pass validation, activation fails.

If you need to pause the pipeline to investigate an issue with a component, such as a data source or script, call `DeactivatePipeline`.

To activate a finished pipeline, modify the end date for the pipeline and then activate it.

Request Syntax

```
response = client.activate_pipeline(
    pipelineId='string',
    parameterValues=[
        {
            'id': 'string',
            'stringValue': 'string'
        },
    ],
    startTimestamp=datetime(2015, 1, 1)
)
```

Parameters

- **pipelineId** (*string*) – [REQUIRED]

The ID of the pipeline.

- **parameterValues** (*list*) – A list of parameter values to pass to the pipeline at activation.

– (*dict*) –

A value or list of parameter values.

- * **id** (*string*) – [REQUIRED]

The ID of the parameter value.

- * **stringValue** (*string*) – [REQUIRED]

The field value, expressed as a String.

- **startTimestamp** (*datetime*) – The date and time to resume the pipeline. By default, the pipeline resumes from the last completed execution.

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- *(dict)* –

Contains the output of ActivatePipeline.

`add_tags` (***kwargs*)

Adds or modifies tags for the specified pipeline.

Request Syntax

```
response = client.add_tags(
    pipelineId='string',
    tags=[
        {
            'key': 'string',
            'value': 'string'
        },
    ]
)
```

Parameters

- **pipelineId** (*string*) – **[REQUIRED]**

The ID of the pipeline.

- **tags** (*list*) – **[REQUIRED]**

The tags to add, as key/value pairs.

– *(dict)* –

Tags are key/value pairs defined by a user and associated with a pipeline to control access. AWS Data Pipeline allows you to associate ten tags per pipeline. For more information, see [Controlling User Access to Pipelines](#) in the *AWS Data Pipeline Developer Guide*.

- * **key** (*string*) – **[REQUIRED]**

The key name of a tag defined by a user. For more information, see [Controlling User Access to Pipelines](#) in the *AWS Data Pipeline Developer Guide*.

- * **value** (*string*) – **[REQUIRED]**

The optional value portion of a tag defined by a user. For more information, see [Controlling User Access to Pipelines](#) in the *AWS Data Pipeline Developer Guide*.

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- *(dict)* –

Contains the output of AddTags.

`can_paginate` (*operation_name*)

Check if an operation can be paginated.

Parameters `operation_name` (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

create_pipeline (***kwargs*)

Creates a new, empty pipeline. Use `PutPipelineDefinition` to populate the pipeline.

Request Syntax

```
response = client.create_pipeline(
    name='string',
    uniqueId='string',
    description='string',
    tags=[
        {
            'key': 'string',
            'value': 'string'
        },
    ]
)
```

Parameters

- **name** (*string*) – [REQUIRED]

The name for the pipeline. You can use the same name for multiple pipelines associated with your AWS account, because AWS Data Pipeline assigns each pipeline a unique pipeline identifier.

- **uniqueId** (*string*) – [REQUIRED]

A unique identifier. This identifier is not the same as the pipeline identifier assigned by AWS Data Pipeline. You are responsible for defining the format and ensuring the uniqueness of this identifier. You use this parameter to ensure idempotency during repeated calls to `CreatePipeline`. For example, if the first call to `CreatePipeline` does not succeed, you can pass in the same unique identifier and pipeline name combination on a subsequent call to `CreatePipeline`. `CreatePipeline` ensures that if a pipeline already exists with the same name and unique identifier, a new pipeline is not created. Instead, you'll receive the pipeline identifier from the previous attempt. The uniqueness of the name and unique identifier combination is scoped to the AWS account or IAM user credentials.

- **description** (*string*) – The description for the pipeline.
- **tags** (*list*) – A list of tags to associate with the pipeline at creation. Tags let you control access to pipelines. For more information, see [Controlling User Access to Pipelines](#) in the *AWS Data Pipeline Developer Guide*.
 - (*dict*) –

Tags are key/value pairs defined by a user and associated with a pipeline to control access. AWS Data Pipeline allows you to associate ten tags per pipeline. For more information, see [Controlling User Access to Pipelines](#) in the *AWS Data Pipeline Developer Guide*.

- * **key** (*string*) – [REQUIRED]

The key name of a tag defined by a user. For more information, see [Controlling User Access to Pipelines](#) in the *AWS Data Pipeline Developer Guide*.

* **value** (*string*) – [REQUIRED]

The optional value portion of a tag defined by a user. For more information, see [Controlling User Access to Pipelines](#) in the *AWS Data Pipeline Developer Guide*.

Return type dict

Returns

Response Syntax

```
{
    'pipelineId': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output of CreatePipeline.

- **pipelineId** (*string*) –

The ID that AWS Data Pipeline assigns the newly created pipeline. For example, df-06372391ZG65EXAMPLE.

deactivate_pipeline (***kwargs*)

Deactivates the specified running pipeline. The pipeline is set to the DEACTIVATING state until the deactivation process completes.

To resume a deactivated pipeline, use `ActivatePipeline`. By default, the pipeline resumes from the last completed execution. Optionally, you can specify the date and time to resume the pipeline.

Request Syntax

```
response = client.deactivate_pipeline(
    pipelineId='string',
    cancelActive=True|False
)
```

Parameters

- **pipelineId** (*string*) – [REQUIRED]

The ID of the pipeline.

- **cancelActive** (*boolean*) – Indicates whether to cancel any running objects. The default is true, which sets the state of any running objects to CANCELED. If this value is false, the pipeline is deactivated after all running objects finish.

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –

Contains the output of DeactivatePipeline.

delete_pipeline (***kwargs*)

Deletes a pipeline, its pipeline definition, and its run history. AWS Data Pipeline attempts to cancel instances associated with the pipeline that are currently being processed by task runners.

Deleting a pipeline cannot be undone. You cannot query or restore a deleted pipeline. To temporarily pause a pipeline instead of deleting it, call `SetStatus` with the status set to `PAUSE` on individual components. Components that are paused by `SetStatus` can be resumed.

Request Syntax

```
response = client.delete_pipeline(  
    pipelineId='string'  
)
```

Parameters `pipelineId` (*string*) – [REQUIRED]

The ID of the pipeline.

Returns None

`describe_objects` (***kwargs*)

Gets the object definitions for a set of objects associated with the pipeline. Object definitions are composed of a set of fields that define the properties of the object.

Request Syntax

```
response = client.describe_objects(  
    pipelineId='string',  
    objectIds=[  
        'string',  
    ],  
    evaluateExpressions=True|False,  
    marker='string'  
)
```

Parameters

- **`pipelineId`** (*string*) – [REQUIRED]

The ID of the pipeline that contains the object definitions.

- **`objectIds`** (*list*) – [REQUIRED]

The IDs of the pipeline objects that contain the definitions to be described. You can pass as many as 25 identifiers in a single call to `DescribeObjects`.

– (*string*) –

- **`evaluateExpressions`** (*boolean*) – Indicates whether any expressions in the object should be evaluated when the object descriptions are returned.
- **`marker`** (*string*) – The starting point for the results to be returned. For the first call, this value should be empty. As long as there are more results, continue to call `DescribeObjects` with the marker value from the previous call to retrieve the next set of results.

Return type dict

Returns

Response Syntax

```
{  
    'pipelineObjects': [  
        {  
            'id': 'string',  
            'name': 'string',  
            'fields': [  
                {  
                    'key': 'string',  
                    'stringValue': 'string',
```

```

        'refValue': 'string'
    },
]
},
],
'marker': 'string',
'hasMoreResults': True|False
}

```

Response Structure

- (*dict*) –

Contains the output of DescribeObjects.

- **pipelineObjects** (*list*) –

An array of object definitions.

- * (*dict*) –

Contains information about a pipeline object. This can be a logical, physical, or physical attempt pipeline object. The complete set of components of a pipeline defines the pipeline.

- **id** (*string*) –

The ID of the object.

- **name** (*string*) –

The name of the object.

- **fields** (*list*) –

Key-value pairs that define the properties of the object.

- (*dict*) –

A key-value pair that describes a property of a pipeline object. The value is specified as either a string value (`StringValue`) or a reference to another object (`RefValue`) but not as both.

- **key** (*string*) –

The field identifier.

- **stringValue** (*string*) –

The field value, expressed as a String.

- **refValue** (*string*) –

The field value, expressed as the identifier of another object.

- **marker** (*string*) –

The starting point for the next page of results. To view the next page of results, call `DescribeObjects` again with this marker value. If the value is null, there are no more results.

- **hasMoreResults** (*boolean*) –

Indicates whether there are more results to return.

describe_pipelines (***kwargs*)

Retrieves metadata about one or more pipelines. The information retrieved includes the name of the pipeline, the pipeline identifier, its current state, and the user account that owns the pipeline. Using account credentials, you can retrieve metadata about pipelines that you or your IAM users have created. If you are using an IAM user account, you can retrieve metadata about only those pipelines for which you have read permissions.

To retrieve the full pipeline definition instead of metadata about the pipeline, call `GetPipelineDefinition`.

Request Syntax

```
response = client.describe_pipelines(
    pipelineIds=[
        'string',
    ]
)
```

Parameters `pipelineIds` (*list*) – [REQUIRED]

The IDs of the pipelines to describe. You can pass as many as 25 identifiers in a single call. To obtain pipeline IDs, call `ListPipelines`.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'pipelineDescriptionList': [
        {
            'pipelineId': 'string',
            'name': 'string',
            'fields': [
                {
                    'key': 'string',
                    'stringValue': 'string',
                    'refValue': 'string'
                },
            ],
            'description': 'string',
            'tags': [
                {
                    'key': 'string',
                    'value': 'string'
                },
            ],
        },
    ]
}
```

Response Structure

- (*dict*) –

Contains the output of `DescribePipelines`.

- **pipelineDescriptionList** (*list*) –

An array of descriptions for the specified pipelines.

- * (*dict*) –

Contains pipeline metadata.

- **pipelineId** (*string*) –

The pipeline identifier that was assigned by AWS Data Pipeline. This is a string of the form `df-297EG78HU43EEXAMPLE`.

- **name** (*string*) –
The name of the pipeline.
- **fields** (*list*) –
A list of read-only fields that contain metadata about the pipeline: @userId, @accountId, and @pipelineState.
- (*dict*) –
A key-value pair that describes a property of a pipeline object. The value is specified as either a string value (`StringValue`) or a reference to another object (`RefValue`) but not as both.
- **key** (*string*) –
The field identifier.
- **stringValue** (*string*) –
The field value, expressed as a String.
- **refValue** (*string*) –
The field value, expressed as the identifier of another object.
- **description** (*string*) –
Description of the pipeline.
- **tags** (*list*) –
A list of tags to associated with a pipeline. Tags let you control access to pipelines. For more information, see [Controlling User Access to Pipelines](#) in the *AWS Data Pipeline Developer Guide*.
- (*dict*) –
Tags are key/value pairs defined by a user and associated with a pipeline to control access. AWS Data Pipeline allows you to associate ten tags per pipeline. For more information, see [Controlling User Access to Pipelines](#) in the *AWS Data Pipeline Developer Guide*.
- **key** (*string*) –
The key name of a tag defined by a user. For more information, see [Controlling User Access to Pipelines](#) in the *AWS Data Pipeline Developer Guide*.
- **value** (*string*) –
The optional value portion of a tag defined by a user. For more information, see [Controlling User Access to Pipelines](#) in the *AWS Data Pipeline Developer Guide*.

evaluate_expression (**kwargs)

Task runners call `EvaluateExpression` to evaluate a string in the context of the specified object. For example, a task runner can evaluate SQL queries stored in Amazon S3.

Request Syntax

```
response = client.evaluate_expression(
    pipelineId='string',
    objectId='string',
    expression='string'
)
```

Parameters

- **pipelineId** (*string*) – [REQUIRED]
The ID of the pipeline.
- **objectId** (*string*) – [REQUIRED]
The ID of the object.
- **expression** (*string*) – [REQUIRED]
The expression to evaluate.

Return type dict**Returns****Response Syntax**

```
{  
    'evaluatedExpression': 'string'  
}
```

Response Structure

- (*dict*) –
Contains the output of EvaluateExpression.
 - **evaluatedExpression** (*string*) –
The evaluated expression.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to *ClientMethod*.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url**get_paginator** (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type L{botocore.paginate.Paginator}**Returns** A paginator object.**get_pipeline_definition** (***kwargs*)

Gets the definition of the specified pipeline. You can call `GetPipelineDefinition` to retrieve the pipeline definition that you provided using `PutPipelineDefinition`.

Request Syntax

```
response = client.get_pipeline_definition(  
    pipelineId='string',
```

```

    version='string'
)

```

Parameters

- **pipelineId** (*string*) – [REQUIRED]

The ID of the pipeline.

- **version** (*string*) – The version of the pipeline definition to retrieve. Set this parameter to `latest` (default) to use the last definition saved to the pipeline or `active` to use the last definition that was activated.

Return type dict

Returns

Response Syntax

```

{
  'pipelineObjects': [
    {
      'id': 'string',
      'name': 'string',
      'fields': [
        {
          'key': 'string',
          'stringValue': 'string',
          'refValue': 'string'
        },
      ],
    },
  ],
  'parameterObjects': [
    {
      'id': 'string',
      'attributes': [
        {
          'key': 'string',
          'stringValue': 'string'
        },
      ],
    },
  ],
  'parameterValues': [
    {
      'id': 'string',
      'stringValue': 'string'
    },
  ],
}

```

Response Structure

- (*dict*) –

Contains the output of `GetPipelineDefinition`.

- **pipelineObjects** (*list*) –

The objects defined in the pipeline.

- * (*dict*) –

Contains information about a pipeline object. This can be a logical, physical, or physical attempt pipeline object. The complete set of

components of a pipeline defines the pipeline.

- **id** (*string*) –

The ID of the object.

- **name** (*string*) –

The name of the object.

- **fields** (*list*) –

Key-value pairs that define the properties of the object.

- (*dict*) –

A key-value pair that describes a property of a pipeline object. The value is specified as either a string value (`StringValue`) or a reference to another object (`RefValue`) but not as both.

- **key** (*string*) –

The field identifier.

- **stringValue** (*string*) –

The field value, expressed as a String.

- **refValue** (*string*) –

The field value, expressed as the identifier of another object.

- **parameterObjects** (*list*) –

The parameter objects used in the pipeline definition.

- * (*dict*) –

Contains information about a parameter object.

- **id** (*string*) –

The ID of the parameter object.

- **attributes** (*list*) –

The attributes of the parameter object.

- (*dict*) –

The attributes allowed or specified with a parameter object.

- **key** (*string*) –

The field identifier.

- **stringValue** (*string*) –

The field value, expressed as a String.

- **parameterValues** (*list*) –

The parameter values used in the pipeline definition.

- * (*dict*) –

A value or list of parameter values.

- **id** (*string*) –

The ID of the parameter value.

- **stringValue** (*string*) –

The field value, expressed as a String.

get_waiter (*waiter_name*)

list_pipelines (***kwargs*)

Lists the pipeline identifiers for all active pipelines that you have permission to access.

Request Syntax

```
response = client.list_pipelines(
    marker='string'
)
```

Parameters **marker** (*string*) – The starting point for the results to be returned. For the first call, this value should be empty. As long as there are more results, continue to call `ListPipelines` with the marker value from the previous call to retrieve the next set of results.

Return type dict

Returns

Response Syntax

```
{
    'pipelineIdList': [
        {
            'id': 'string',
            'name': 'string'
        },
    ],
    'marker': 'string',
    'hasMoreResults': True|False
}
```

Response Structure

- (*dict*) –

Contains the output of `ListPipelines`.

- **pipelineIdList** (*list*) –

The pipeline identifiers. If you require additional information about the pipelines, you can use these identifiers to call `DescribePipelines` and `GetPipelineDefinition`.

- * (*dict*) –

Contains the name and identifier of a pipeline.

- **id** (*string*) –

The ID of the pipeline that was assigned by AWS Data Pipeline. This is a string of the form `df-297EG78HU43EEXAMPLE`.

- **name** (*string*) –

The name of the pipeline.

- **marker** (*string*) –

The starting point for the next page of results. To view the next page of results, call `ListPipelinesOutput` again with this marker value. If the value is null, there are no more results.

- **hasMoreResults** (*boolean*) –

Indicates whether there are more results that can be obtained by a subsequent call.

poll_for_task (***kwargs*)

Task runners call `PollForTask` to receive a task to perform from AWS Data Pipeline. The task runner specifies which tasks it can perform by setting a value for the `workerGroup` parameter. The task

returned can come from any of the pipelines that match the `workerGroup` value passed in by the task runner and that was launched using the IAM user credentials specified by the task runner.

If tasks are ready in the work queue, `PollForTask` returns a response immediately. If no tasks are available in the queue, `PollForTask` uses long-polling and holds on to a poll connection for up to a 90 seconds, during which time the first newly scheduled task is handed to the task runner. To accomodate this, set the socket timeout in your task runner to 90 seconds. The task runner should not call `PollForTask` again on the same `workerGroup` until it receives a response, and this can take up to 90 seconds.

Request Syntax

```
response = client.poll_for_task(  
    workerGroup='string',  
    hostname='string',  
    instanceIdentity={  
        'document': 'string',  
        'signature': 'string'  
    }  
)
```

Parameters

- **workerGroup** (*string*) – [REQUIRED]

The type of task the task runner is configured to accept and process. The worker group is set as a field on objects in the pipeline when they are created. You can only specify a single value for `workerGroup` in the call to `PollForTask`. There are no wildcard values permitted in `workerGroup`; the string must be an exact, case-sensitive, match.

- **hostname** (*string*) – The public DNS name of the calling task runner.

- **instanceIdentity** (*dict*) – Identity information for the EC2 instance that is hosting the task runner. You can get this value from the instance using `http://169.254.169.254/latest/meta-data/instance-id`. For more information, see [Instance Metadata](#) in the *Amazon Elastic Compute Cloud User Guide*. Passing in this value proves that your task runner is running on an EC2 instance, and ensures the proper AWS Data Pipeline service charges are applied to your pipeline.

- **document** (*string*) –

A description of an EC2 instance that is generated when the instance is launched and exposed to the instance via the instance metadata service in the form of a JSON representation of an object.

- **signature** (*string*) –

A signature which can be used to verify the accuracy and authenticity of the information provided in the instance identity document.

Return type dict

Returns

Response Syntax

```
{  
    'taskObject': {  
        'taskId': 'string',  
        'pipelineId': 'string',  
        'attemptId': 'string',  
        'objects': {  
            'string': {  
                'id': 'string',
```

```

        'name': 'string',
        'fields': [
            {
                'key': 'string',
                'stringValue': 'string',
                'refValue': 'string'
            },
        ]
    }
}

```

Response Structure

- *(dict)* –

Contains the output of PollForTask.

- **taskObject** (*dict*) –

The information needed to complete the task that is being assigned to the task runner. One of the fields returned in this object is `taskId`, which contains an identifier for the task being assigned. The calling task runner uses `taskId` in subsequent calls to `ReportTaskProgress` and `SetTaskStatus`.

- * **taskId** (*string*) –

An internal identifier for the task. This ID is passed to the `SetTaskStatus` and `ReportTaskProgress` actions.

- * **pipelineId** (*string*) –

The ID of the pipeline that provided the task.

- * **attemptId** (*string*) –

The ID of the pipeline task attempt object. AWS Data Pipeline uses this value to track how many times a task is attempted.

- * **objects** (*dict*) –

Connection information for the location where the task runner will publish the output of the task.

- (*string*) –
- (*dict*) –

Contains information about a pipeline object. This can be a logical, physical, or physical attempt pipeline object. The complete set of components of a pipeline defines the pipeline.

- **id** (*string*) –

The ID of the object.

- **name** (*string*) –

The name of the object.

- **fields** (*list*) –

Key-value pairs that define the properties of the object.

- (*dict*) –

A key-value pair that describes a property of a pipeline object. The value is specified as either a string value (`StringValue`) or a reference to another object (`RefValue`) but not as both.

- **key** (*string*) –
The field identifier.
- **stringValue** (*string*) –
The field value, expressed as a String.
- **refValue** (*string*) –
The field value, expressed as the identifier of another object.

put_pipeline_definition (**kwargs)

Adds tasks, schedules, and preconditions to the specified pipeline. You can use `PutPipelineDefinition` to populate a new pipeline.

`PutPipelineDefinition` also validates the configuration as it adds it to the pipeline. Changes to the pipeline are saved unless one of the following three validation errors exists in the pipeline.

- An object is missing a name or identifier field.
- A string or reference field is empty.
- The number of objects in the pipeline exceeds the maximum allowed objects.
- The pipeline is in a FINISHED state.

Pipeline object definitions are passed to the `PutPipelineDefinition` action and returned by the `GetPipelineDefinition` action.

Request Syntax

```
response = client.put_pipeline_definition(
    pipelineId='string',
    pipelineObjects=[
        {
            'id': 'string',
            'name': 'string',
            'fields': [
                {
                    'key': 'string',
                    'stringValue': 'string',
                    'refValue': 'string'
                }
            ],
        },
    ],
    parameterObjects=[
        {
            'id': 'string',
            'attributes': [
                {
                    'key': 'string',
                    'stringValue': 'string'
                }
            ],
        },
    ],
    parameterValues=[
        {
            'id': 'string',
            'stringValue': 'string'
        },
    ],
)
```


Parameters

- **pipelineId** (*string*) – [REQUIRED]

The ID of the pipeline.

- **pipelineObjects** (*list*) – [REQUIRED]

The objects that define the pipeline. These objects overwrite the existing pipeline definition.

– (*dict*) –

Contains information about a pipeline object. This can be a logical, physical, or physical attempt pipeline object. The complete set of components of a pipeline defines the pipeline.

- * **id** (*string*) – [REQUIRED]

The ID of the object.

- * **name** (*string*) – [REQUIRED]

The name of the object.

- * **fields** (*list*) – [REQUIRED]

Key-value pairs that define the properties of the object.

· (*dict*) –

A key-value pair that describes a property of a pipeline object. The value is specified as either a string value (`StringValue`) or a reference to another object (`RefValue`) but not as both.

- **key** (*string*) – [REQUIRED]

The field identifier.

- **stringValue** (*string*) –

The field value, expressed as a String.

- **refValue** (*string*) –

The field value, expressed as the identifier of another object.

- **parameterObjects** (*list*) – The parameter objects used with the pipeline.

– (*dict*) –

Contains information about a parameter object.

- * **id** (*string*) – [REQUIRED]

The ID of the parameter object.

- * **attributes** (*list*) – [REQUIRED]

The attributes of the parameter object.

· (*dict*) –

The attributes allowed or specified with a parameter object.

- **key** (*string*) – [REQUIRED]

The field identifier.

- **stringValue** (*string*) – [REQUIRED]

The field value, expressed as a String.

- **parameterValues** (*list*) – The parameter values used with the pipeline.

– (*dict*) –

A value or list of parameter values.

- * **id** (*string*) – [REQUIRED]

The ID of the parameter value.

* **stringValue** (*string*) – [REQUIRED]

The field value, expressed as a String.

Return type dict

Returns

Response Syntax

```
{
  'validationErrors': [
    {
      'id': 'string',
      'errors': [
        'string',
      ]
    },
  ],
  'validationWarnings': [
    {
      'id': 'string',
      'warnings': [
        'string',
      ]
    },
  ],
  'errored': True|False
}
```

Response Structure

- (*dict*) –

Contains the output of PutPipelineDefinition.

- **validationErrors** (*list*) –

The validation errors that are associated with the objects defined in `pipelineObjects`.

- * (*dict*) –

Defines a validation error. Validation errors prevent pipeline activation. The set of validation errors that can be returned are defined by AWS Data Pipeline.

- **id** (*string*) –

The identifier of the object that contains the validation error.

- **errors** (*list*) –

A description of the validation error.

- (*string*) –

- **validationWarnings** (*list*) –

The validation warnings that are associated with the objects defined in `pipelineObjects`.

- * (*dict*) –

Defines a validation warning. Validation warnings do not prevent pipeline activation. The set of validation warnings that can be returned are defined by AWS Data Pipeline.

- **id** (*string*) –

The identifier of the object that contains the validation warning.

- **warnings** (*list*) –
A description of the validation warning.
- (*string*) –
- **errored** (*boolean*) –
Indicates whether there were validation errors, and the pipeline definition is stored but cannot be activated until you correct the pipeline and call `PutPipelineDefinition` to commit the corrected pipeline.

query_objects (***kwargs*)

Queries the specified pipeline for the names of objects that match the specified set of conditions.

Request Syntax

```
response = client.query_objects(
    pipelineId='string',
    query={
        'selectors': [
            {
                'fieldName': 'string',
                'operator': {
                    'type': 'EQ' | 'REF_EQ' | 'LE' | 'GE' | 'BETWEEN',
                    'values': [
                        'string',
                    ]
                }
            }
        ],
        sphere='string',
        marker='string',
        limit=123
    )
```

Parameters

- **pipelineId** (*string*) – **[REQUIRED]**
The ID of the pipeline.
- **query** (*dict*) – The query that defines the objects to be returned. The `Query` object can contain a maximum of ten selectors. The conditions in the query are limited to top-level String fields in the object. These filters can be applied to components, instances, and attempts.
 - **selectors** (*list*) –
List of selectors that define the query. An object must satisfy all of the selectors to match the query.
 - * (*dict*) –
A comparison that is used to determine whether a query should return this object.
 - **fieldName** (*string*) –
The name of the field that the operator will be applied to. The field name is the “key” portion of the field definition in the pipeline definition syntax that is used by the AWS Data Pipeline API. If the field is not set on the object, the condition fails.
 - **operator** (*dict*) –

Contains a logical operation for comparing the value of a field with a specified value.

- **type** (*string*) –

The logical operation to be performed: equal (EQ), equal reference (REF_EQ), less than or equal (LE), greater than or equal (GE), or between (BETWEEN). Equal reference (REF_EQ) can be used only with reference fields. The other comparison types can be used only with String fields. The comparison types you can use apply only to certain object fields, as detailed below.

The comparison operators EQ and REF_EQ act on the following fields:

- name
- @sphere
- parent
- @componentParent
- @instanceParent
- @status
- @scheduledStartTime
- @scheduledEndTime
- @actualStartTime
- @actualEndTime

The comparison operators GE , LE , and BETWEEN act on the following fields:

- @scheduledStartTime
- @scheduledEndTime
- @actualStartTime
- @actualEndTime

Note that fields beginning with the at sign (@) are read-only and set by the web service. When you name fields, you should choose names containing only alpha-numeric values, as symbols may be reserved by AWS Data Pipeline. User-defined fields that you add to a pipeline should prefix their name with the string “my”.

- **values** (*list*) –

The value that the actual field value will be compared with.

- (*string*) –

- **sphere** (*string*) – **[REQUIRED]**

Indicates whether the query applies to components or instances. The possible values are: COMPONENT , INSTANCE , and ATTEMPT .

- **marker** (*string*) – The starting point for the results to be returned. For the first call, this value should be empty. As long as there are more results, continue to call `QueryObjects` with the marker value from the previous call to retrieve the next set of results.
- **limit** (*integer*) – The maximum number of object names that `QueryObjects` will return in a single call. The default value is 100.

Return type dict

Returns

Response Syntax

```
{
    'ids': [
        'string',
    ],
    'marker': 'string',
    'hasMoreResults': True|False
}
```

Response Structure

- *(dict)* –

Contains the output of QueryObjects.

- **ids** (*list*) –

The identifiers that match the query selectors.

- *(string)* –

- **marker** (*string*) –

The starting point for the next page of results. To view the next page of results, call QueryObjects again with this marker value. If the value is null, there are no more results.

- **hasMoreResults** (*boolean*) –

Indicates whether there are more results that can be obtained by a subsequent call.

remove_tags (***kwargs*)

Removes existing tags from the specified pipeline.

Request Syntax

```
response = client.remove_tags(
    pipelineId='string',
    tagKeys=[
        'string',
    ]
)
```

Parameters

- **pipelineId** (*string*) – [REQUIRED]

The ID of the pipeline.

- **tagKeys** (*list*) – [REQUIRED]

The keys of the tags to remove.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- *(dict)* –

Contains the output of RemoveTags.

report_task_progress (***kwargs*)

Task runners call ReportTaskProgress when assigned a task to acknowledge that it has the task. If

the web service does not receive this acknowledgement within 2 minutes, it assigns the task in a subsequent PollForTask call. After this initial acknowledgement, the task runner only needs to report progress every 15 minutes to maintain its ownership of the task. You can change this reporting time from 15 minutes by specifying a `reportProgressTimeout` field in your pipeline.

If a task runner does not report its status after 5 minutes, AWS Data Pipeline assumes that the task runner is unable to process the task and reassigns the task in a subsequent response to PollForTask . Task runners should call `ReportTaskProgress` every 60 seconds.

Request Syntax

```
response = client.report_task_progress(
    taskId='string',
    fields=[
        {
            'key': 'string',
            'stringValue': 'string',
            'refValue': 'string'
        },
    ]
)
```

Parameters

- **taskId** (*string*) – **[REQUIRED]**

The ID of the task assigned to the task runner. This value is provided in the response for PollForTask .

- **fields** (*list*) – Key-value pairs that define the properties of the ReportTaskProgressInput object.
 - (*dict*) –

A key-value pair that describes a property of a pipeline object. The value is specified as either a string value (`StringValue`) or a reference to another object (`RefValue`) but not as both.

- * **key** (*string*) – **[REQUIRED]**

The field identifier.

- * **stringValue** (*string*) –

The field value, expressed as a String.

- * **refValue** (*string*) –

The field value, expressed as the identifier of another object.

Return type dict

Returns

Response Syntax

```
{
    'canceled': True|False
}
```

Response Structure

- (*dict*) –

Contains the output of ReportTaskProgress.

- **canceled** (*boolean*) –

If true, the calling task runner should cancel processing of the task. The task runner does not need to call `SetTaskStatus` for canceled tasks.

report_task_runner_heartbeat (***kwargs*)

Task runners call `ReportTaskRunnerHeartbeat` every 15 minutes to indicate that they are operational. If the AWS Data Pipeline Task Runner is launched on a resource managed by AWS Data Pipeline, the web service can use this call to detect when the task runner application has failed and restart a new instance.

Request Syntax

```
response = client.report_task_runner_heartbeat(
    taskrunnerId='string',
    workerGroup='string',
    hostname='string'
)
```

Parameters

- **taskrunnerId** (*string*) – [REQUIRED]

The ID of the task runner. This value should be unique across your AWS account. In the case of AWS Data Pipeline Task Runner launched on a resource managed by AWS Data Pipeline, the web service provides a unique identifier when it launches the application. If you have written a custom task runner, you should assign a unique identifier for the task runner.

- **workerGroup** (*string*) – The type of task the task runner is configured to accept and process. The worker group is set as a field on objects in the pipeline when they are created. You can only specify a single value for `workerGroup`. There are no wildcard values permitted in `workerGroup`; the string must be an exact, case-sensitive, match.
- **hostname** (*string*) – The public DNS name of the task runner.

Return type dict

Returns**Response Syntax**

```
{
    'terminate': True|False
}
```

Response Structure

- (*dict*) –

Contains the output of `ReportTaskRunnerHeartbeat`.

- **terminate** (*boolean*) –

Indicates whether the calling task runner should terminate.

set_status (***kwargs*)

Requests that the status of the specified physical or logical pipeline objects be updated in the specified pipeline. This update might not occur immediately, but is eventually consistent. The status that can be set depends on the type of object (for example, `DataNode` or `Activity`). You cannot perform this operation on `FINISHED` pipelines and attempting to do so returns `InvalidRequestException`.

Request Syntax

```
response = client.set_status(
    pipelineId='string',
    objectIds=[
        'string',
    ],
)
```

```
status='string'  
)
```

Parameters

- **pipelineId** (*string*) – [REQUIRED]
The ID of the pipeline that contains the objects.
- **objectIds** (*list*) – [REQUIRED]
The IDs of the objects. The corresponding objects can be either physical or components, but not a mix of both types.
 - (*string*) –
- **status** (*string*) – [REQUIRED]
The status to be set on all the objects specified in `objectIds` . For components, use `PAUSE` or `RESUME` . For instances, use `TRY_CANCEL` , `RERUN` , or `MARK_FINISHED` .

Returns None

set_task_status (***kwargs*)

Task runners call `SetTaskStatus` to notify AWS Data Pipeline that a task is completed and provide information about the final status. A task runner makes this call regardless of whether the task was successful. A task runner does not need to call `SetTaskStatus` for tasks that are canceled by the web service during a call to `ReportTaskProgress` .

Request Syntax

```
response = client.set_task_status(  
    taskId='string',  
    taskStatus='FINISHED' | 'FAILED' | 'FALSE',  
    errorId='string',  
    errorMessage='string',  
    errorStackTrace='string'  
)
```

Parameters

- **taskId** (*string*) – [REQUIRED]
The ID of the task assigned to the task runner. This value is provided in the response for `PollForTask` .
- **taskStatus** (*string*) – [REQUIRED]
If `FINISHED` , the task successfully completed. If `FAILED` , the task ended unsuccessfully. Preconditions use false.
- **errorId** (*string*) – If an error occurred during the task, this value specifies the error code. This value is set on the physical attempt object. It is used to display error information to the user. It should not start with string “**Service_**” which is reserved by the system.
- **errorMessage** (*string*) – If an error occurred during the task, this value specifies a text description of the error. This value is set on the physical attempt object. It is used to display error information to the user. The web service does not parse this value.
- **errorStackTrace** (*string*) – If an error occurred during the task, this value specifies the stack trace associated with the error. This value is set on the physical attempt object. It is used to display error information to the user. The web service does not parse this value.

Return type dict

Returns**Response Syntax**

```
{}
```

Response Structure

- (*dict*) –
Contains the output of SetTaskStatus.

validate_pipeline_definition (***kwargs*)

Validates the specified pipeline definition to ensure that it is well formed and can be run without error.

Request Syntax

```
response = client.validate_pipeline_definition(
    pipelineId='string',
    pipelineObjects=[
        {
            'id': 'string',
            'name': 'string',
            'fields': [
                {
                    'key': 'string',
                    'stringValue': 'string',
                    'refValue': 'string'
                },
            ],
        },
    ],
    parameterObjects=[
        {
            'id': 'string',
            'attributes': [
                {
                    'key': 'string',
                    'stringValue': 'string'
                },
            ],
        },
    ],
    parameterValues=[
        {
            'id': 'string',
            'stringValue': 'string'
        },
    ],
)
```

Parameters

- **pipelineId** (*string*) – **[REQUIRED]**
The ID of the pipeline.
- **pipelineObjects** (*list*) – **[REQUIRED]**
The objects that define the pipeline changes to validate against the pipeline.
– (*dict*) –

Contains information about a pipeline object. This can be a logical, physical, or physical attempt pipeline object. The complete set of components of a pipeline defines the pipeline.

- * **id** (*string*) – [REQUIRED]

The ID of the object.

- * **name** (*string*) – [REQUIRED]

The name of the object.

- * **fields** (*list*) – [REQUIRED]

Key-value pairs that define the properties of the object.

- (*dict*) –

A key-value pair that describes a property of a pipeline object. The value is specified as either a string value (`StringValue`) or a reference to another object (`RefValue`) but not as both.

- **key** (*string*) – [REQUIRED]

The field identifier.

- **stringValue** (*string*) –

The field value, expressed as a String.

- **refValue** (*string*) –

The field value, expressed as the identifier of another object.

- **parameterObjects** (*list*) – The parameter objects used with the pipeline.

- (*dict*) –

Contains information about a parameter object.

- * **id** (*string*) – [REQUIRED]

The ID of the parameter object.

- * **attributes** (*list*) – [REQUIRED]

The attributes of the parameter object.

- (*dict*) –

The attributes allowed or specified with a parameter object.

- **key** (*string*) – [REQUIRED]

The field identifier.

- **stringValue** (*string*) – [REQUIRED]

The field value, expressed as a String.

- **parameterValues** (*list*) – The parameter values used with the pipeline.

- (*dict*) –

A value or list of parameter values.

- * **id** (*string*) – [REQUIRED]

The ID of the parameter value.

- * **stringValue** (*string*) – [REQUIRED]

The field value, expressed as a String.

Return type dict

Returns

Response Syntax

```
{
  'validationErrors': [
    {
      'id': 'string',
      'errors': [
        'string',
      ]
    },
  ],
  'validationWarnings': [
    {
      'id': 'string',
      'warnings': [
        'string',
      ]
    },
  ],
  'errored': True|False
}
```

Response Structure

- (*dict*) –

Contains the output of `ValidatePipelineDefinition`.

- **validationErrors** (*list*) –

Any validation errors that were found.

- * (*dict*) –

Defines a validation error. Validation errors prevent pipeline activation. The set of validation errors that can be returned are defined by AWS Data Pipeline.

- **id** (*string*) –

The identifier of the object that contains the validation error.

- **errors** (*list*) –

A description of the validation error.

- (*string*) –

- **validationWarnings** (*list*) –

Any validation warnings that were found.

- * (*dict*) –

Defines a validation warning. Validation warnings do not prevent pipeline activation. The set of validation warnings that can be returned are defined by AWS Data Pipeline.

- **id** (*string*) –

The identifier of the object that contains the validation warning.

- **warnings** (*list*) –

A description of the validation warning.

- (*string*) –

- **errored** (*boolean*) –

Indicates whether there were validation errors.

Paginators

The available paginators are:

- `DataPipeline.Paginator.DescribeObjects`
- `DataPipeline.Paginator.ListPipelines`
- `DataPipeline.Paginator.QueryObjects`

class `DataPipeline.Paginator.DescribeObjects`

```
paginator = client.get_paginator('describe_objects')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `DataPipeline.Client.describe_objects()`.

Request Syntax

```
response_iterator = paginator.paginate(
    pipelineId='string',
    objectIds=[
        'string',
    ],
    evaluateExpressions=True|False,
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **pipelineId** (*string*) – [REQUIRED]

The ID of the pipeline that contains the object definitions.

- **objectIds** (*list*) – [REQUIRED]

The IDs of the pipeline objects that contain the definitions to be described. You can pass as many as 25 identifiers in a single call to `DescribeObjects`.

– (*string*) –

- **evaluateExpressions** (*boolean*) – Indicates whether any expressions in the object should be evaluated when the object descriptions are returned.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

– **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

– **PageSize** (*integer*) –

The size of each page.

– **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
  'pipelineObjects': [
    {
      'id': 'string',
      'name': 'string',
      'fields': [
        {
          'key': 'string',
          'stringValue': 'string',
          'refValue': 'string'
        },
        ...
      ]
    },
    ...
  ],
  'hasMoreResults': True|False,
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output of DescribeObjects.

- **pipelineObjects** (*list*) –

An array of object definitions.

- * (*dict*) –

Contains information about a pipeline object. This can be a logical, physical, or physical attempt pipeline object. The complete set of components of a pipeline defines the pipeline.

- **id** (*string*) –

The ID of the object.

- **name** (*string*) –

The name of the object.

- **fields** (*list*) –

Key-value pairs that define the properties of the object.

- (*dict*) –

A key-value pair that describes a property of a pipeline object. The value is specified as either a string value (`StringValue`) or a reference to another object (`RefValue`) but not as both.

- **key** (*string*) –

The field identifier.

- **stringValue** (*string*) –

The field value, expressed as a String.

- **refValue** (*string*) –

The field value, expressed as the identifier of another object.

- **hasMoreResults** (*boolean*) –

Indicates whether there are more results to return.

– **NextToken** (*string*) –

A token to resume pagination.

class `DataPipeline.Paginator.ListPipelines`

```
paginator = client.get_paginator('list_pipelines')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `DataPipeline.Client.list_pipelines()`.

Request Syntax

```
response_iterator = paginator.paginate(
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
    'pipelineIdList': [
        {
            'id': 'string',
            'name': 'string'
        },
    ],
    'hasMoreResults': True|False,
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output of `ListPipelines`.

– **pipelineIdList** (*list*) –

The pipeline identifiers. If you require additional information about the pipelines, you can use these identifiers to call `DescribePipelines` and `GetPipelineDefinition`.

* (*dict*) –

Contains the name and identifier of a pipeline.

· **id** (*string*) –

The ID of the pipeline that was assigned by AWS Data Pipeline. This is a string of the form `df-297EG78HU43EEXAMPLE`.

· **name** (*string*) –

The name of the pipeline.

– **hasMoreResults** (*boolean*) –

Indicates whether there are more results that can be obtained by a subsequent call.

– **NextToken** (*string*) –

A token to resume pagination.

class `DataPipeline.Paginator.QueryObjects`

```
paginator = client.get_paginator('query_objects')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `DataPipeline.Client.query_objects()`.

Request Syntax

```
response_iterator = paginator.paginate(
    pipelineId='string',
    query={
        'selectors': [
            {
                'fieldName': 'string',
                'operator': {
                    'type': 'EQ' | 'REF_EQ' | 'LE' | 'GE' | 'BETWEEN',
                    'values': [
                        'string',
                    ]
                }
            }
        ],
        'sphere': 'string',
        'PaginationConfig': {
            'MaxItems': 123,
            'PageSize': 123,
            'StartingToken': 'string'
        }
    )
```

Parameters

- **pipelineId** (*string*) – [REQUIRED]

The ID of the pipeline.

- **query** (*dict*) – The query that defines the objects to be returned. The `Query` object can contain a maximum of ten selectors. The conditions in the query are limited to top-level String fields in the object. These filters can be applied to components, instances, and attempts.

- **selectors** (*list*) –

List of selectors that define the query. An object must satisfy all of the selectors to match the query.

- * (*dict*) –

A comparison that is used to determine whether a query should return this object.

- **fieldName** (*string*) –

The name of the field that the operator will be applied to. The field name is the “key” portion of the field definition in the pipeline definition syntax that is used by the AWS Data Pipeline API. If the field is not set on the object, the condition fails.

- **operator** (*dict*) –

Contains a logical operation for comparing the value of a field with a specified value.

- **type** (*string*) –

The logical operation to be performed: `equal` (`EQ`), `equal reference` (`REF_EQ`), `less than or equal` (`LE`), `greater than or equal` (`GE`), or `between` (`BETWEEN`). `Equal reference` (`REF_EQ`) can be used only with reference fields. The other comparison types can be used only with String fields. The comparison types you can use apply only to certain object fields, as detailed below.

The comparison operators `EQ` and `REF_EQ` act on the following fields:

- `name`
- `@sphere`
- `parent`
- `@componentParent`
- `@instanceParent`
- `@status`
- `@scheduledStartTime`
- `@scheduledEndTime`
- `@actualStartTime`
- `@actualEndTime`

The comparison operators `GE`, `LE`, and `BETWEEN` act on the following fields:

- `@scheduledStartTime`
- `@scheduledEndTime`
- `@actualStartTime`
- `@actualEndTime`

Note that fields beginning with the at sign (`@`) are read-only and set by the web service. When you name fields, you should choose names containing only alpha-numeric values, as sym-

bols may be reserved by AWS Data Pipeline. User-defined fields that you add to a pipeline should prefix their name with the string “my”.

- **values** (*list*) –

The value that the actual field value will be compared with.

- (*string*) –

- **sphere** (*string*) – **[REQUIRED]**

Indicates whether the query applies to components or instances. The possible values are: `COMPONENT`, `INSTANCE`, and `ATTEMPT`.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
  'ids': [
    'string',
  ],
  'hasMoreResults': True|False,
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output of `QueryObjects`.

- **ids** (*list*) –

The identifiers that match the query selectors.

- (*string*) –

- **hasMoreResults** (*boolean*) –

Indicates whether there are more results that can be obtained by a subsequent call.

- **NextToken** (*string*) –

A token to resume pagination.

DeviceFarm

Table of Contents

- [DeviceFarm](#)
 - [Client](#)

Client**class DeviceFarm.Client**

A low-level client representing AWS Device Farm:

```
import boto3
```

```
client = boto3.client('devicefarm')
```

These are the available methods:

- `can_paginate()`
- `create_device_pool()`
- `create_project()`
- `create_upload()`
- `generate_presigned_url()`
- `get_account_settings()`
- `get_device()`
- `get_device_pool()`
- `get_device_pool_compatibility()`
- `get_job()`
- `get_paginator()`
- `get_project()`
- `get_run()`
- `get_suite()`
- `get_test()`
- `get_upload()`
- `get_waiter()`
- `list_artifacts()`
- `list_device_pools()`
- `list_devices()`
- `list_jobs()`
- `list_projects()`
- `list_runs()`
- `list_samples()`
- `list_suites()`
- `list_tests()`
- `list_unique_problems()`
- `list_uploads()`
- `schedule_run()`

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

create_device_pool (**kwargs)

Creates a device pool.

Request Syntax

```
response = client.create_device_pool(
    projectArn='string',
    name='string',
    description='string',
    rules=[
        {
            'attribute': 'ARN' | 'PLATFORM' | 'FORM_FACTOR' | 'MANUFACTURER',
            'operator': 'EQUALS' | 'LESS_THAN' | 'GREATER_THAN' | 'IN' | 'NOT_IN',
            'value': 'string'
        },
    ]
)
```

Parameters

- **projectArn** (*string*) – **[REQUIRED]**

The ARN of the project for the device pool.

- **name** (*string*) – **[REQUIRED]**

The device pool's name.

- **description** (*string*) – The device pool's description.

- **rules** (*list*) – **[REQUIRED]**

The device pool's rules.

– (*dict*) –

Represents a condition for a device pool.

* **attribute** (*string*) –

The rule's attribute.

Allowed values include:

- ARN: The ARN.
- FORM_FACTOR: The form factor (for example, phone or tablet).
- MANUFACTURER: The manufacturer.
- PLATFORM: The platform (for example, Android or iOS).

* **operator** (*string*) –

The rule's operator.

- EQUALS: The equals operator.
- GREATER_THAN: The greater-than operator.
- IN: The in operator.
- LESS_THAN: The less-than operator.
- NOT_IN: The not-in operator.

* **value** (*string*) –

The rule's value.

Return type dict

Returns

Response Syntax

```
{
    'devicePool': {
```

```
'arn': 'string',
'name': 'string',
'description': 'string',
'type': 'CURATED' | 'PRIVATE',
'rules': [
    {
        'attribute': 'ARN' | 'PLATFORM' | 'FORM_FACTOR' | 'MANUFACTURER',
        'operator': 'EQUALS' | 'LESS_THAN' | 'GREATER_THAN' | 'IN' | 'NOT_IN',
        'value': 'string'
    },
]
}
```

Response Structure

- *(dict)* –

Represents the result of a create device pool request.

- **devicePool** (*dict*) –

The newly created device pool.

- * **arn** (*string*) –

The device pool's ARN.

- * **name** (*string*) –

The device pool's name.

- * **description** (*string*) –

The device pool's description.

- * **type** (*string*) –

The device pool's type.

Allowed values include:

- CURATED: A device pool that is created and managed by AWS Device Farm.
- PRIVATE: A device pool that is created and managed by the device pool developer.

- * **rules** (*list*) –

Information about the device pool's rules.

- *(dict)* –

Represents a condition for a device pool.

- **attribute** (*string*) –

The rule's attribute.

Allowed values include:

- ARN: The ARN.
- FORM_FACTOR: The form factor (for example, phone or tablet).
- MANUFACTURER: The manufacturer.
- PLATFORM: The platform (for example, Android or iOS).
- **operator** (*string*) –

The rule's operator.

- EQUALS: The equals operator.
- GREATER_THAN: The greater-than operator.
- IN: The in operator.

- LESS_THAN: The less-than operator.
- NOT_IN: The not-in operator.
- **value** (*string*) –
The rule's value.

create_project (***kwargs*)

Creates a new project.

Request Syntax

```
response = client.create_project(
    name='string'
)
```

Parameters *name* (*string*) – [REQUIRED]

The project's name.

Return type dict

Returns

Response Syntax

```
{
  'project': {
    'arn': 'string',
    'name': 'string',
    'created': datetime(2015, 1, 1)
  }
}
```

Response Structure

- (*dict*) –

Represents the result of a create project request.

– **project** (*dict*) –

The newly created project.

* **arn** (*string*) –

The project's ARN.

* **name** (*string*) –

The project's name.

* **created** (*datetime*) –

When the project was created.

create_upload (***kwargs*)

Uploads an app or test scripts.

Request Syntax

```
response = client.create_upload(
    projectArn='string',
    name='string',
    type='ANDROID_APP'|'IOS_APP'|'EXTERNAL_DATA'|'APPIUM_JAVA_JUNIT_TEST_PACKAGE'|'APPIUM_J
    contentType='string'
)
```

Parameters

- **projectArn** (*string*) – [REQUIRED]

The ARN of the project for the upload.

- **name** (*string*) – [REQUIRED]

The upload's file name.

- **type** (*string*) – [REQUIRED]

The upload's upload type.

Must be one of the following values:

- **ANDROID_APP**: An Android upload.
 - **IOS_APP**: An iOS upload.
 - **EXTERNAL_DATA**: An external data upload.
 - **APPIUM_JAVA_JUNIT_TEST_PACKAGE**: An Appium Java JUnit test package upload.
 - **APPIUM_JAVA_TESTNG_TEST_PACKAGE**: An Appium Java TestNG test package upload.
 - **CALABASH_TEST_PACKAGE**: A Calabash test package upload.
 - **INSTRUMENTATION_TEST_PACKAGE**: An instrumentation upload.
 - **UIAUTOMATOR_TEST_PACKAGE**: A uiautomator test package upload.
 - **XCTEST_TEST_PACKAGE**: An XCode test package upload.
- **contentType** (*string*) – The upload's content type (for example, "application/octet-stream").

Return type dict

Returns

Response Syntax

```
{
  'upload': {
    'arn': 'string',
    'name': 'string',
    'created': datetime(2015, 1, 1),
    'type': 'ANDROID_APP'|'IOS_APP'|'EXTERNAL_DATA'|'APPIUM_JAVA_JUNIT_TEST_PA
    'status': 'INITIALIZED'|'PROCESSING'|'SUCCEEDED'|'FAILED',
    'url': 'string',
    'metadata': 'string',
    'contentType': 'string',
    'message': 'string'
  }
}
```

Response Structure

- (*dict*) –

Represents the result of a create upload request.

- **upload** (*dict*) –

The newly created upload.

- * **arn** (*string*) –

The upload's ARN.

- * **name** (*string*) –

The upload's file name.

- * **created** (*datetime*) –

When the upload was created.

* **type** (*string*) –

The upload's type.

Must be one of the following values:

- **ANDROID_APP**: An Android upload.
- **IOS_APP**: An iOS upload.
- **EXTERNAL_DATA**: An external data upload.
- **APPIUM_JAVA_JUNIT_TEST_PACKAGE**: An Appium Java JUnit test package upload.
- **APPIUM_JAVA_TESTNG_TEST_PACKAGE**: An Appium Java TestNG test package upload.
- **CALABASH_TEST_PACKAGE**: A Calabash test package upload.
- **INSTRUMENTATION_TEST_PACKAGE**: An instrumentation upload.
- **UIAUTOMATOR_TEST_PACKAGE**: A uiautomator test package upload.
- **XCTEST_TEST_PACKAGE**: An XCode test package upload.

* **status** (*string*) –

The upload's status.

Must be one of the following values:

- **FAILED**: A failed status.
- **INITIALIZED**: An initialized status.
- **PROCESSING**: A processing status.
- **SUCCEEDED**: A succeeded status.

* **url** (*string*) –

The pre-signed Amazon S3 URL that was used to store a file through a corresponding PUT request.

* **metadata** (*string*) –

The upload's metadata. For example, for Android, this contains information that is parsed from the manifest and is displayed in the AWS Device Farm console after the associated app is uploaded.

* **contentType** (*string*) –

The upload's content type (for example, "application/octet-stream").

* **message** (*string*) –

A message about the upload's result.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to *ClientMethod*.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_account_settings ()

Returns the number of unmetered iOS and/or unmetered Android devices that have been purchased by the

account.

Request Syntax

```
response = client.get_account_settings()
```

Return type dict

Returns

Response Syntax

```
{
    'accountSettings': {
        'awsAccountNumber': 'string',
        'unmeteredDevices': {
            'string': 123
        }
    }
}
```

Response Structure

- (dict) –

- **accountSettings** (dict) –

A container for account-level settings within AWS Device Farm.

- * **awsAccountNumber** (string) –

The AWS account number specified in the AccountSettings container.

- * **unmeteredDevices** (dict) –

Returns the unmetered devices you have purchased.

- (string) –

- (integer) –

get_device (**kwargs)

Gets information about a unique device type.

Request Syntax

```
response = client.get_device(
    arn='string'
)
```

Parameters **arn** (string) – [REQUIRED]

The device type's ARN.

Return type dict

Returns

Response Syntax

```
{
    'device': {
        'arn': 'string',
        'name': 'string',
        'manufacturer': 'string',
        'model': 'string',
        'formFactor': 'PHONE'|'TABLET',
        'platform': 'ANDROID'|'IOS',
    }
}
```



```

        'os': 'string',
        'cpu': {
            'frequency': 'string',
            'architecture': 'string',
            'clock': 123.0
        },
        'resolution': {
            'width': 123,
            'height': 123
        },
        'heapSize': 123,
        'memory': 123,
        'image': 'string',
        'carrier': 'string',
        'radio': 'string'
    }
}

```

Response Structure

- *(dict)* –

Represents the result of a get device request.

- **device** *(dict)* –

Represents a device type that an app is tested against.

- * **arn** *(string)* –

The device's ARN.

- * **name** *(string)* –

The device's display name.

- * **manufacturer** *(string)* –

The device's manufacturer name.

- * **model** *(string)* –

The device's model name.

- * **formFactor** *(string)* –

The device's form factor.

Allowed values include:

- PHONE: The phone form factor.
- TABLET: The tablet form factor.

- * **platform** *(string)* –

The device's platform.

Allowed values include:

- ANDROID: The Android platform.
- IOS: The iOS platform.

- * **os** *(string)* –

The device's operating system type.

- * **cpu** *(dict)* –

Information about the device's CPU.

- **frequency** *(string)* –

The CPU's frequency.

- **architecture** *(string)* –

The CPU's architecture, for example x86 or ARM.

- **clock** (*float*) –

The clock speed of the device's CPU, expressed in hertz (Hz).

For example, a 1.2 GHz CPU is expressed as 1200000000.

- * **resolution** (*dict*) –

Represents the screen resolution of a device in height and width, expressed in pixels.

- **width** (*integer*) –

The screen resolution's width, expressed in pixels.

- **height** (*integer*) –

The screen resolution's height, expressed in pixels.

- * **heapSize** (*integer*) –

The device's heap size, expressed in bytes.

- * **memory** (*integer*) –

The device's total memory size, expressed in bytes.

- * **image** (*string*) –

The device's image name.

- * **carrier** (*string*) –

The device's carrier.

- * **radio** (*string*) –

The device's radio.

get_device_pool (***kwargs*)

Gets information about a device pool.

Request Syntax

```
response = client.get_device_pool(
    arn='string'
)
```

Parameters **arn** (*string*) – [REQUIRED]

The device pool's ARN.

Return type dict

Returns

Response Syntax

```
{
  'devicePool': {
    'arn': 'string',
    'name': 'string',
    'description': 'string',
    'type': 'CURATED' | 'PRIVATE',
    'rules': [
      {
        'attribute': 'ARN' | 'PLATFORM' | 'FORM_FACTOR' | 'MANUFACTURER',
        'operator': 'EQUALS' | 'LESS_THAN' | 'GREATER_THAN' | 'IN' | 'NOT_IN',
        'value': 'string'
      },
    ],
  },
}
```

```
    }
}
```

Response Structure

- *(dict)* –

Represents the result of a get device pool request.

- **devicePool** (*dict*) –

Represents a collection of device types.

- * **arn** (*string*) –

The device pool's ARN.

- * **name** (*string*) –

The device pool's name.

- * **description** (*string*) –

The device pool's description.

- * **type** (*string*) –

The device pool's type.

Allowed values include:

- CURATED: A device pool that is created and managed by AWS Device Farm.
- PRIVATE: A device pool that is created and managed by the device pool developer.

- * **rules** (*list*) –

Information about the device pool's rules.

- (*dict*) –

Represents a condition for a device pool.

- **attribute** (*string*) –

The rule's attribute.

Allowed values include:

- ARN: The ARN.
- FORM_FACTOR: The form factor (for example, phone or tablet).
- MANUFACTURER: The manufacturer.
- PLATFORM: The platform (for example, Android or iOS).
- **operator** (*string*) –

The rule's operator.

- EQUALS: The equals operator.
- GREATER_THAN: The greater-than operator.
- IN: The in operator.
- LESS_THAN: The less-than operator.
- NOT_IN: The not-in operator.

- **value** (*string*) –

The rule's value.

get_device_pool_compatibility (***kwargs*)

Gets information about compatibility with a device pool.

Request Syntax

```
response = client.get_device_pool_compatibility(
    devicePoolArn='string',
    appArn='string',
    testType='BUILTIN_FUZZ' | 'BUILTIN_EXPLORER' | 'APPIUM_JAVA_JUNIT' | 'APPIUM_JAVA_TESTNG' | 'CA
)
```

Parameters

- **devicePoolArn** (*string*) – [REQUIRED]

The device pool's ARN.

- **appArn** (*string*) – [REQUIRED]

The ARN of the app that is associated with the specified device pool.

- **testType** (*string*) – The test type for the specified device pool.

Allowed values include the following:

- BUILTIN_FUZZ: The built-in fuzz type.
- BUILTIN_EXPLORER: For Android, an app explorer that will traverse an Android app, interacting with it and capturing screenshots at the same time.
- APPIUM_JAVA_JUNIT: The Appium Java JUnit type.
- APPIUM_JAVA_TESTNG: The Appium Java TestNG type.
- CALABASH: The Calabash type.
- INSTRUMENTATION: The Instrumentation type.
- UIAUTOMATION: The uiautomation type.
- UIAUTOMATOR: The uiautomator type.
- XCTEST: The XCode test type.

Return type dict

Returns

Response Syntax

```
{
  'compatibleDevices': [
    {
      'device': {
        'arn': 'string',
        'name': 'string',
        'manufacturer': 'string',
        'model': 'string',
        'formFactor': 'PHONE' | 'TABLET',
        'platform': 'ANDROID' | 'IOS',
        'os': 'string',
        'cpu': {
          'frequency': 'string',
          'architecture': 'string',
          'clock': 123.0
        },
        'resolution': {
          'width': 123,
          'height': 123
        },
        'heapSize': 123,
        'memory': 123,
        'image': 'string',
        'carrier': 'string',
        'radio': 'string'
      }
    }
  ]
}
```

```

    },
    'compatible': True|False,
    'incompatibilityMessages': [
        {
            'message': 'string',
            'type': 'ARN'|'PLATFORM'|'FORM_FACTOR'|'MANUFACTURER'
        },
    ]
},
],
'incompatibleDevices': [
    {
        'device': {
            'arn': 'string',
            'name': 'string',
            'manufacturer': 'string',
            'model': 'string',
            'formFactor': 'PHONE'|'TABLET',
            'platform': 'ANDROID'|'IOS',
            'os': 'string',
            'cpu': {
                'frequency': 'string',
                'architecture': 'string',
                'clock': 123.0
            },
            'resolution': {
                'width': 123,
                'height': 123
            },
            'heapSize': 123,
            'memory': 123,
            'image': 'string',
            'carrier': 'string',
            'radio': 'string'
        },
        'compatible': True|False,
        'incompatibilityMessages': [
            {
                'message': 'string',
                'type': 'ARN'|'PLATFORM'|'FORM_FACTOR'|'MANUFACTURER'
            },
        ]
    },
]
}

```

Response Structure

- (*dict*) –

Represents the result of describe device pool compatibility request.

- **compatibleDevices** (*list*) –

Information about compatible devices.

- * (*dict*) –

Represents a device pool compatibility result.

- **device** (*dict*) –

Represents a device type that an app is tested against.

- **arn** (*string*) –
The device's ARN.
- **name** (*string*) –
The device's display name.
- **manufacturer** (*string*) –
The device's manufacturer name.
- **model** (*string*) –
The device's model name.
- **formFactor** (*string*) –
The device's form factor.

Allowed values include:
 - PHONE: The phone form factor.
 - TABLET: The tablet form factor.
- **platform** (*string*) –
The device's platform.

Allowed values include:
 - ANDROID: The Android platform.
 - IOS: The iOS platform.
- **os** (*string*) –
The device's operating system type.
- **cpu** (*dict*) –
Information about the device's CPU.
- **frequency** (*string*) –
The CPU's frequency.
- **architecture** (*string*) –
The CPU's architecture, for example x86 or ARM.
- **clock** (*float*) –
The clock speed of the device's CPU, expressed in hertz (Hz).
For example, a 1.2 GHz CPU is expressed as 1200000000.
- **resolution** (*dict*) –
Represents the screen resolution of a device in height and width, expressed in pixels.
- **width** (*integer*) –
The screen resolution's width, expressed in pixels.
- **height** (*integer*) –
The screen resolution's height, expressed in pixels.
- **heapSize** (*integer*) –
The device's heap size, expressed in bytes.
- **memory** (*integer*) –
The device's total memory size, expressed in bytes.
- **image** (*string*) –
The device's image name.
- **carrier** (*string*) –
The device's carrier.

- **radio** (*string*) –
The device’s radio.
- **compatible** (*boolean*) –
Whether the result was compatible with the device pool.
- **incompatibilityMessages** (*list*) –
Information about the compatibility.
- (*dict*) –
Represents information about incompatibility.
- **message** (*string*) –
A message about the incompatibility.
- **type** (*string*) –
The type of incompatibility.
Allowed values include:
 - ARN: The ARN.
 - FORM_FACTOR: The form factor (for example, phone or tablet).
 - MANUFACTURER: The manufacturer.
 - PLATFORM: The platform (for example, Android or iOS).
- **incompatibleDevices** (*list*) –
Information about incompatible devices.
 - * (*dict*) –
Represents a device pool compatibility result.
 - **device** (*dict*) –
Represents a device type that an app is tested against.
 - **arn** (*string*) –
The device’s ARN.
 - **name** (*string*) –
The device’s display name.
 - **manufacturer** (*string*) –
The device’s manufacturer name.
 - **model** (*string*) –
The device’s model name.
 - **formFactor** (*string*) –
The device’s form factor.
Allowed values include:
 - PHONE: The phone form factor.
 - TABLET: The tablet form factor.
 - **platform** (*string*) –
The device’s platform.
Allowed values include:
 - ANDROID: The Android platform.
 - IOS: The iOS platform.
 - **os** (*string*) –
The device’s operating system type.

- **cpu** (*dict*) –
Information about the device's CPU.
- **frequency** (*string*) –
The CPU's frequency.
- **architecture** (*string*) –
The CPU's architecture, for example x86 or ARM.
- **clock** (*float*) –
The clock speed of the device's CPU, expressed in hertz (Hz).
For example, a 1.2 GHz CPU is expressed as 1200000000.
- **resolution** (*dict*) –
Represents the screen resolution of a device in height and width, expressed in pixels.
- **width** (*integer*) –
The screen resolution's width, expressed in pixels.
- **height** (*integer*) –
The screen resolution's height, expressed in pixels.
- **heapSize** (*integer*) –
The device's heap size, expressed in bytes.
- **memory** (*integer*) –
The device's total memory size, expressed in bytes.
- **image** (*string*) –
The device's image name.
- **carrier** (*string*) –
The device's carrier.
- **radio** (*string*) –
The device's radio.
- **compatible** (*boolean*) –
Whether the result was compatible with the device pool.
- **incompatibilityMessages** (*list*) –
Information about the compatibility.
- (*dict*) –
Represents information about incompatibility.
- **message** (*string*) –
A message about the incompatibility.
- **type** (*string*) –
The type of incompatibility.
Allowed values include:
 - ARN: The ARN.
 - FORM_FACTOR: The form factor (for example, phone or tablet).
 - MANUFACTURER: The manufacturer.
 - PLATFORM: The platform (for example, Android or iOS).

get_job (***kwargs*)
Gets information about a job.

Request Syntax

```
response = client.get_job(
    arn='string'
)
```

Parameters `arn` (*string*) – [REQUIRED]

The job's ARN.

Return type dict

Returns

Response Syntax

```
{
    'job': {
        'arn': 'string',
        'name': 'string',
        'type': 'BUILTIN_FUZZ'|'BUILTIN_EXPLORER'|'APPIUM_JAVA_JUNIT'|'APPIUM_JAVA',
        'created': datetime(2015, 1, 1),
        'status': 'PENDING'|'PROCESSING'|'SCHEDULING'|'RUNNING'|'COMPLETED',
        'result': 'PENDING'|'PASSED'|'WARNED'|'FAILED'|'SKIPPED'|'ERRORED'|'STOPPED',
        'started': datetime(2015, 1, 1),
        'stopped': datetime(2015, 1, 1),
        'counters': {
            'total': 123,
            'passed': 123,
            'failed': 123,
            'warned': 123,
            'errored': 123,
            'stopped': 123,
            'skipped': 123
        },
        'message': 'string',
        'device': {
            'arn': 'string',
            'name': 'string',
            'manufacturer': 'string',
            'model': 'string',
            'formFactor': 'PHONE'|'TABLET',
            'platform': 'ANDROID'|'IOS',
            'os': 'string',
            'cpu': {
                'frequency': 'string',
                'architecture': 'string',
                'clock': 123.0
            },
            'resolution': {
                'width': 123,
                'height': 123
            },
            'heapSize': 123,
            'memory': 123,
            'image': 'string',
            'carrier': 'string',
            'radio': 'string'
        }
    }
}
```

Response Structure

- (*dict*) –

Represents the result of a get job request.

- **job** (*dict*) –

Represents a device.

- * **arn** (*string*) –

The job's ARN.

- * **name** (*string*) –

The job's name.

- * **type** (*string*) –

The job's type.

Allowed values include the following:

- BUILTIN_FUZZ: The built-in fuzz type.
- BUILTIN_EXPLORER: For Android, an app explorer that will traverse an Android app, interacting with it and capturing screenshots at the same time.
- APPIUM_JAVA_JUNIT: The Appium Java JUnit type.
- APPIUM_JAVA_TESTNG: The Appium Java TestNG type.
- CALABASH: The Calabash type.
- INSTRUMENTATION: The Instrumentation type.
- UIAUTOMATION: The uiautomation type.
- UIAUTOMATOR: The uiautomator type.
- XCTEST: The XCode test type.

- * **created** (*datetime*) –

When the job was created.

- * **status** (*string*) –

The job's status.

Allowed values include:

- COMPLETED: A completed status.
- PENDING: A pending status.
- PROCESSING: A processing status.
- RUNNING: A running status.
- SCHEDULING: A scheduling status.

- * **result** (*string*) –

The job's result.

Allowed values include:

- ERRORED: An error condition.
- FAILED: A failed condition.
- SKIPPED: A skipped condition.
- STOPPED: A stopped condition.
- PASSED: A passing condition.
- PENDING: A pending condition.
- WARNED: A warning condition.

- * **started** (*datetime*) –

The job's start time.

- * **stopped** (*datetime*) –

The job's stop time.

* **counters** (*dict*) –

The job's result counters.

· **total** (*integer*) –

The total number of entities.

· **passed** (*integer*) –

The number of passed entities.

· **failed** (*integer*) –

The number of failed entities.

· **warned** (*integer*) –

The number of warned entities.

· **errored** (*integer*) –

The number of errored entities.

· **stopped** (*integer*) –

The number of stopped entities.

· **skipped** (*integer*) –

The number of skipped entities.

* **message** (*string*) –

A message about the job's result.

* **device** (*dict*) –

Represents a device type that an app is tested against.

· **arn** (*string*) –

The device's ARN.

· **name** (*string*) –

The device's display name.

· **manufacturer** (*string*) –

The device's manufacturer name.

· **model** (*string*) –

The device's model name.

· **formFactor** (*string*) –

The device's form factor.

Allowed values include:

· **PHONE**: The phone form factor.

· **TABLET**: The tablet form factor.

· **platform** (*string*) –

The device's platform.

Allowed values include:

· **ANDROID**: The Android platform.

· **IOS**: The iOS platform.

· **os** (*string*) –

The device's operating system type.

· **cpu** (*dict*) –

Information about the device's CPU.

- **frequency** (*string*) –
The CPU's frequency.
- **architecture** (*string*) –
The CPU's architecture, for example x86 or ARM.
- **clock** (*float*) –
The clock speed of the device's CPU, expressed in hertz (Hz).
For example, a 1.2 GHz CPU is expressed as 1200000000.
- **resolution** (*dict*) –
Represents the screen resolution of a device in height and width, expressed in pixels.
- **width** (*integer*) –
The screen resolution's width, expressed in pixels.
- **height** (*integer*) –
The screen resolution's height, expressed in pixels.
- **heapSize** (*integer*) –
The device's heap size, expressed in bytes.
- **memory** (*integer*) –
The device's total memory size, expressed in bytes.
- **image** (*string*) –
The device's image name.
- **carrier** (*string*) –
The device's carrier.
- **radio** (*string*) –
The device's radio.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_project (***kwargs*)

Gets information about a project.

Request Syntax

```
response = client.get_project(  
    arn='string'  
)
```

Parameters *arn* (*string*) – **[REQUIRED]**

The project's ARN.

Return type `dict`

Returns

Response Syntax

```
{
  'project': {
    'arn': 'string',
    'name': 'string',
    'created': datetime(2015, 1, 1)
  }
}
```

Response Structure

- (*dict*) –

Represents the result of a get project request.

- **project** (*dict*) –

Represents an operating-system neutral workspace for running and managing tests.

- * **arn** (*string*) –

The project's ARN.

- * **name** (*string*) –

The project's name.

- * **created** (*datetime*) –

When the project was created.

get_run (***kwargs*)

Gets information about a run.

Request Syntax

```
response = client.get_run(
    arn='string'
)
```

Parameters **arn** (*string*) – **[REQUIRED]**

The run's ARN.

Return type dict

Returns

Response Syntax

```
{
  'run': {
    'arn': 'string',
    'name': 'string',
    'type': 'BUILTIN_FUZZ' | 'BUILTIN_EXPLORER' | 'APPIUM_JAVA_JUNIT' | 'APPIUM_JAVA',
    'platform': 'ANDROID' | 'IOS',
    'created': datetime(2015, 1, 1),
    'status': 'PENDING' | 'PROCESSING' | 'SCHEDULING' | 'RUNNING' | 'COMPLETED',
    'result': 'PENDING' | 'PASSED' | 'WARNED' | 'FAILED' | 'SKIPPED' | 'ERRORED' | 'STOPPED',
    'started': datetime(2015, 1, 1),
    'stopped': datetime(2015, 1, 1),
    'counters': {
      'total': 123,
      'passed': 123,
    }
  }
}
```

```
        'failed': 123,
        'warned': 123,
        'errored': 123,
        'stopped': 123,
        'skipped': 123
    },
    'message': 'string',
    'totalJobs': 123,
    'completedJobs': 123,
    'billingMethod': 'METERED' | 'UNMETERED'
}
}
```

Response Structure

- *(dict)* –

Represents the result of a get run request.

- **run** (*dict*) –

Represents an app on a set of devices with a specific test and configuration.

- * **arn** (*string*) –

The run's ARN.

- * **name** (*string*) –

The run's name.

- * **type** (*string*) –

The run's type.

Must be one of the following values:

- BUILTIN_FUZZ: The built-in fuzz type.
- BUILTIN_EXPLORER: For Android, an app explorer that will traverse an Android app, interacting with it and capturing screenshots at the same time.
- APPIUM_JAVA_JUNIT: The Appium Java JUnit type.
- APPIUM_JAVA_TESTNG: The Appium Java TestNG type.
- CALABASH: The Calabash type.
- INSTRUMENTATION: The Instrumentation type.
- UIAUTOMATION: The uiautomation type.
- UIAUTOMATOR: The uiautomator type.
- XCTEST: The XCode test type.

- * **platform** (*string*) –

The run's platform.

Allowed values include:

- ANDROID: The Android platform.
- IOS: The iOS platform.

- * **created** (*datetime*) –

When the run was created.

- * **status** (*string*) –

The run's status.

Allowed values include:

- COMPLETED: A completed status.
- PENDING: A pending status.
- PROCESSING: A processing status.

- **RUNNING**: A running status.
- **SCHEDULING**: A scheduling status.
- * **result** (*string*) –
The run's result.
Allowed values include:
 - **ERRORED**: An error condition.
 - **FAILED**: A failed condition.
 - **SKIPPED**: A skipped condition.
 - **STOPPED**: A stopped condition.
 - **PASSED**: A passing condition.
 - **PENDING**: A pending condition.
 - **WARNED**: A warning condition.
- * **started** (*datetime*) –
The run's start time.
- * **stopped** (*datetime*) –
The run's stop time.
- * **counters** (*dict*) –
The run's result counters.
 - **total** (*integer*) –
The total number of entities.
 - **passed** (*integer*) –
The number of passed entities.
 - **failed** (*integer*) –
The number of failed entities.
 - **warned** (*integer*) –
The number of warned entities.
 - **errored** (*integer*) –
The number of errored entities.
 - **stopped** (*integer*) –
The number of stopped entities.
 - **skipped** (*integer*) –
The number of skipped entities.
- * **message** (*string*) –
A message about the run's result.
- * **totalJobs** (*integer*) –
The total number of jobs for the run.
- * **completedJobs** (*integer*) –
The total number of completed jobs.
- * **billingMethod** (*string*) –
Specifies the billing method for a test run: `metered` or `unmetered`. If the parameter is not specified, the default value is `unmetered`.

get_suite (***kwargs*)
Gets information about a suite.

Request Syntax

```
response = client.get_suite(
    arn='string'
)
```

Parameters **arn** (*string*) – **[REQUIRED]**

The suite's ARN.

Return type dict

Returns

Response Syntax

```
{
    'suite': {
        'arn': 'string',
        'name': 'string',
        'type': 'BUILTIN_FUZZ' | 'BUILTIN_EXPLORER' | 'APPIUM_JAVA_JUNIT' | 'APPIUM_JAVA_TESTNG',
        'created': datetime(2015, 1, 1),
        'status': 'PENDING' | 'PROCESSING' | 'SCHEDULING' | 'RUNNING' | 'COMPLETED',
        'result': 'PENDING' | 'PASSED' | 'WARNED' | 'FAILED' | 'SKIPPED' | 'ERRORED' | 'STOPPED',
        'started': datetime(2015, 1, 1),
        'stopped': datetime(2015, 1, 1),
        'counters': {
            'total': 123,
            'passed': 123,
            'failed': 123,
            'warned': 123,
            'errored': 123,
            'stopped': 123,
            'skipped': 123
        },
        'message': 'string'
    }
}
```

Response Structure

- (*dict*) –

Represents the result of a get suite request.

- **suite** (*dict*) –

Represents a collection of one or more tests.

- * **arn** (*string*) –

The suite's ARN.

- * **name** (*string*) –

The suite's name.

- * **type** (*string*) –

The suite's type.

Must be one of the following values:

- BUILTIN_FUZZ: The built-in fuzz type.
- BUILTIN_EXPLORER: For Android, an app explorer that will traverse an Android app, interacting with it and capturing screenshots at the same time.
- APPIUM_JAVA_JUNIT: The Appium Java JUnit type.
- APPIUM_JAVA_TESTNG: The Appium Java TestNG type.

- CALABASH: The Calabash type.
- INSTRUMENTATION: The Instrumentation type.
- UIAUTOMATION: The uiautomation type.
- UIAUTOMATOR: The uiautomator type.
- XCTEST: The XCode test type.

* **created** (*datetime*) –

When the suite was created.

* **status** (*string*) –

The suite's status.

Allowed values include:

- COMPLETED: A completed status.
- PENDING: A pending status.
- PROCESSING: A processing status.
- RUNNING: A running status.
- SCHEDULING: A scheduling status.

* **result** (*string*) –

The suite's result.

Allowed values include:

- ERRORED: An error condition.
- FAILED: A failed condition.
- SKIPPED: A skipped condition.
- STOPPED: A stopped condition.
- PASSED: A passing condition.
- PENDING: A pending condition.
- WARNED: A warning condition.

* **started** (*datetime*) –

The suite's start time.

* **stopped** (*datetime*) –

The suite's stop time.

* **counters** (*dict*) –

The suite's result counters.

· **total** (*integer*) –

The total number of entities.

· **passed** (*integer*) –

The number of passed entities.

· **failed** (*integer*) –

The number of failed entities.

· **warned** (*integer*) –

The number of warned entities.

· **errored** (*integer*) –

The number of errored entities.

· **stopped** (*integer*) –

The number of stopped entities.

· **skipped** (*integer*) –

The number of skipped entities.

- * **message** (*string*) –
A message about the suite's result.

get_test (***kwargs*)
Gets information about a test.

Request Syntax

```
response = client.get_test(  
    arn='string'  
)
```

Parameters **arn** (*string*) – [REQUIRED]

The test's ARN.

Return type dict

Returns

Response Syntax

```
{  
    'test': {  
        'arn': 'string',  
        'name': 'string',  
        'type': 'BUILTIN_FUZZ' | 'BUILTIN_EXPLORER' | 'APPIUM_JAVA_JUNIT' | 'APPIUM_JAVA',  
        'created': datetime(2015, 1, 1),  
        'status': 'PENDING' | 'PROCESSING' | 'SCHEDULING' | 'RUNNING' | 'COMPLETED',  
        'result': 'PENDING' | 'PASSED' | 'WARNED' | 'FAILED' | 'SKIPPED' | 'ERRORED' | 'STOPPED',  
        'started': datetime(2015, 1, 1),  
        'stopped': datetime(2015, 1, 1),  
        'counters': {  
            'total': 123,  
            'passed': 123,  
            'failed': 123,  
            'warned': 123,  
            'errored': 123,  
            'stopped': 123,  
            'skipped': 123  
        },  
        'message': 'string'  
    },  
}
```

Response Structure

- (*dict*) –

Represents the result of a get test request.

- **test** (*dict*) –

Represents a condition that is evaluated.

- * **arn** (*string*) –

The test's ARN.

- * **name** (*string*) –

The test's name.

- * **type** (*string*) –

The test's type.

Must be one of the following values:

- BUILTIN_FUZZ: The built-in fuzz type.
- BUILTIN_EXPLORER: For Android, an app explorer that will traverse an Android app, interacting with it and capturing screenshots at the same time.
- APPIUM_JAVA_JUNIT: The Appium Java JUnit type.
- APPIUM_JAVA_TESTNG: The Appium Java TestNG type.
- CALABASH: The Calabash type.
- INSTRUMENTATION: The Instrumentation type.
- UIAUTOMATION: The uiautomation type.
- UIAUTOMATOR: The uiautomator type.
- XCTEST: The XCode test type.

* **created** (*datetime*) –

When the test was created.

* **status** (*string*) –

The test's status.

Allowed values include:

- COMPLETED: A completed status.
- PENDING: A pending status.
- PROCESSING: A processing status.
- RUNNING: A running status.
- SCHEDULING: A scheduling status.

* **result** (*string*) –

The test's result.

Allowed values include:

- ERRORED: An error condition.
- FAILED: A failed condition.
- SKIPPED: A skipped condition.
- STOPPED: A stopped condition.
- PASSED: A passing condition.
- PENDING: A pending condition.
- WARNED: A warning condition.

* **started** (*datetime*) –

The test's start time.

* **stopped** (*datetime*) –

The test's stop time.

* **counters** (*dict*) –

The test's result counters.

· **total** (*integer*) –

The total number of entities.

· **passed** (*integer*) –

The number of passed entities.

· **failed** (*integer*) –

The number of failed entities.

· **warned** (*integer*) –

The number of warned entities.

· **errored** (*integer*) –

The number of errored entities.

- **stopped** (*integer*) –
The number of stopped entities.
- **skipped** (*integer*) –
The number of skipped entities.
- * **message** (*string*) –
A message about the test's result.

get_upload (**kwargs)

Gets information about an upload.

Request Syntax

```
response = client.get_upload(  
    arn='string'  
)
```

Parameters **arn** (*string*) – [REQUIRED]

The upload's ARN.

Return type dict

Returns

Response Syntax

```
{  
    'upload': {  
        'arn': 'string',  
        'name': 'string',  
        'created': datetime(2015, 1, 1),  
        'type': 'ANDROID_APP'|'IOS_APP'|'EXTERNAL_DATA'|'APPIUM_JAVA_JUNIT_TEST_PA',  
        'status': 'INITIALIZED'|'PROCESSING'|'SUCCEEDED'|'FAILED',  
        'url': 'string',  
        'metadata': 'string',  
        'contentType': 'string',  
        'message': 'string'  
    }  
}
```

Response Structure

- (*dict*) –

Represents the result of a get upload request.

- **upload** (*dict*) –

An app or a set of one or more tests to upload or that have been uploaded.

- * **arn** (*string*) –

The upload's ARN.

- * **name** (*string*) –

The upload's file name.

- * **created** (*datetime*) –

When the upload was created.

- * **type** (*string*) –

The upload's type.

Must be one of the following values:

- **ANDROID_APP**: An Android upload.
- **IOS_APP**: An iOS upload.
- **EXTERNAL_DATA**: An external data upload.
- **APPIUM_JAVA_JUNIT_TEST_PACKAGE**: An Appium Java JUnit test package upload.
- **APPIUM_JAVA_TESTNG_TEST_PACKAGE**: An Appium Java TestNG test package upload.
- **CALABASH_TEST_PACKAGE**: A Calabash test package upload.
- **INSTRUMENTATION_TEST_PACKAGE**: An instrumentation upload.
- **UIAUTOMATOR_TEST_PACKAGE**: A uiautomator test package upload.
- **XCTEST_TEST_PACKAGE**: An XCode test package upload.

* **status** (*string*) –

The upload’s status.

Must be one of the following values:

- **FAILED**: A failed status.
- **INITIALIZED**: An initialized status.
- **PROCESSING**: A processing status.
- **SUCCEEDED**: A succeeded status.

* **url** (*string*) –

The pre-signed Amazon S3 URL that was used to store a file through a corresponding PUT request.

* **metadata** (*string*) –

The upload’s metadata. For example, for Android, this contains information that is parsed from the manifest and is displayed in the AWS Device Farm console after the associated app is uploaded.

* **contentType** (*string*) –

The upload’s content type (for example, “application/octet-stream”).

* **message** (*string*) –

A message about the upload’s result.

get_waiter (*waiter_name*)

list_artifacts (**kwargs)

Gets information about artifacts.

Request Syntax

```
response = client.list_artifacts(
    arn='string',
    type='SCREENSHOT'|'FILE'|'LOG',
    nextToken='string'
)
```

Parameters

- **arn** (*string*) – [REQUIRED]

The Run, Job, Suite, or Test ARN.

- **type** (*string*) – [REQUIRED]

The artifacts' type.

Allowed values include:

- FILE: The artifacts are files.
 - LOG: The artifacts are logs.
 - SCREENSHOT: The artifacts are screenshots.
- **nextToken** (*string*) – An identifier that was returned from the previous call to this operation, which can be used to return the next set of items in the list.

Return type dict

Returns

Response Syntax

```
{
  'artifacts': [
    {
      'arn': 'string',
      'name': 'string',
      'type': 'UNKNOWN' | 'SCREENSHOT' | 'DEVICE_LOG' | 'MESSAGE_LOG' | 'RESULT_LOG',
      'extension': 'string',
      'url': 'string'
    },
  ],
  'nextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents the result of a list artifacts operation.

- **artifacts** (*list*) –

Information about the artifacts.

- * (*dict*) –

Represents the output of a test. Examples of artifacts include logs and screenshots.

- **arn** (*string*) –

The artifact's ARN.

- **name** (*string*) –

The artifact's name.

- **type** (*string*) –

The artifact's type.

Allowed values include the following:

- APPIUM_JAVA_OUTPUT: The Appium Java output type.
- APPIUM_JAVA_XML_OUTPUT: The Appium Java XML output type.
- APPIUM_SERVER_OUTPUT: The Appium server output type.
- AUTOMATION_OUTPUT: The automation output type.
- CALABASH_JSON_OUTPUT: The Calabash JSON output type.
- CALABASH_JAVA_XML_OUTPUT: The Calabash Java XML output type.

- CALABASH_PRETTY_OUTPUT: The Calabash pretty output type.
- CALABASH_STANDARD_OUTPUT: The Calabash standard output type.
- DEVICE_LOG: The device log type.
- EXERCISER_MONKEY_OUTPUT: For Android, the artifact (log) generated by an Android fuzz test.
- INSTRUMENTATION_OUTPUT: The instrumentation type.
- MESSAGE_LOG: The message log type.
- RESULT_LOG: The result log type.
- SCREENSHOT: The screenshot type.
- SERVICE_LOG: The service log type.
- UNKNOWN: An unknown type.
- **extension** (*string*) –
The artifact's file extension.
- **url** (*string*) –
The pre-signed Amazon S3 URL that can be used with a corresponding GET request to download the artifact's file.

– **nextToken** (*string*) –

If the number of items that are returned is significantly large, this is an identifier that is also returned, which can be used in a subsequent call to this operation to return the next set of items in the list.

list_device_pools (***kwargs*)

Gets information about device pools.

Request Syntax

```
response = client.list_device_pools(
    arn='string',
    type='CURATED' | 'PRIVATE',
    nextToken='string'
)
```

Parameters

- **arn** (*string*) – [REQUIRED]

The project ARN.

- **type** (*string*) – The device pools' type.

Allowed values include:

- CURATED: A device pool that is created and managed by AWS Device Farm.
- PRIVATE: A device pool that is created and managed by the device pool developer.

- **nextToken** (*string*) – An identifier that was returned from the previous call to this operation, which can be used to return the next set of items in the list.

Return type dict

Returns

Response Syntax

```
{
    'devicePools': [
        {
```

```
    'arn': 'string',
    'name': 'string',
    'description': 'string',
    'type': 'CURATED'|'PRIVATE',
    'rules': [
        {
            'attribute': 'ARN'|'PLATFORM'|'FORM_FACTOR'|'MANUFACTURER',
            'operator': 'EQUALS'|'LESS_THAN'|'GREATER_THAN'|'IN'|'NOT_IN',
            'value': 'string'
        },
    ]
},
],
'nextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents the result of a list device pools request.

- **devicePools** (*list*) –

Information about the device pools.

- * (*dict*) –

Represents a collection of device types.

- **arn** (*string*) –

The device pool's ARN.

- **name** (*string*) –

The device pool's name.

- **description** (*string*) –

The device pool's description.

- **type** (*string*) –

The device pool's type.

Allowed values include:

- **CURATED**: A device pool that is created and managed by AWS Device Farm.
- **PRIVATE**: A device pool that is created and managed by the device pool developer.
- **rules** (*list*) –

Information about the device pool's rules.

- (*dict*) –

Represents a condition for a device pool.

- **attribute** (*string*) –

The rule's attribute.

Allowed values include:

- **ARN**: The ARN.
- **FORM_FACTOR**: The form factor (for example, phone or tablet).
- **MANUFACTURER**: The manufacturer.
- **PLATFORM**: The platform (for example, Android or iOS).

- **operator** (*string*) –
The rule's operator.
 - **EQUALS**: The equals operator.
 - **GREATER_THAN**: The greater-than operator.
 - **IN**: The in operator.
 - **LESS_THAN**: The less-than operator.
 - **NOT_IN**: The not-in operator.
 - **value** (*string*) –
The rule's value.
- **nextToken** (*string*) –

If the number of items that are returned is significantly large, this is an identifier that is also returned, which can be used in a subsequent call to this operation to return the next set of items in the list.

list_devices (***kwargs*)

Gets information about unique device types.

Request Syntax

```
response = client.list_devices(
    arn='string',
    nextToken='string'
)
```

Parameters

- **arn** (*string*) – The device types' ARNs.
- **nextToken** (*string*) – An identifier that was returned from the previous call to this operation, which can be used to return the next set of items in the list.

Return type dict

Returns

Response Syntax

```
{
  'devices': [
    {
      'arn': 'string',
      'name': 'string',
      'manufacturer': 'string',
      'model': 'string',
      'formFactor': 'PHONE'|'TABLET',
      'platform': 'ANDROID'|'IOS',
      'os': 'string',
      'cpu': {
        'frequency': 'string',
        'architecture': 'string',
        'clock': 123.0
      },
      'resolution': {
        'width': 123,
        'height': 123
      },
      'heapSize': 123,
      'memory': 123,
      'image': 'string',
      'carrier': 'string',
```

```
        'radio': 'string'
    },
],
'nextToken': 'string'
}
```

Response Structure

- *(dict)* –

Represents the result of a list devices operation.

- **devices** (*list*) –

Information about the devices.

- * *(dict)* –

Represents a device type that an app is tested against.

- **arn** (*string*) –

The device's ARN.

- **name** (*string*) –

The device's display name.

- **manufacturer** (*string*) –

The device's manufacturer name.

- **model** (*string*) –

The device's model name.

- **formFactor** (*string*) –

The device's form factor.

Allowed values include:

- PHONE: The phone form factor.

- TABLET: The tablet form factor.

- **platform** (*string*) –

The device's platform.

Allowed values include:

- ANDROID: The Android platform.

- IOS: The iOS platform.

- **os** (*string*) –

The device's operating system type.

- **cpu** (*dict*) –

Information about the device's CPU.

- **frequency** (*string*) –

The CPU's frequency.

- **architecture** (*string*) –

The CPU's architecture, for example x86 or ARM.

- **clock** (*float*) –

The clock speed of the device's CPU, expressed in hertz (Hz).

For example, a 1.2 GHz CPU is expressed as 1200000000.

- **resolution** (*dict*) –

Represents the screen resolution of a device in height and width, expressed in pixels.

- **width** (*integer*) –
The screen resolution's width, expressed in pixels.
 - **height** (*integer*) –
The screen resolution's height, expressed in pixels.
 - **heapSize** (*integer*) –
The device's heap size, expressed in bytes.
 - **memory** (*integer*) –
The device's total memory size, expressed in bytes.
 - **image** (*string*) –
The device's image name.
 - **carrier** (*string*) –
The device's carrier.
 - **radio** (*string*) –
The device's radio.
- **nextToken** (*string*) –
- If the number of items that are returned is significantly large, this is an identifier that is also returned, which can be used in a subsequent call to this operation to return the next set of items in the list.

list_jobs (***kwargs*)
Gets information about jobs.

Request Syntax

```
response = client.list_jobs(
    arn='string',
    nextToken='string'
)
```

Parameters

- **arn** (*string*) – [REQUIRED]
The jobs' ARNs.
- **nextToken** (*string*) – An identifier that was returned from the previous call to this operation, which can be used to return the next set of items in the list.

Return type dict

Returns

Response Syntax

```
{
    'jobs': [
        {
            'arn': 'string',
            'name': 'string',
            'type': 'BUILTIN_FUZZ' | 'BUILTIN_EXPLORER' | 'APPIUM_JAVA_JUNIT' | 'APPIUM_
            'created': datetime(2015, 1, 1),
            'status': 'PENDING' | 'PROCESSING' | 'SCHEDULING' | 'RUNNING' | 'COMPLETED',
            'result': 'PENDING' | 'PASSED' | 'WARNED' | 'FAILED' | 'SKIPPED' | 'ERRORED' | 'ST
            'started': datetime(2015, 1, 1),
            'stopped': datetime(2015, 1, 1),
            'counters': {
                'total': 123,
```

```
        'passed': 123,
        'failed': 123,
        'warned': 123,
        'errored': 123,
        'stopped': 123,
        'skipped': 123
    },
    'message': 'string',
    'device': {
        'arn': 'string',
        'name': 'string',
        'manufacturer': 'string',
        'model': 'string',
        'formFactor': 'PHONE'|'TABLET',
        'platform': 'ANDROID'|'IOS',
        'os': 'string',
        'cpu': {
            'frequency': 'string',
            'architecture': 'string',
            'clock': 123.0
        },
        'resolution': {
            'width': 123,
            'height': 123
        },
        'heapSize': 123,
        'memory': 123,
        'image': 'string',
        'carrier': 'string',
        'radio': 'string'
    },
    },
    'nextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents the result of a list jobs request.

- **jobs** (*list*) –

Information about the jobs.

- * (*dict*) –

Represents a device.

- **arn** (*string*) –

The job's ARN.

- **name** (*string*) –

The job's name.

- **type** (*string*) –

The job's type.

Allowed values include the following:

- **BUILTIN_FUZZ**: The built-in fuzz type.
- **BUILTIN_EXPLORER**: For Android, an app explorer that will traverse an Android app, interacting with it and captur-

ing screenshots at the same time.

- **APPIUM_JAVA_JUNIT**: The Appium Java JUnit type.
- **APPIUM_JAVA_TESTNG**: The Appium Java TestNG type.
- **CALABASH**: The Calabash type.
- **INSTRUMENTATION**: The Instrumentation type.
- **UIAUTOMATION**: The uiautomation type.
- **UIAUTOMATOR**: The uiautomator type.
- **XCTEST**: The XCode test type.

- **created** (*datetime*) –

When the job was created.

- **status** (*string*) –

The job's status.

Allowed values include:

- **COMPLETED**: A completed status.
- **PENDING**: A pending status.
- **PROCESSING**: A processing status.
- **RUNNING**: A running status.
- **SCHEDULING**: A scheduling status.

- **result** (*string*) –

The job's result.

Allowed values include:

- **ERRORED**: An error condition.
- **FAILED**: A failed condition.
- **SKIPPED**: A skipped condition.
- **STOPPED**: A stopped condition.
- **PASSED**: A passing condition.
- **PENDING**: A pending condition.
- **WARNED**: A warning condition.

- **started** (*datetime*) –

The job's start time.

- **stopped** (*datetime*) –

The job's stop time.

- **counters** (*dict*) –

The job's result counters.

- **total** (*integer*) –

The total number of entities.

- **passed** (*integer*) –

The number of passed entities.

- **failed** (*integer*) –

The number of failed entities.

- **warned** (*integer*) –

The number of warned entities.

- **errored** (*integer*) –

The number of errored entities.

- **stopped** (*integer*) –

The number of stopped entities.

- **skipped** (*integer*) –
The number of skipped entities.
- **message** (*string*) –
A message about the job's result.
- **device** (*dict*) –
Represents a device type that an app is tested against.
- **arn** (*string*) –
The device's ARN.
- **name** (*string*) –
The device's display name.
- **manufacturer** (*string*) –
The device's manufacturer name.
- **model** (*string*) –
The device's model name.
- **formFactor** (*string*) –
The device's form factor.

Allowed values include:
 - PHONE: The phone form factor.
 - TABLET: The tablet form factor.
- **platform** (*string*) –
The device's platform.

Allowed values include:
 - ANDROID: The Android platform.
 - IOS: The iOS platform.
- **os** (*string*) –
The device's operating system type.
- **cpu** (*dict*) –
Information about the device's CPU.
- **frequency** (*string*) –
The CPU's frequency.
- **architecture** (*string*) –
The CPU's architecture, for example x86 or ARM.
- **clock** (*float*) –
The clock speed of the device's CPU, expressed in hertz (Hz).
For example, a 1.2 GHz CPU is expressed as 1200000000.
- **resolution** (*dict*) –
Represents the screen resolution of a device in height and width, expressed in pixels.
- **width** (*integer*) –
The screen resolution's width, expressed in pixels.
- **height** (*integer*) –
The screen resolution's height, expressed in pixels.
- **heapSize** (*integer*) –
The device's heap size, expressed in bytes.

- **memory** (*integer*) –
The device’s total memory size, expressed in bytes.
- **image** (*string*) –
The device’s image name.
- **carrier** (*string*) –
The device’s carrier.
- **radio** (*string*) –
The device’s radio.
- **nextToken** (*string*) –
If the number of items that are returned is significantly large, this is an identifier that is also returned, which can be used in a subsequent call to this operation to return the next set of items in the list.

list_projects (***kwargs*)

Gets information about projects.

Request Syntax

```
response = client.list_projects(
    arn='string',
    nextToken='string'
)
```

Parameters

- **arn** (*string*) – The projects’ ARNs.
- **nextToken** (*string*) – An identifier that was returned from the previous call to this operation, which can be used to return the next set of items in the list.

Return type dict

Returns

Response Syntax

```
{
    'projects': [
        {
            'arn': 'string',
            'name': 'string',
            'created': datetime(2015, 1, 1)
        },
    ],
    'nextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents the result of a list projects request.

- **projects** (*list*) –

Information about the projects.

- * (*dict*) –

Represents an operating-system neutral workspace for running and managing tests.

- **arn** (*string*) –
The project's ARN.
- **name** (*string*) –
The project's name.
- **created** (*datetime*) –
When the project was created.
- **nextToken** (*string*) –

If the number of items that are returned is significantly large, this is an identifier that is also returned, which can be used in a subsequent call to this operation to return the next set of items in the list.

list_runs (***kwargs*)

Gets information about runs.

Request Syntax

```
response = client.list_runs(  
    arn='string',  
    nextToken='string'  
)
```

Parameters

- **arn** (*string*) – **[REQUIRED]**
The runs' ARNs.
- **nextToken** (*string*) – An identifier that was returned from the previous call to this operation, which can be used to return the next set of items in the list.

Return type dict

Returns

Response Syntax

```
{  
    'runs': [  
        {  
            'arn': 'string',  
            'name': 'string',  
            'type': 'BUILTIN_FUZZ' | 'BUILTIN_EXPLORER' | 'APPIUM_JAVA_JUNIT' | 'APPIUM_...  
            'platform': 'ANDROID' | 'IOS',  
            'created': datetime(2015, 1, 1),  
            'status': 'PENDING' | 'PROCESSING' | 'SCHEDULING' | 'RUNNING' | 'COMPLETED',  
            'result': 'PENDING' | 'PASSED' | 'WARNED' | 'FAILED' | 'SKIPPED' | 'ERRORED' | 'ST...  
            'started': datetime(2015, 1, 1),  
            'stopped': datetime(2015, 1, 1),  
            'counters': {  
                'total': 123,  
                'passed': 123,  
                'failed': 123,  
                'warned': 123,  
                'errored': 123,  
                'stopped': 123,  
                'skipped': 123  
            },  
            'message': 'string',  
            'totalJobs': 123,  
            'completedJobs': 123,  
        },  
    ],  
}
```



```

        'billingMethod': 'METERED' | 'UNMETERED'
    },
    ],
    'nextToken': 'string'
}

```

Response Structure

- *(dict)* –

Represents the result of a list runs request.

- **runs** (*list*) –

Information about the runs.

- * *(dict)* –

Represents an app on a set of devices with a specific test and configuration.

- **arn** (*string*) –

The run's ARN.

- **name** (*string*) –

The run's name.

- **type** (*string*) –

The run's type.

Must be one of the following values:

- BUILTIN_FUZZ: The built-in fuzz type.
- BUILTIN_EXPLORER: For Android, an app explorer that will traverse an Android app, interacting with it and capturing screenshots at the same time.
- APPIUM_JAVA_JUNIT: The Appium Java JUnit type.
- APPIUM_JAVA_TESTNG: The Appium Java TestNG type.
- CALABASH: The Calabash type.
- INSTRUMENTATION: The Instrumentation type.
- UIAUTOMATION: The uiautomation type.
- UIAUTOMATOR: The uiautomator type.
- XCTEST: The XCode test type.
- **platform** (*string*) –

The run's platform.

Allowed values include:

- ANDROID: The Android platform.
- IOS: The iOS platform.
- **created** (*datetime*) –

When the run was created.

- **status** (*string*) –

The run's status.

Allowed values include:

- COMPLETED: A completed status.
- PENDING: A pending status.
- PROCESSING: A processing status.
- RUNNING: A running status.
- SCHEDULING: A scheduling status.

- **result** (*string*) –

The run's result.

Allowed values include:

- **ERRORED**: An error condition.
- **FAILED**: A failed condition.
- **SKIPPED**: A skipped condition.
- **STOPPED**: A stopped condition.
- **PASSED**: A passing condition.
- **PENDING**: A pending condition.
- **WARNED**: A warning condition.
- **started** (*datetime*) –

The run's start time.

- **stopped** (*datetime*) –

The run's stop time.

- **counters** (*dict*) –

The run's result counters.

- **total** (*integer*) –

The total number of entities.

- **passed** (*integer*) –

The number of passed entities.

- **failed** (*integer*) –

The number of failed entities.

- **warned** (*integer*) –

The number of warned entities.

- **errored** (*integer*) –

The number of errored entities.

- **stopped** (*integer*) –

The number of stopped entities.

- **skipped** (*integer*) –

The number of skipped entities.

- **message** (*string*) –

A message about the run's result.

- **totalJobs** (*integer*) –

The total number of jobs for the run.

- **completedJobs** (*integer*) –

The total number of completed jobs.

- **billingMethod** (*string*) –

Specifies the billing method for a test run: `metered` or `unmetered`. If the parameter is not specified, the default value is `unmetered`.

- **nextToken** (*string*) –

If the number of items that are returned is significantly large, this is an identifier that is also returned, which can be used in a subsequent call to this operation to return the next set of items in the list.

list_samples (**kwargs)
Gets information about samples.

Request Syntax

```
response = client.list_samples(
    arn='string',
    nextToken='string'
)
```

Parameters

- **arn** (*string*) – [REQUIRED]
The samples' ARNs.
- **nextToken** (*string*) – An identifier that was returned from the previous call to this operation, which can be used to return the next set of items in the list.

Return type dict

Returns

Response Syntax

```
{
  'samples': [
    {
      'arn': 'string',
      'type': 'CPU' | 'MEMORY' | 'THREADS' | 'RX_RATE' | 'TX_RATE' | 'RX' | 'TX' | 'NATIVE',
      'url': 'string'
    },
  ],
  'nextToken': 'string'
}
```

Response Structure

- (*dict*) –
Represents the result of a list samples request.
 - **samples** (*list*) –
Information about the samples.
 - * (*dict*) –
Represents a sample of performance data.
 - **arn** (*string*) –
The sample's ARN.
 - **type** (*string*) –
The sample's type.
Must be one of the following values:
 - CPU: A CPU sample type. This is expressed as the app processing CPU time (including child processes) as reported by process, as a percentage.
 - MEMORY: A memory usage sample type. This is expressed as the total proportional set size of an app process, in kilobytes.
 - NATIVE_AVG_DRAWTIME
 - NATIVE_FPS
 - NATIVE_FRAMES
 - NATIVE_MAX_DRAWTIME

- NATIVE_MIN_DRAWTIME
- OPENGL_AVG_DRAWTIME
- OPENGL_FPS
- OPENGL_FRAMES
- OPENGL_MAX_DRAWTIME
- OPENGL_MIN_DRAWTIME
- RX
- RX_RATE: The total number of bytes per second (TCP and UDP) that are sent, by app process.
- THREADS: A threads sample type. This is expressed as the total number of threads per app process.
- TX
- TX_RATE: The total number of bytes per second (TCP and UDP) that are received, by app process.
- **url** (*string*) –

The pre-signed Amazon S3 URL that can be used with a corresponding GET request to download the sample's file.

– **nextToken** (*string*) –

If the number of items that are returned is significantly large, this is an identifier that is also returned, which can be used in a subsequent call to this operation to return the next set of items in the list.

list_suites (**kwargs)

Gets information about suites.

Request Syntax

```
response = client.list_suites(
    arn='string',
    nextToken='string'
)
```

Parameters

- **arn** (*string*) – **[REQUIRED]**

The suites' ARNs.

- **nextToken** (*string*) – An identifier that was returned from the previous call to this operation, which can be used to return the next set of items in the list.

Return type dict

Returns

Response Syntax

```
{
    'suites': [
        {
            'arn': 'string',
            'name': 'string',
            'type': 'BUILTIN_FUZZ' | 'BUILTIN_EXPLORER' | 'APPIUM_JAVA_JUNIT' | 'APPIUM_...',
            'created': datetime(2015, 1, 1),
            'status': 'PENDING' | 'PROCESSING' | 'SCHEDULING' | 'RUNNING' | 'COMPLETED',
            'result': 'PENDING' | 'PASSED' | 'WARNED' | 'FAILED' | 'SKIPPED' | 'ERRORED' | 'ST...',
            'started': datetime(2015, 1, 1),
            'stopped': datetime(2015, 1, 1),
            'counters': {
                'total': 123,
```

```

        'passed': 123,
        'failed': 123,
        'warned': 123,
        'errored': 123,
        'stopped': 123,
        'skipped': 123
    },
    'message': 'string'
},
],
'nextToken': 'string'
}

```

Response Structure

- (*dict*) –

Represents the result of a list suites request.

- **suites** (*list*) –

Information about the suites.

- * (*dict*) –

Represents a collection of one or more tests.

- **arn** (*string*) –

The suite's ARN.

- **name** (*string*) –

The suite's name.

- **type** (*string*) –

The suite's type.

Must be one of the following values:

- BUILTIN_FUZZ: The built-in fuzz type.
- BUILTIN_EXPLORER: For Android, an app explorer that will traverse an Android app, interacting with it and capturing screenshots at the same time.
- APPIUM_JAVA_JUNIT: The Appium Java JUnit type.
- APPIUM_JAVA_TESTNG: The Appium Java TestNG type.
- CALABASH: The Calabash type.
- INSTRUMENTATION: The Instrumentation type.
- UIAUTOMATION: The uiautomation type.
- UIAUTOMATOR: The uiautomator type.
- XCTEST: The XCode test type.
- **created** (*datetime*) –

When the suite was created.

- **status** (*string*) –

The suite's status.

Allowed values include:

- COMPLETED: A completed status.
- PENDING: A pending status.
- PROCESSING: A processing status.
- RUNNING: A running status.
- SCHEDULING: A scheduling status.
- **result** (*string*) –

The suite's result.

Allowed values include:

- **ERRORED**: An error condition.
- **FAILED**: A failed condition.
- **SKIPPED**: A skipped condition.
- **STOPPED**: A stopped condition.
- **PASSED**: A passing condition.
- **PENDING**: A pending condition.
- **WARNED**: A warning condition.
- **started** (*datetime*) –

The suite's start time.

- **stopped** (*datetime*) –

The suite's stop time.

- **counters** (*dict*) –

The suite's result counters.

- **total** (*integer*) –

The total number of entities.

- **passed** (*integer*) –

The number of passed entities.

- **failed** (*integer*) –

The number of failed entities.

- **warned** (*integer*) –

The number of warned entities.

- **errored** (*integer*) –

The number of errored entities.

- **stopped** (*integer*) –

The number of stopped entities.

- **skipped** (*integer*) –

The number of skipped entities.

- **message** (*string*) –

A message about the suite's result.

- **nextToken** (*string*) –

If the number of items that are returned is significantly large, this is an identifier that is also returned, which can be used in a subsequent call to this operation to return the next set of items in the list.

list_tests (***kwargs*)

Gets information about tests.

Request Syntax

```
response = client.list_tests(  
    arn='string',  
    nextToken='string'  
)
```

Parameters

- **arn** (*string*) – [REQUIRED]

The tests' ARNs.

- **nextToken** (*string*) – An identifier that was returned from the previous call to this operation, which can be used to return the next set of items in the list.

Return type dict

Returns

Response Syntax

```
{
  'tests': [
    {
      'arn': 'string',
      'name': 'string',
      'type': 'BUILTIN_FUZZ' | 'BUILTIN_EXPLORER' | 'APPIUM_JAVA_JUNIT' | 'APPIUM_...',
      'created': datetime(2015, 1, 1),
      'status': 'PENDING' | 'PROCESSING' | 'SCHEDULING' | 'RUNNING' | 'COMPLETED',
      'result': 'PENDING' | 'PASSED' | 'WARNED' | 'FAILED' | 'SKIPPED' | 'ERRORED' | 'ST...',
      'started': datetime(2015, 1, 1),
      'stopped': datetime(2015, 1, 1),
      'counters': {
        'total': 123,
        'passed': 123,
        'failed': 123,
        'warned': 123,
        'errored': 123,
        'stopped': 123,
        'skipped': 123
      },
      'message': 'string'
    },
  ],
  'nextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents the result of a list tests request.

- **tests** (*list*) –

Information about the tests.

- * (*dict*) –

Represents a condition that is evaluated.

- **arn** (*string*) –

The test's ARN.

- **name** (*string*) –

The test's name.

- **type** (*string*) –

The test's type.

Must be one of the following values:

- **BUILTIN_FUZZ**: The built-in fuzz type.
- **BUILTIN_EXPLORER**: For Android, an app explorer that will traverse an Android app, interacting with it and capturing screenshots at the same time.

- **APPIUM_JAVA_JUNIT**: The Appium Java JUnit type.
- **APPIUM_JAVA_TESTNG**: The Appium Java TestNG type.
- **CALABASH**: The Calabash type.
- **INSTRUMENTATION**: The Instrumentation type.
- **UIAUTOMATION**: The uiautomation type.
- **UIAUTOMATOR**: The uiautomator type.
- **XCTEST**: The XCode test type.
- **created** (*datetime*) –
When the test was created.
- **status** (*string*) –
The test's status.

Allowed values include:
 - **COMPLETED**: A completed status.
 - **PENDING**: A pending status.
 - **PROCESSING**: A processing status.
 - **RUNNING**: A running status.
 - **SCHEDULING**: A scheduling status.
- **result** (*string*) –
The test's result.

Allowed values include:
 - **ERRORED**: An error condition.
 - **FAILED**: A failed condition.
 - **SKIPPED**: A skipped condition.
 - **STOPPED**: A stopped condition.
 - **PASSED**: A passing condition.
 - **PENDING**: A pending condition.
 - **WARNED**: A warning condition.
- **started** (*datetime*) –
The test's start time.
- **stopped** (*datetime*) –
The test's stop time.
- **counters** (*dict*) –
The test's result counters.
- **total** (*integer*) –
The total number of entities.
- **passed** (*integer*) –
The number of passed entities.
- **failed** (*integer*) –
The number of failed entities.
- **warned** (*integer*) –
The number of warned entities.
- **errored** (*integer*) –
The number of errored entities.
- **stopped** (*integer*) –
The number of stopped entities.
- **skipped** (*integer*) –
The number of skipped entities.

The number of skipped entities.

- **message** (*string*) –

A message about the test's result.

- **nextToken** (*string*) –

If the number of items that are returned is significantly large, this is an identifier that is also returned, which can be used in a subsequent call to this operation to return the next set of items in the list.

list_unique_problems (***kwargs*)

Gets information about unique problems.

Request Syntax

```
response = client.list_unique_problems(
    arn='string',
    nextToken='string'
)
```

Parameters

- **arn** (*string*) – [REQUIRED]

The unique problems' ARNs.

- **nextToken** (*string*) – An identifier that was returned from the previous call to this operation, which can be used to return the next set of items in the list.

Return type dict

Returns

Response Syntax

```
{
    'uniqueProblems': {
        'string': [
            {
                'message': 'string',
                'problems': [
                    {
                        'run': {
                            'arn': 'string',
                            'name': 'string'
                        },
                        'job': {
                            'arn': 'string',
                            'name': 'string'
                        },
                        'suite': {
                            'arn': 'string',
                            'name': 'string'
                        },
                        'test': {
                            'arn': 'string',
                            'name': 'string'
                        },
                        'device': {
                            'arn': 'string',
                            'name': 'string',
                            'manufacturer': 'string',
                            'model': 'string',
```

```
        'formFactor': 'PHONE'|'TABLET',
        'platform': 'ANDROID'|'IOS',
        'os': 'string',
        'cpu': {
            'frequency': 'string',
            'architecture': 'string',
            'clock': 123.0
        },
        'resolution': {
            'width': 123,
            'height': 123
        },
        'heapSize': 123,
        'memory': 123,
        'image': 'string',
        'carrier': 'string',
        'radio': 'string'
    },
    'result': 'PENDING'|'PASSED'|'WARNED'|'FAILED'|'SKIPPED'|'
    'message': 'string'
},
]
},
]
},
'nextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents the result of a list unique problems request.

- **uniqueProblems** (*dict*) –

Information about the unique problems.

Allowed values include:

- * **ERRORED**: An error condition.
- * **FAILED**: A failed condition.
- * **SKIPPED**: A skipped condition.
- * **STOPPED**: A stopped condition.
- * **PASSED**: A passing condition.
- * **PENDING**: A pending condition.
- * **WARNED**: A warning condition.

- * (*string*) –

- (*list*) –

- (*dict*) –

- A collection of one or more problems, grouped by their result.

- **message** (*string*) –

- A message about the unique problems' result.

- **problems** (*list*) –

- Information about the problems.

- (*dict*) –

- Represents a specific warning or failure.

- **run** (*dict*) –

Information about the associated run.

- **arn** (*string*) –

The problem detail's ARN.

- **name** (*string*) –

The problem detail's name.

- **job** (*dict*) –

Information about the associated job.

- **arn** (*string*) –

The problem detail's ARN.

- **name** (*string*) –

The problem detail's name.

- **suite** (*dict*) –

Information about the associated suite.

- **arn** (*string*) –

The problem detail's ARN.

- **name** (*string*) –

The problem detail's name.

- **test** (*dict*) –

Information about the associated test.

- **arn** (*string*) –

The problem detail's ARN.

- **name** (*string*) –

The problem detail's name.

- **device** (*dict*) –

Information about the associated device.

- **arn** (*string*) –

The device's ARN.

- **name** (*string*) –

The device's display name.

- **manufacturer** (*string*) –

The device's manufacturer name.

- **model** (*string*) –

The device's model name.

- **formFactor** (*string*) –

The device's form factor.

Allowed values include:

- PHONE: The phone form factor.
- TABLET: The tablet form factor.
- **platform** (*string*) –

The device's platform.

Allowed values include:

- ANDROID: The Android platform.
- IOS: The iOS platform.

- **os** (*string*) –
The device's operating system type.
- **cpu** (*dict*) –
Information about the device's CPU.
- **frequency** (*string*) –
The CPU's frequency.
- **architecture** (*string*) –
The CPU's architecture, for example x86 or ARM.
- **clock** (*float*) –
The clock speed of the device's CPU, expressed in hertz (Hz).
For example, a 1.2 GHz CPU is expressed as 1200000000.
- **resolution** (*dict*) –
Represents the screen resolution of a device in height and width, expressed in pixels.
- **width** (*integer*) –
The screen resolution's width, expressed in pixels.
- **height** (*integer*) –
The screen resolution's height, expressed in pixels.
- **heapSize** (*integer*) –
The device's heap size, expressed in bytes.
- **memory** (*integer*) –
The device's total memory size, expressed in bytes.
- **image** (*string*) –
The device's image name.
- **carrier** (*string*) –
The device's carrier.
- **radio** (*string*) –
The device's radio.
- **result** (*string*) –
The problem's result.

Allowed values include:
 - **ERRORED**: An error condition.
 - **FAILED**: A failed condition.
 - **SKIPPED**: A skipped condition.
 - **STOPPED**: A stopped condition.
 - **PASSED**: A passing condition.
 - **PENDING**: A pending condition.
 - **WARNED**: A warning condition.
- **message** (*string*) –
A message about the problem's result.

– **nextToken** (*string*) –

If the number of items that are returned is significantly large, this is an identifier that is also returned, which can be used in a subsequent call to this operation to return the next set of items in the list.

list_uploads (**kwargs)
Gets information about uploads.

Request Syntax

```
response = client.list_uploads(
    arn='string',
    nextToken='string'
)
```

Parameters

- **arn** (*string*) – [REQUIRED]
The uploads' ARNs.
- **nextToken** (*string*) – An identifier that was returned from the previous call to this operation, which can be used to return the next set of items in the list.

Return type dict

Returns

Response Syntax

```
{
  'uploads': [
    {
      'arn': 'string',
      'name': 'string',
      'created': datetime(2015, 1, 1),
      'type': 'ANDROID_APP' | 'IOS_APP' | 'EXTERNAL_DATA' | 'APPIUM_JAVA_JUNIT_TEST',
      'status': 'INITIALIZED' | 'PROCESSING' | 'SUCCEEDED' | 'FAILED',
      'url': 'string',
      'metadata': 'string',
      'contentType': 'string',
      'message': 'string'
    },
  ],
  'nextToken': 'string'
}
```

Response Structure

- (*dict*) –
Represents the result of a list uploads request.
 - **uploads** (*list*) –
Information about the uploads.
 - * (*dict*) –
An app or a set of one or more tests to upload or that have been uploaded.
 - **arn** (*string*) –
The upload's ARN.
 - **name** (*string*) –
The upload's file name.
 - **created** (*datetime*) –
When the upload was created.
 - **type** (*string*) –
The upload's type.

Must be one of the following values:

- **ANDROID_APP**: An Android upload.
- **IOS_APP**: An iOS upload.
- **EXTERNAL_DATA**: An external data upload.
- **APPIUM_JAVA_JUNIT_TEST_PACKAGE**: An Appium Java JUnit test package upload.
- **APPIUM_JAVA_TESTNG_TEST_PACKAGE**: An Appium Java TestNG test package upload.
- **CALABASH_TEST_PACKAGE**: A Calabash test package upload.
- **INSTRUMENTATION_TEST_PACKAGE**: An instrumentation upload.
- **UIAUTOMATOR_TEST_PACKAGE**: A uiautomator test package upload.
- **XCTEST_TEST_PACKAGE**: An XCode test package upload.
- **status** (*string*) –

The upload's status.

Must be one of the following values:

- **FAILED**: A failed status.
- **INITIALIZED**: An initialized status.
- **PROCESSING**: A processing status.
- **SUCCEEDED**: A succeeded status.
- **url** (*string*) –

The pre-signed Amazon S3 URL that was used to store a file through a corresponding PUT request.

- **metadata** (*string*) –

The upload's metadata. For example, for Android, this contains information that is parsed from the manifest and is displayed in the AWS Device Farm console after the associated app is uploaded.

- **contentType** (*string*) –

The upload's content type (for example, "application/octet-stream").

- **message** (*string*) –

A message about the upload's result.

- **nextToken** (*string*) –

If the number of items that are returned is significantly large, this is an identifier that is also returned, which can be used in a subsequent call to this operation to return the next set of items in the list.

schedule_run (***kwargs*)

Schedules a run.

Request Syntax

```
response = client.schedule_run(
    projectArn='string',
    appArn='string',
    devicePoolArn='string',
    name='string',
    test={
```

```

        'type': 'BUILTIN_FUZZ' | 'BUILTIN_EXPLORER' | 'APPIUM_JAVA_JUNIT' | 'APPIUM_JAVA_TESTNG' |
        'testPackageArn': 'string',
        'filter': 'string',
        'parameters': {
            'string': 'string'
        }
    },
    configuration={
        'extraDataPackageArn': 'string',
        'networkProfileArn': 'string',
        'locale': 'string',
        'location': {
            'latitude': 123.0,
            'longitude': 123.0
        },
        'radios': {
            'wifi': True|False,
            'bluetooth': True|False,
            'nfc': True|False,
            'gps': True|False
        },
        'auxiliaryApps': [
            'string',
        ],
        'billingMethod': 'METERED' | 'UNMETERED'
    }
)

```

Parameters

- **projectArn** (*string*) – **[REQUIRED]**

The ARN of the project for the run to be scheduled.

- **appArn** (*string*) – **[REQUIRED]**

The ARN of the app to schedule a run.

- **devicePoolArn** (*string*) – **[REQUIRED]**

The ARN of the device pool for the run to be scheduled.

- **name** (*string*) – The name for the run to be scheduled.
- **test** (*dict*) – **[REQUIRED]**

Information about the test for the run to be scheduled.

- **type** (*string*) – **[REQUIRED]**

The test's type.

Must be one of the following values:

- * BUILTIN_FUZZ: The built-in fuzz type.
- * BUILTIN_EXPLORER: For Android, an app explorer that will traverse an Android app, interacting with it and capturing screenshots at the same time.
- * APPIUM_JAVA_JUNIT: The Appium Java JUnit type.
- * APPIUM_JAVA_TESTNG: The Appium Java TestNG type.
- * CALABASH: The Calabash type.
- * INSTRUMENTATION: The Instrumentation type.
- * UIAUTOMATION: The uiautomation type.
- * UIAUTOMATOR: The uiautomator type.
- * XCTEST: The XCode test type.

- **testPackageArn** (*string*) –

The ARN of the uploaded test that will be run.

- **filter** (*string*) –

The test’s filter.

- **parameters** (*dict*) –

The test’s parameters, such as test framework parameters and fixture settings.

- * (*string*) –

- (*string*) –

- **configuration** (*dict*) – Information about the settings for the run to be scheduled.

- **extraDataPackageArn** (*string*) –

The ARN of the extra data for the run. The extra data is a .zip file that AWS Device Farm will extract to external data for Android or the app’s sandbox for iOS.

- **networkProfileArn** (*string*) –

Reserved for internal use.

- **locale** (*string*) –

Information about the locale that is used for the run.

- **location** (*dict*) –

Information about the location that is used for the run.

- * **latitude** (*float*) – [REQUIRED]

- The latitude.

- * **longitude** (*float*) – [REQUIRED]

- The longitude.

- **radios** (*dict*) –

Information about the radio states for the run.

- * **wifi** (*boolean*) –

- True if Wi-Fi is enabled at the beginning of the test; otherwise, false.

- * **bluetooth** (*boolean*) –

- True if Bluetooth is enabled at the beginning of the test; otherwise, false.

- * **nfc** (*boolean*) –

- True if NFC is enabled at the beginning of the test; otherwise, false.

- * **gps** (*boolean*) –

- True if GPS is enabled at the beginning of the test; otherwise, false.

- **auxiliaryApps** (*list*) –

A list of auxiliary apps for the run.

- * (*string*) –

- **billingMethod** (*string*) –

Specifies the billing method for a test run: `metered` or `unmetered`. If the parameter is not specified, the default value is `unmetered`.

Return type `dict`

Returns

Response Syntax


```

{
  'run': {
    'arn': 'string',
    'name': 'string',
    'type': 'BUILTIN_FUZZ'|'BUILTIN_EXPLORER'|'APPIUM_JAVA_JUNIT'|'APPIUM_JAVA_TESTNG'|'CALABASH'|'INSTRUMENTATION'|'UIAUTOMATION'|'UIAUTOMATOR'|'XCTEST',
    'platform': 'ANDROID'|'IOS',
    'created': datetime(2015, 1, 1),
    'status': 'PENDING'|'PROCESSING'|'SCHEDULING'|'RUNNING'|'COMPLETED',
    'result': 'PENDING'|'PASSED'|'WARNED'|'FAILED'|'SKIPPED'|'ERRORED'|'STOPPED',
    'started': datetime(2015, 1, 1),
    'stopped': datetime(2015, 1, 1),
    'counters': {
      'total': 123,
      'passed': 123,
      'failed': 123,
      'warned': 123,
      'errored': 123,
      'stopped': 123,
      'skipped': 123
    },
    'message': 'string',
    'totalJobs': 123,
    'completedJobs': 123,
    'billingMethod': 'METERED'|'UNMETERED'
  }
}

```

Response Structure

- (*dict*) –

Represents the result of a schedule run request.

- **run** (*dict*) –

Information about the scheduled run.

- * **arn** (*string*) –

The run's ARN.

- * **name** (*string*) –

The run's name.

- * **type** (*string*) –

The run's type.

Must be one of the following values:

- BUILTIN_FUZZ: The built-in fuzz type.
- BUILTIN_EXPLORER: For Android, an app explorer that will traverse an Android app, interacting with it and capturing screenshots at the same time.
- APPIUM_JAVA_JUNIT: The Appium Java JUnit type.
- APPIUM_JAVA_TESTNG: The Appium Java TestNG type.
- CALABASH: The Calabash type.
- INSTRUMENTATION: The Instrumentation type.
- UIAUTOMATION: The uiautomation type.
- UIAUTOMATOR: The uiautomator type.
- XCTEST: The XCode test type.

- * **platform** (*string*) –

The run's platform.

Allowed values include:

- **ANDROID**: The Android platform.
- **IOS**: The iOS platform.

* **created** (*datetime*) –

When the run was created.

* **status** (*string*) –

The run's status.

Allowed values include:

- **COMPLETED**: A completed status.
- **PENDING**: A pending status.
- **PROCESSING**: A processing status.
- **RUNNING**: A running status.
- **SCHEDULING**: A scheduling status.

* **result** (*string*) –

The run's result.

Allowed values include:

- **ERRORED**: An error condition.
- **FAILED**: A failed condition.
- **SKIPPED**: A skipped condition.
- **STOPPED**: A stopped condition.
- **PASSED**: A passing condition.
- **PENDING**: A pending condition.
- **WARNED**: A warning condition.

* **started** (*datetime*) –

The run's start time.

* **stopped** (*datetime*) –

The run's stop time.

* **counters** (*dict*) –

The run's result counters.

· **total** (*integer*) –

The total number of entities.

· **passed** (*integer*) –

The number of passed entities.

· **failed** (*integer*) –

The number of failed entities.

· **warned** (*integer*) –

The number of warned entities.

· **errored** (*integer*) –

The number of errored entities.

· **stopped** (*integer*) –

The number of stopped entities.

· **skipped** (*integer*) –

The number of skipped entities.

* **message** (*string*) –

A message about the run's result.

* **totalJobs** (*integer*) –

The total number of jobs for the run.

* **completedJobs** (*integer*) –

The total number of completed jobs.

* **billingMethod** (*string*) –

Specifies the billing method for a test run: `metered` or `unmetered`. If the parameter is not specified, the default value is `unmetered`.

DirectConnect

Table of Contents

- [DirectConnect](#)
 - [Client](#)

Client

class DirectConnect.**Client**

A low-level client representing AWS Direct Connect:

```
import boto3
```

```
client = boto3.client('directconnect')
```

These are the available methods:

- `allocate_connection_on_interconnect()`
- `allocate_private_virtual_interface()`
- `allocate_public_virtual_interface()`
- `can_paginate()`
- `confirm_connection()`
- `confirm_private_virtual_interface()`
- `confirm_public_virtual_interface()`
- `create_connection()`
- `create_interconnect()`
- `create_private_virtual_interface()`
- `create_public_virtual_interface()`
- `delete_connection()`
- `delete_interconnect()`
- `delete_virtual_interface()`
- `describe_connections()`
- `describe_connections_on_interconnect()`
- `describe_interconnects()`
- `describe_locations()`
- `describe_virtual_gateways()`
- `describe_virtual_interfaces()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_waiter()`

allocate_connection_on_interconnect (**kwargs)

Creates a hosted connection on an interconnect.

Allocates a VLAN number and a specified amount of bandwidth for use by a hosted connection on the given interconnect.

Request Syntax

```
response = client.allocate_connection_on_interconnect(
    bandwidth='string',
    connectionName='string',
    ownerAccount='string',
    interconnectId='string',
    vlan=123
)
```

Parameters

- **bandwidth** (*string*) – [REQUIRED]
Bandwidth of the connection.
Example: “500Mbps “
Default: None
- **connectionName** (*string*) – [REQUIRED]
Name of the provisioned connection.
Example: “500M Connection to AWS “
Default: None
- **ownerAccount** (*string*) – [REQUIRED]
Numeric account Id of the customer for whom the connection will be provisioned.
Example: 123443215678
Default: None
- **interconnectId** (*string*) – [REQUIRED]
ID of the interconnect on which the connection will be provisioned.
Example: dxcon-456abc78
Default: None
- **vlan** (*integer*) – [REQUIRED]
The dedicated VLAN provisioned to the connection.
Example: 101
Default: None

Return type dict

Returns

Response Syntax

```
{
    'ownerAccount': 'string',
    'connectionId': 'string',
    'connectionName': 'string',
    'connectionState': 'ordering'|'requested'|'pending'|'available'|'down'|'deleti
    'region': 'string',
```

```

'location': 'string',
'bandwidth': 'string',
'vlan': 123,
'partnerName': 'string'
}

```

Response Structure

- (*dict*) –

A connection represents the physical network connection between the AWS Direct Connect location and the customer.

- **ownerAccount** (*string*) –
- **connectionId** (*string*) –

ID of the connection.

Example: dxcon-fg5678gh

Default: None

- **connectionName** (*string*) –

The name of the connection.

Example: “*My Connection to AWS*”

Default: None

- **connectionState** (*string*) – State of the connection.
 - * **Ordering** : The initial state of a hosted connection provisioned on an interconnect. The connection stays in the ordering state until the owner of the hosted connection confirms or declines the connection order.
 - * **Requested** : The initial state of a standard connection. The connection stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.
 - * **Pending** : The connection has been approved, and is being initialized.
 - * **Available** : The network link is up, and the connection is ready for use.
 - * **Down** : The network link is down.
 - * **Deleted** : The connection has been deleted.
 - * **Rejected** : A hosted connection in the ‘Ordering’ state will enter the ‘Rejected’ state if it is deleted by the end customer.

- **region** (*string*) –

The AWS region where the connection is located.

Example: us-east-1

Default: None

- **location** (*string*) –

Where the connection is located.

Example: EqSV5

Default: None

- **bandwidth** (*string*) –

Bandwidth of the connection.

Example: 1Gbps (for regular connections), or 500Mbps (for hosted connections)

- Default: None
- **vlan** (*integer*) –
The VLAN ID.
Example: 101
- **partnerName** (*string*) –

allocate_private_virtual_interface (***kwargs*)

Provisions a private virtual interface to be owned by a different customer.

The owner of a connection calls this function to provision a private virtual interface which will be owned by another AWS customer.

Virtual interfaces created using this function must be confirmed by the virtual interface owner by calling `ConfirmPrivateVirtualInterface`. Until this step has been completed, the virtual interface will be in ‘Confirming’ state, and will not be available for handling traffic.

Request Syntax

```
response = client.allocate_private_virtual_interface(
    connectionId='string',
    ownerAccount='string',
    newPrivateVirtualInterfaceAllocation={
        'virtualInterfaceName': 'string',
        'vlan': 123,
        'asn': 123,
        'authKey': 'string',
        'amazonAddress': 'string',
        'customerAddress': 'string'
    }
)
```

Parameters

- **connectionId** (*string*) – **[REQUIRED]**
The connection ID on which the private virtual interface is provisioned.
Default: None
- **ownerAccount** (*string*) – **[REQUIRED]**
The AWS account that will own the new private virtual interface.
Default: None
- **newPrivateVirtualInterfaceAllocation** (*dict*) – **[REQUIRED]**
Detailed information for the private virtual interface to be provisioned.
Default: None
 - **virtualInterfaceName** (*string*) – **[REQUIRED]**
The name of the virtual interface assigned by the customer.
Example: “My VPC”
 - **vlan** (*integer*) – **[REQUIRED]**
The VLAN ID.
Example: 101
 - **asn** (*integer*) – **[REQUIRED]**
Autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.

Example: 65000

– **authKey** (*string*) –

Authentication key for BGP configuration.

Example: asdf34example

– **amazonAddress** (*string*) –

IP address assigned to the Amazon interface.

Example: 192.168.1.1/30

– **customerAddress** (*string*) –

IP address assigned to the customer interface.

Example: 192.168.1.2/30

Return type dict

Returns

Response Syntax

```
{
  'ownerAccount': 'string',
  'virtualInterfaceId': 'string',
  'location': 'string',
  'connectionId': 'string',
  'virtualInterfaceType': 'string',
  'virtualInterfaceName': 'string',
  'vlan': 123,
  'asn': 123,
  'authKey': 'string',
  'amazonAddress': 'string',
  'customerAddress': 'string',
  'virtualInterfaceState': 'confirming'|'verifying'|'pending'|'available'|'delet
  'customerRouterConfig': 'string',
  'virtualGatewayId': 'string',
  'routeFilterPrefixes': [
    {
      'cidr': 'string'
    },
  ],
}
```

Response Structure

- (*dict*) –

A virtual interface (VLAN) transmits the traffic between the AWS Direct Connect location and the customer.

- **ownerAccount** (*string*) –
- **virtualInterfaceId** (*string*) –

ID of the virtual interface.

Example: dxvif-123dfg56

Default: None

- **location** (*string*) –

Where the connection is located.

Example: EqSV5

Default: None

- **connectionId** (*string*) –
ID of the connection.
Example: dxcon-fg5678gh
Default: None
- **virtualInterfaceType** (*string*) –
The type of virtual interface.
Example: private (Amazon VPC) or public (Amazon S3, Amazon DynamoDB, and so on.)
- **virtualInterfaceName** (*string*) –
The name of the virtual interface assigned by the customer.
Example: “My VPC”
- **vlan** (*integer*) –
The VLAN ID.
Example: 101
- **asn** (*integer*) –
Autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.
Example: 65000
- **authKey** (*string*) –
Authentication key for BGP configuration.
Example: asdf34example
- **amazonAddress** (*string*) –
IP address assigned to the Amazon interface.
Example: 192.168.1.1/30
- **customerAddress** (*string*) –
IP address assigned to the customer interface.
Example: 192.168.1.2/30
- **virtualInterfaceState** (*string*) – State of the virtual interface.
 - * **Confirming** : The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.
 - * **Verifying** : This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.
 - * **Pending** : A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.
 - * **Available** : A virtual interface that is able to forward traffic.
 - * **Deleting** : A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.
 - * **Deleted** : A virtual interface that cannot forward traffic.
 - * **Rejected** : The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the ‘Confirming’ state is

deleted by the virtual interface owner, the virtual interface will enter the 'Rejected' state.

– **customerRouterConfig** (*string*) –

Information for generating the customer router configuration.

– **virtualGatewayId** (*string*) –

The ID of the virtual private gateway to a VPC. This only applies to private virtual interfaces.

Example: vgw-123er56

– **routeFilterPrefixes** (*list*) –

A list of routes to be advertised to the AWS network in this region (public virtual interface) or your VPC (private virtual interface).

* (*dict*) –

A route filter prefix that the customer can advertise through Border Gateway Protocol (BGP) over a public virtual interface.

· **cidr** (*string*) –

CIDR notation for the advertised route. Multiple routes are separated by commas.

Example: 10.10.10.0/24,10.10.11.0/24

allocate_public_virtual_interface (***kwargs*)

Provisions a public virtual interface to be owned by a different customer.

The owner of a connection calls this function to provision a public virtual interface which will be owned by another AWS customer.

Virtual interfaces created using this function must be confirmed by the virtual interface owner by calling `ConfirmPublicVirtualInterface`. Until this step has been completed, the virtual interface will be in 'Confirming' state, and will not be available for handling traffic.

Request Syntax

```
response = client.allocate_public_virtual_interface(
    connectionId='string',
    ownerAccount='string',
    newPublicVirtualInterfaceAllocation={
        'virtualInterfaceName': 'string',
        'vlan': 123,
        'asn': 123,
        'authKey': 'string',
        'amazonAddress': 'string',
        'customerAddress': 'string',
        'routeFilterPrefixes': [
            {
                'cidr': 'string'
            },
        ]
    }
)
```

Parameters

• **connectionId** (*string*) – [REQUIRED]

The connection ID on which the public virtual interface is provisioned.

Default: None

- **ownerAccount** (*string*) – [REQUIRED]

The AWS account that will own the new public virtual interface.

Default: None

- **newPublicVirtualInterfaceAllocation** (*dict*) – [REQUIRED]

Detailed information for the public virtual interface to be provisioned.

Default: None

- **virtualInterfaceName** (*string*) – [REQUIRED]

The name of the virtual interface assigned by the customer.

Example: “My VPC”

- **vlan** (*integer*) – [REQUIRED]

The VLAN ID.

Example: 101

- **asn** (*integer*) – [REQUIRED]

Autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.

Example: 65000

- **authKey** (*string*) –

Authentication key for BGP configuration.

Example: asdf34example

- **amazonAddress** (*string*) – [REQUIRED]

IP address assigned to the Amazon interface.

Example: 192.168.1.1/30

- **customerAddress** (*string*) – [REQUIRED]

IP address assigned to the customer interface.

Example: 192.168.1.2/30

- **routeFilterPrefixes** (*list*) – [REQUIRED]

A list of routes to be advertised to the AWS network in this region (public virtual interface) or your VPC (private virtual interface).

* (*dict*) –

A route filter prefix that the customer can advertise through Border Gateway Protocol (BGP) over a public virtual interface.

· **cidr** (*string*) –

CIDR notation for the advertised route. Multiple routes are separated by commas.

Example: 10.10.10.0/24,10.10.11.0/24

Return type dict

Returns

Response Syntax

```
{
    'ownerAccount': 'string',
    'virtualInterfaceId': 'string',
    'location': 'string',
```

```

'connectionId': 'string',
'virtualInterfaceType': 'string',
'virtualInterfaceName': 'string',
'vlan': 123,
'asn': 123,
'authKey': 'string',
'amazonAddress': 'string',
'customerAddress': 'string',
'virtualInterfaceState': 'confirming'|'verifying'|'pending'|'available'|'delet
'customerRouterConfig': 'string',
'virtualGatewayId': 'string',
'routeFilterPrefixes': [
    {
        'cidr': 'string'
    },
]
}

```

Response Structure

- (*dict*) –

A virtual interface (VLAN) transmits the traffic between the AWS Direct Connect location and the customer.

- **ownerAccount** (*string*) –
- **virtualInterfaceId** (*string*) –

ID of the virtual interface.

Example: dxvif-123dfg56

Default: None

- **location** (*string*) –

Where the connection is located.

Example: EqSV5

Default: None

- **connectionId** (*string*) –

ID of the connection.

Example: dxcon-fg5678gh

Default: None

- **virtualInterfaceType** (*string*) –

The type of virtual interface.

Example: private (Amazon VPC) or public (Amazon S3, Amazon DynamoDB, and so on.)

- **virtualInterfaceName** (*string*) –

The name of the virtual interface assigned by the customer.

Example: “My VPC”

- **vlan** (*integer*) –

The VLAN ID.

Example: 101

- **asn** (*integer*) –

Autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.

Example: 65000

– **authKey** (*string*) –

Authentication key for BGP configuration.

Example: asdf34example

– **amazonAddress** (*string*) –

IP address assigned to the Amazon interface.

Example: 192.168.1.1/30

– **customerAddress** (*string*) –

IP address assigned to the customer interface.

Example: 192.168.1.2/30

– **virtualInterfaceState** (*string*) – State of the virtual interface.

- * **Confirming** : The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.

- * **Verifying** : This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.

- * **Pending** : A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.

- * **Available** : A virtual interface that is able to forward traffic.

- * **Deleting** : A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.

- * **Deleted** : A virtual interface that cannot forward traffic.

- * **Rejected** : The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the ‘Confirming’ state is deleted by the virtual interface owner, the virtual interface will enter the ‘Rejected’ state.

– **customerRouterConfig** (*string*) –

Information for generating the customer router configuration.

– **virtualGatewayId** (*string*) –

The ID of the virtual private gateway to a VPC. This only applies to private virtual interfaces.

Example: vgw-123er56

– **routeFilterPrefixes** (*list*) –

A list of routes to be advertised to the AWS network in this region (public virtual interface) or your VPC (private virtual interface).

- * (*dict*) –

A route filter prefix that the customer can advertise through Border Gateway Protocol (BGP) over a public virtual interface.

- **cidr** (*string*) –

CIDR notation for the advertised route. Multiple routes are separated by commas.

Example: 10.10.10.0/24,10.10.11.0/24

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

confirm_connection (***kwargs*)

Confirm the creation of a hosted connection on an interconnect.

Upon creation, the hosted connection is initially in the 'Ordering' state, and will remain in this state until the owner calls `ConfirmConnection` to confirm creation of the hosted connection.

Request Syntax

```
response = client.confirm_connection(
    connectionId='string'
)
```

Parameters *connectionId* (*string*) – [REQUIRED]

ID of the connection.

Example: `dxcon-fg5678gh`

Default: None

Return type dict

Returns**Response Syntax**

```
{
    'connectionState': 'ordering'|'requested'|'pending'|'available'|'down'|'deleted'
}
```

Response Structure

- (*dict*) –

The response received when `ConfirmConnection` is called.

– **connectionState** (*string*) – State of the connection.

- * **Ordering** : The initial state of a hosted connection provisioned on an interconnect. The connection stays in the ordering state until the owner of the hosted connection confirms or declines the connection order.
- * **Requested** : The initial state of a standard connection. The connection stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.
- * **Pending** : The connection has been approved, and is being initialized.
- * **Available** : The network link is up, and the connection is ready for use.
- * **Down** : The network link is down.
- * **Deleted** : The connection has been deleted.
- * **Rejected** : A hosted connection in the 'Ordering' state will enter the 'Rejected' state if it is deleted by the end customer.

confirm_private_virtual_interface (***kwargs*)

Accept ownership of a private virtual interface created by another customer.

After the virtual interface owner calls this function, the virtual interface will be created and attached to the given virtual private gateway, and will be available for handling traffic.

Request Syntax

```
response = client.confirm_private_virtual_interface(
    virtualInterfaceId='string',
    virtualGatewayId='string'
)
```

Parameters

- **virtualInterfaceId** (*string*) – [REQUIRED]

ID of the virtual interface.

Example: dxvif-123dfg56

Default: None

- **virtualGatewayId** (*string*) – [REQUIRED]

ID of the virtual private gateway that will be attached to the virtual interface.

A virtual private gateway can be managed via the Amazon Virtual Private Cloud (VPC) console or the [EC2 CreateVpnGateway](#) action.

Default: None

Return type dict

Returns**Response Syntax**

```
{
    'virtualInterfaceState': 'confirming'|'verifying'|'pending'|'available'|'deleting'|'deleted'
}
```

Response Structure

- (*dict*) –

The response received when ConfirmPrivateVirtualInterface is called.

- **virtualInterfaceState** (*string*) – State of the virtual interface.

- * **Confirming** : The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.
- * **Verifying** : This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.
- * **Pending** : A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.
- * **Available** : A virtual interface that is able to forward traffic.
- * **Deleting** : A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.
- * **Deleted** : A virtual interface that cannot forward traffic.
- * **Rejected** : The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the ‘Confirming’ state is

deleted by the virtual interface owner, the virtual interface will enter the ‘Rejected’ state.

confirm_public_virtual_interface (***kwargs*)

Accept ownership of a public virtual interface created by another customer.

After the virtual interface owner calls this function, the specified virtual interface will be created and made available for handling traffic.

Request Syntax

```
response = client.confirm_public_virtual_interface(
    virtualInterfaceId='string'
)
```

Parameters **virtualInterfaceId** (*string*) – [REQUIRED]

ID of the virtual interface.

Example: dxvif-123dfg56

Default: None

Return type dict

Returns

Response Syntax

```
{
    'virtualInterfaceState': 'confirming'|'verifying'|'pending'|'available'|'deleting'|'deleted'|'rejected'
}
```

Response Structure

- (*dict*) –

The response received when ConfirmPublicVirtualInterface is called.

– **virtualInterfaceState** (*string*) – State of the virtual interface.

- * **Confirming** : The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.
- * **Verifying** : This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.
- * **Pending** : A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.
- * **Available** : A virtual interface that is able to forward traffic.
- * **Deleting** : A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.
- * **Deleted** : A virtual interface that cannot forward traffic.
- * **Rejected** : The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the ‘Confirming’ state is deleted by the virtual interface owner, the virtual interface will enter the ‘Rejected’ state.

create_connection (***kwargs*)

Creates a new connection between the customer network and a specific AWS Direct Connect location.

A connection links your internal network to an AWS Direct Connect location over a standard 1 gigabit or 10 gigabit Ethernet fiber-optic cable. One end of the cable is connected to your router, the other to an

AWS Direct Connect router. An AWS Direct Connect location provides access to Amazon Web Services in the region it is associated with. You can establish connections with AWS Direct Connect locations in multiple regions, but a connection in one region does not provide connectivity to other regions.

Request Syntax

```
response = client.create_connection(
    location='string',
    bandwidth='string',
    connectionName='string'
)
```

Parameters

- **location** (*string*) – [REQUIRED]
Where the connection is located.
Example: EqSV5
Default: None
- **bandwidth** (*string*) – [REQUIRED]
Bandwidth of the connection.
Example: 1Gbps
Default: None
- **connectionName** (*string*) – [REQUIRED]
The name of the connection.
Example: “*My Connection to AWS*”
Default: None

Return type dict

Returns

Response Syntax

```
{
    'ownerAccount': 'string',
    'connectionId': 'string',
    'connectionName': 'string',
    'connectionState': 'ordering'|'requested'|'pending'|'available'|'down'|'delet
    'region': 'string',
    'location': 'string',
    'bandwidth': 'string',
    'vlan': 123,
    'partnerName': 'string'
}
```

Response Structure

- (*dict*) –
A connection represents the physical network connection between the AWS Direct Connect location and the customer.
 - **ownerAccount** (*string*) –
 - **connectionId** (*string*) –
ID of the connection.
Example: dxcon-fg5678gh

Default: None

– **connectionName** (*string*) –

The name of the connection.

Example: “*My Connection to AWS*”

Default: None

– **connectionState** (*string*) – State of the connection.

* **Ordering** : The initial state of a hosted connection provisioned on an interconnect. The connection stays in the ordering state until the owner of the hosted connection confirms or declines the connection order.

* **Requested** : The initial state of a standard connection. The connection stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.

* **Pending** : The connection has been approved, and is being initialized.

* **Available** : The network link is up, and the connection is ready for use.

* **Down** : The network link is down.

* **Deleted** : The connection has been deleted.

* **Rejected** : A hosted connection in the ‘Ordering’ state will enter the ‘Rejected’ state if it is deleted by the end customer.

– **region** (*string*) –

The AWS region where the connection is located.

Example: us-east-1

Default: None

– **location** (*string*) –

Where the connection is located.

Example: EqSV5

Default: None

– **bandwidth** (*string*) –

Bandwidth of the connection.

Example: 1Gbps (for regular connections), or 500Mbps (for hosted connections)

Default: None

– **vlan** (*integer*) –

The VLAN ID.

Example: 101

– **partnerName** (*string*) –

create_interconnect (***kwargs*)

Creates a new interconnect between a AWS Direct Connect partner’s network and a specific AWS Direct Connect location.

An interconnect is a connection which is capable of hosting other connections. The AWS Direct Connect partner can use an interconnect to provide sub-1Gbps AWS Direct Connect service to tier 2 customers who do not have their own connections. Like a standard connection, an interconnect links the AWS Direct Connect partner’s network to an AWS Direct Connect location over a standard 1 Gbps or 10 Gbps

Ethernet fiber-optic cable. One end is connected to the partner's router, the other to an AWS Direct Connect router.

For each end customer, the AWS Direct Connect partner provisions a connection on their interconnect by calling `AllocateConnectionOnInterconnect`. The end customer can then connect to AWS resources by creating a virtual interface on their connection, using the VLAN assigned to them by the AWS Direct Connect partner.

Request Syntax

```
response = client.create_interconnect(
    interconnectName='string',
    bandwidth='string',
    location='string'
)
```

Parameters

- **interconnectName** (*string*) – [REQUIRED]

The name of the interconnect.

Example: “1G Interconnect to AWS “

Default: None

- **bandwidth** (*string*) – [REQUIRED]

The port bandwidth

Example: 1Gbps

Default: None

Available values: 1Gbps,10Gbps

- **location** (*string*) – [REQUIRED]

Where the interconnect is located

Example: EqSV5

Default: None

Return type dict

Returns

Response Syntax

```
{
    'interconnectId': 'string',
    'interconnectName': 'string',
    'interconnectState': 'requested'|'pending'|'available'|'down'|'deleting'|'deleted',
    'region': 'string',
    'location': 'string',
    'bandwidth': 'string'
}
```

Response Structure

- (*dict*) –

An interconnect is a connection that can host other connections.

Like a standard AWS Direct Connect connection, an interconnect represents the physical connection between an AWS Direct Connect partner's network and a specific Direct Connect location. An AWS Direct Connect partner who owns an interconnect can provision hosted connections on the interconnect for their

end customers, thereby providing the end customers with connectivity to AWS services.

The resources of the interconnect, including bandwidth and VLAN numbers, are shared by all of the hosted connections on the interconnect, and the owner of the interconnect determines how these resources are assigned.

– **interconnectId** (*string*) –

The ID of the interconnect.

Example: dxcon-abc123

– **interconnectName** (*string*) –

The name of the interconnect.

Example: “IG Interconnect to AWS “

– **interconnectState** (*string*) – State of the interconnect.

* **Requested** : The initial state of an interconnect. The interconnect stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.

* **Pending** : The interconnect has been approved, and is being initialized.

* **Available** : The network link is up, and the interconnect is ready for use.

* **Down** : The network link is down.

* **Deleted** : The interconnect has been deleted.

– **region** (*string*) –

The AWS region where the connection is located.

Example: us-east-1

Default: None

– **location** (*string*) –

Where the connection is located.

Example: EqSV5

Default: None

– **bandwidth** (*string*) –

Bandwidth of the connection.

Example: 1Gbps

Default: None

create_private_virtual_interface (***kwargs*)

Creates a new private virtual interface. A virtual interface is the VLAN that transports AWS Direct Connect traffic. A private virtual interface supports sending traffic to a single virtual private cloud (VPC).

Request Syntax

```
response = client.create_private_virtual_interface(
    connectionId='string',
    newPrivateVirtualInterface={
        'virtualInterfaceName': 'string',
        'vlan': 123,
        'asn': 123,
        'authKey': 'string',
        'amazonAddress': 'string',
```

```
        'customerAddress': 'string',
        'virtualGatewayId': 'string'
    }
)
```

Parameters

- **connectionId** (*string*) – [REQUIRED]
ID of the connection.
Example: dxcon-fg5678gh
Default: None
- **newPrivateVirtualInterface** (*dict*) – [REQUIRED]
Detailed information for the private virtual interface to be created.
Default: None
 - **virtualInterfaceName** (*string*) – [REQUIRED]
The name of the virtual interface assigned by the customer.
Example: “My VPC”
 - **vlan** (*integer*) – [REQUIRED]
The VLAN ID.
Example: 101
 - **asn** (*integer*) – [REQUIRED]
Autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.
Example: 65000
 - **authKey** (*string*) –
Authentication key for BGP configuration.
Example: asdf34example
 - **amazonAddress** (*string*) –
IP address assigned to the Amazon interface.
Example: 192.168.1.1/30
 - **customerAddress** (*string*) –
IP address assigned to the customer interface.
Example: 192.168.1.2/30
 - **virtualGatewayId** (*string*) – [REQUIRED]
The ID of the virtual private gateway to a VPC. This only applies to private virtual interfaces.
Example: vgw-123er56

Return type dict

Returns

Response Syntax

```
{
    'ownerAccount': 'string',
    'virtualInterfaceId': 'string',
    'location': 'string',
```

```

'connectionId': 'string',
'virtualInterfaceType': 'string',
'virtualInterfaceName': 'string',
'vlan': 123,
'asn': 123,
'authKey': 'string',
'amazonAddress': 'string',
'customerAddress': 'string',
'virtualInterfaceState': 'confirming'|'verifying'|'pending'|'available'|'delet
'customerRouterConfig': 'string',
'virtualGatewayId': 'string',
'routeFilterPrefixes': [
    {
        'cidr': 'string'
    },
]
}

```

Response Structure

- *(dict)* –

A virtual interface (VLAN) transmits the traffic between the AWS Direct Connect location and the customer.

- **ownerAccount** (*string*) –
- **virtualInterfaceId** (*string*) –

ID of the virtual interface.

Example: dxvif-123dfg56

Default: None

- **location** (*string*) –

Where the connection is located.

Example: EqSV5

Default: None

- **connectionId** (*string*) –

ID of the connection.

Example: dxcon-fg5678gh

Default: None

- **virtualInterfaceType** (*string*) –

The type of virtual interface.

Example: private (Amazon VPC) or public (Amazon S3, Amazon DynamoDB, and so on.)

- **virtualInterfaceName** (*string*) –

The name of the virtual interface assigned by the customer.

Example: “My VPC”

- **vlan** (*integer*) –

The VLAN ID.

Example: 101

- **asn** (*integer*) –

Autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.

Example: 65000

– **authKey** (*string*) –

Authentication key for BGP configuration.

Example: asdf34example

– **amazonAddress** (*string*) –

IP address assigned to the Amazon interface.

Example: 192.168.1.1/30

– **customerAddress** (*string*) –

IP address assigned to the customer interface.

Example: 192.168.1.2/30

– **virtualInterfaceState** (*string*) – State of the virtual interface.

- * **Confirming** : The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.

- * **Verifying** : This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.

- * **Pending** : A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.

- * **Available** : A virtual interface that is able to forward traffic.

- * **Deleting** : A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.

- * **Deleted** : A virtual interface that cannot forward traffic.

- * **Rejected** : The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the ‘Confirming’ state is deleted by the virtual interface owner, the virtual interface will enter the ‘Rejected’ state.

– **customerRouterConfig** (*string*) –

Information for generating the customer router configuration.

– **virtualGatewayId** (*string*) –

The ID of the virtual private gateway to a VPC. This only applies to private virtual interfaces.

Example: vgw-123er56

– **routeFilterPrefixes** (*list*) –

A list of routes to be advertised to the AWS network in this region (public virtual interface) or your VPC (private virtual interface).

- * (*dict*) –

A route filter prefix that the customer can advertise through Border Gateway Protocol (BGP) over a public virtual interface.

- **cidr** (*string*) –

CIDR notation for the advertised route. Multiple routes are separated by commas.

Example: 10.10.10.0/24,10.10.11.0/24

create_public_virtual_interface (***kwargs*)

Creates a new public virtual interface. A virtual interface is the VLAN that transports AWS Direct Connect traffic. A public virtual interface supports sending traffic to public services of AWS such as Amazon Simple Storage Service (Amazon S3).

Request Syntax

```
response = client.create_public_virtual_interface(
    connectionId='string',
    newPublicVirtualInterface={
        'virtualInterfaceName': 'string',
        'vlan': 123,
        'asn': 123,
        'authKey': 'string',
        'amazonAddress': 'string',
        'customerAddress': 'string',
        'routeFilterPrefixes': [
            {
                'cidr': 'string'
            },
        ]
    }
)
```

Parameters

- **connectionId** (*string*) – **[REQUIRED]**
ID of the connection.
Example: dxcon-fg5678gh
Default: None
- **newPublicVirtualInterface** (*dict*) – **[REQUIRED]**
Detailed information for the public virtual interface to be created.
Default: None
 - **virtualInterfaceName** (*string*) – **[REQUIRED]**
The name of the virtual interface assigned by the customer.
Example: “My VPC”
 - **vlan** (*integer*) – **[REQUIRED]**
The VLAN ID.
Example: 101
 - **asn** (*integer*) – **[REQUIRED]**
Autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.
Example: 65000
 - **authKey** (*string*) –
Authentication key for BGP configuration.
Example: asdf34example
 - **amazonAddress** (*string*) – **[REQUIRED]**
IP address assigned to the Amazon interface.
Example: 192.168.1.1/30

- **customerAddress** (*string*) – [REQUIRED]

IP address assigned to the customer interface.

Example: 192.168.1.2/30

- **routeFilterPrefixes** (*list*) – [REQUIRED]

A list of routes to be advertised to the AWS network in this region (public virtual interface) or your VPC (private virtual interface).

* (*dict*) –

A route filter prefix that the customer can advertise through Border Gateway Protocol (BGP) over a public virtual interface.

· **cidr** (*string*) –

CIDR notation for the advertised route. Multiple routes are separated by commas.

Example: 10.10.10.0/24,10.10.11.0/24

Return type dict

Returns

Response Syntax

```
{
    'ownerAccount': 'string',
    'virtualInterfaceId': 'string',
    'location': 'string',
    'connectionId': 'string',
    'virtualInterfaceType': 'string',
    'virtualInterfaceName': 'string',
    'vlan': 123,
    'asn': 123,
    'authKey': 'string',
    'amazonAddress': 'string',
    'customerAddress': 'string',
    'virtualInterfaceState': 'confirming'|'verifying'|'pending'|'available'|'deleted',
    'customerRouterConfig': 'string',
    'virtualGatewayId': 'string',
    'routeFilterPrefixes': [
        {
            'cidr': 'string'
        },
    ],
}
```

Response Structure

- (*dict*) –

A virtual interface (VLAN) transmits the traffic between the AWS Direct Connect location and the customer.

- **ownerAccount** (*string*) –

- **virtualInterfaceId** (*string*) –

ID of the virtual interface.

Example: dxvif-123dfg56

Default: None

- **location** (*string*) –

Where the connection is located.

Example: EqSV5

Default: None

- **connectionId** (*string*) –

ID of the connection.

Example: dxcon-fg5678gh

Default: None

- **virtualInterfaceType** (*string*) –

The type of virtual interface.

Example: private (Amazon VPC) or public (Amazon S3, Amazon DynamoDB, and so on.)

- **virtualInterfaceName** (*string*) –

The name of the virtual interface assigned by the customer.

Example: “My VPC”

- **vlan** (*integer*) –

The VLAN ID.

Example: 101

- **asn** (*integer*) –

Autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.

Example: 65000

- **authKey** (*string*) –

Authentication key for BGP configuration.

Example: asdf34example

- **amazonAddress** (*string*) –

IP address assigned to the Amazon interface.

Example: 192.168.1.1/30

- **customerAddress** (*string*) –

IP address assigned to the customer interface.

Example: 192.168.1.2/30

- **virtualInterfaceState** (*string*) – State of the virtual interface.

- * **Confirming** : The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.

- * **Verifying** : This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.

- * **Pending** : A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.

- * **Available** : A virtual interface that is able to forward traffic.

- * **Deleting** : A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.

- * **Deleted** : A virtual interface that cannot forward traffic.

* **Rejected** : The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the 'Confirming' state is deleted by the virtual interface owner, the virtual interface will enter the 'Rejected' state.

– **customerRouterConfig** (*string*) –

Information for generating the customer router configuration.

– **virtualGatewayId** (*string*) –

The ID of the virtual private gateway to a VPC. This only applies to private virtual interfaces.

Example: vgw-123er56

– **routeFilterPrefixes** (*list*) –

A list of routes to be advertised to the AWS network in this region (public virtual interface) or your VPC (private virtual interface).

* (*dict*) –

A route filter prefix that the customer can advertise through Border Gateway Protocol (BGP) over a public virtual interface.

· **cidr** (*string*) –

CIDR notation for the advertised route. Multiple routes are separated by commas.

Example: 10.10.10.0/24,10.10.11.0/24

delete_connection (***kwargs*)

Deletes the connection.

Deleting a connection only stops the AWS Direct Connect port hour and data transfer charges. You need to cancel separately with the providers any services or charges for cross-connects or network circuits that connect you to the AWS Direct Connect location.

Request Syntax

```
response = client.delete_connection(
    connectionId='string'
)
```

Parameters **connectionId** (*string*) – **[REQUIRED]**

ID of the connection.

Example: dxcon-fg5678gh

Default: None

Return type dict

Returns

Response Syntax

```
{
    'ownerAccount': 'string',
    'connectionId': 'string',
    'connectionName': 'string',
    'connectionState': 'ordering'|'requested'|'pending'|'available'|'down'|'delet
    'region': 'string',
    'location': 'string',
    'bandwidth': 'string',
    'vlan': 123,
```

```

    'partnerName': 'string'
}

```

Response Structure

- (*dict*) –

A connection represents the physical network connection between the AWS Direct Connect location and the customer.

- **ownerAccount** (*string*) –
- **connectionId** (*string*) –

ID of the connection.

Example: dxcon-fg5678gh

Default: None

- **connectionName** (*string*) –

The name of the connection.

Example: “*My Connection to AWS*”

Default: None

- **connectionState** (*string*) – State of the connection.
 - * **Ordering** : The initial state of a hosted connection provisioned on an interconnect. The connection stays in the ordering state until the owner of the hosted connection confirms or declines the connection order.
 - * **Requested** : The initial state of a standard connection. The connection stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.
 - * **Pending** : The connection has been approved, and is being initialized.
 - * **Available** : The network link is up, and the connection is ready for use.
 - * **Down** : The network link is down.
 - * **Deleted** : The connection has been deleted.
 - * **Rejected** : A hosted connection in the ‘Ordering’ state will enter the ‘Rejected’ state if it is deleted by the end customer.

- **region** (*string*) –

The AWS region where the connection is located.

Example: us-east-1

Default: None

- **location** (*string*) –

Where the connection is located.

Example: EqSV5

Default: None

- **bandwidth** (*string*) –

Bandwidth of the connection.

Example: 1Gbps (for regular connections), or 500Mbps (for hosted connections)

Default: None

- **vlan** (*integer*) –
The VLAN ID.
Example: 101
- **partnerName** (*string*) –

delete_interconnect (***kwargs*)

Deletes the specified interconnect.

Request Syntax

```
response = client.delete_interconnect(  
    interconnectId='string'  
)
```

Parameters **interconnectId** (*string*) – [REQUIRED]

The ID of the interconnect.

Example: dxcon-abc123

Return type dict

Returns

Response Syntax

```
{  
    'interconnectState': 'requested'|'pending'|'available'|'down'|'deleting'|'deleted'  
}
```

Response Structure

- (*dict*) –

The response received when DeleteInterconnect is called.

- **interconnectState** (*string*) – State of the interconnect.
 - * **Requested** : The initial state of an interconnect. The interconnect stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.
 - * **Pending** : The interconnect has been approved, and is being initialized.
 - * **Available** : The network link is up, and the interconnect is ready for use.
 - * **Down** : The network link is down.
 - * **Deleted** : The interconnect has been deleted.

delete_virtual_interface (***kwargs*)

Deletes a virtual interface.

Request Syntax

```
response = client.delete_virtual_interface(  
    virtualInterfaceId='string'  
)
```

Parameters **virtualInterfaceId** (*string*) – [REQUIRED]

ID of the virtual interface.

Example: dxvif-123dfg56

Default: None

Return type dict**Returns****Response Syntax**

```
{
    'virtualInterfaceState': 'confirming'|'verifying'|'pending'|'available'|'delet
}
```

Response Structure

- (*dict*) –

The response received when DeleteVirtualInterface is called.

- **virtualInterfaceState** (*string*) – State of the virtual interface.
 - * **Confirming** : The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.
 - * **Verifying** : This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.
 - * **Pending** : A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.
 - * **Available** : A virtual interface that is able to forward traffic.
 - * **Deleting** : A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.
 - * **Deleted** : A virtual interface that cannot forward traffic.
 - * **Rejected** : The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the ‘Confirming’ state is deleted by the virtual interface owner, the virtual interface will enter the ‘Rejected’ state.

describe_connections (***kwargs*)

Displays all connections in this region.

If a connection ID is provided, the call returns only that particular connection.

Request Syntax

```
response = client.describe_connections(
    connectionId='string'
)
```

Parameters **connectionId** (*string*) – ID of the connection.

Example: dxcon-fg5678gh

Default: None

Return type dict**Returns****Response Syntax**

```
{
    'connections': [
        {
            'ownerAccount': 'string',
            'connectionId': 'string',
```

```
        'connectionName': 'string',
        'connectionState': 'ordering'|'requested'|'pending'|'available'|'down',
        'region': 'string',
        'location': 'string',
        'bandwidth': 'string',
        'vlan': 123,
        'partnerName': 'string'
    },
]
}
```

Response Structure

- (*dict*) –

A structure containing a list of connections.

- **connections** (*list*) –

A list of connections.

- * (*dict*) –

A connection represents the physical network connection between the AWS Direct Connect location and the customer.

- **ownerAccount** (*string*) –
- **connectionId** (*string*) –

ID of the connection.

Example: dxcon-fg5678gh

Default: None

- **connectionName** (*string*) –

The name of the connection.

Example: “My Connection to AWS “

Default: None

- **connectionState** (*string*) – State of the connection.
- **Ordering** : The initial state of a hosted connection provisioned on an interconnect. The connection stays in the ordering state until the owner of the hosted connection confirms or declines the connection order.
- **Requested** : The initial state of a standard connection. The connection stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.
- **Pending** : The connection has been approved, and is being initialized.
- **Available** : The network link is up, and the connection is ready for use.
- **Down** : The network link is down.
- **Deleted** : The connection has been deleted.
- **Rejected** : A hosted connection in the ‘Ordering’ state will enter the ‘Rejected’ state if it is deleted by the end customer.
- **region** (*string*) –

The AWS region where the connection is located.

Example: us-east-1

Default: None

- **location** (*string*) –
Where the connection is located.
Example: EqSV5
Default: None
- **bandwidth** (*string*) –
Bandwidth of the connection.
Example: 1Gbps (for regular connections), or 500Mbps (for hosted connections)
Default: None
- **vlan** (*integer*) –
The VLAN ID.
Example: 101
- **partnerName** (*string*) –

describe_connections_on_interconnect (**kwargs)

Return a list of connections that have been provisioned on the given interconnect.

Request Syntax

```
response = client.describe_connections_on_interconnect(
    interconnectId='string'
)
```

Parameters **interconnectId** (*string*) – **[REQUIRED]**

ID of the interconnect on which a list of connection is provisioned.

Example: dxcon-abc123

Default: None

Return type dict

Returns

Response Syntax

```
{
    'connections': [
        {
            'ownerAccount': 'string',
            'connectionId': 'string',
            'connectionName': 'string',
            'connectionState': 'ordering'|'requested'|'pending'|'available'|'down',
            'region': 'string',
            'location': 'string',
            'bandwidth': 'string',
            'vlan': 123,
            'partnerName': 'string'
        },
    ]
}
```

Response Structure

- (*dict*) –
A structure containing a list of connections.

– **connections** (*list*) –

A list of connections.

* (*dict*) –

A connection represents the physical network connection between the AWS Direct Connect location and the customer.

- **ownerAccount** (*string*) –

- **connectionId** (*string*) –

ID of the connection.

Example: dxcon-fg5678gh

Default: None

- **connectionName** (*string*) –

The name of the connection.

Example: “My Connection to AWS “

Default: None

- **connectionState** (*string*) – State of the connection.

- **Ordering** : The initial state of a hosted connection provisioned on an interconnect. The connection stays in the ordering state until the owner of the hosted connection confirms or declines the connection order.

- **Requested** : The initial state of a standard connection. The connection stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.

- **Pending** : The connection has been approved, and is being initialized.

- **Available** : The network link is up, and the connection is ready for use.

- **Down** : The network link is down.

- **Deleted** : The connection has been deleted.

- **Rejected** : A hosted connection in the ‘Ordering’ state will enter the ‘Rejected’ state if it is deleted by the end customer.

- **region** (*string*) –

The AWS region where the connection is located.

Example: us-east-1

Default: None

- **location** (*string*) –

Where the connection is located.

Example: EqSV5

Default: None

- **bandwidth** (*string*) –

Bandwidth of the connection.

Example: 1Gbps (for regular connections), or 500Mbps (for hosted connections)

Default: None

- **vlan** (*integer*) –

The VLAN ID.

Example: 101

- **partnerName** (*string*) –

describe_interconnects (***kwargs*)

Returns a list of interconnects owned by the AWS account.

If an interconnect ID is provided, it will only return this particular interconnect.

Request Syntax

```
response = client.describe_interconnects(
    interconnectId='string'
)
```

Parameters **interconnectId** (*string*) – The ID of the interconnect.

Example: dxcon-abc123

Return type dict

Returns

Response Syntax

```
{
    'interconnects': [
        {
            'interconnectId': 'string',
            'interconnectName': 'string',
            'interconnectState': 'requested'|'pending'|'available'|'down'|'deleting',
            'region': 'string',
            'location': 'string',
            'bandwidth': 'string'
        },
    ]
}
```

Response Structure

- (*dict*) –

A structure containing a list of interconnects.

- **interconnects** (*list*) –

A list of interconnects.

- * (*dict*) –

An interconnect is a connection that can host other connections.

Like a standard AWS Direct Connect connection, an interconnect represents the physical connection between an AWS Direct Connect partner's network and a specific Direct Connect location. An AWS Direct Connect partner who owns an interconnect can provision hosted connections on the interconnect for their end customers, thereby providing the end customers with connectivity to AWS services.

The resources of the interconnect, including bandwidth and VLAN numbers, are shared by all of the hosted connections on the interconnect, and the owner of the interconnect determines how these resources are assigned.

- **interconnectId** (*string*) –

The ID of the interconnect.

Example: dxcon-abc123

- **interconnectName** (*string*) –

The name of the interconnect.

Example: “1G Interconnect to AWS “

- **interconnectState** (*string*) – State of the interconnect.
- **Requested** : The initial state of an interconnect. The interconnect stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.
- **Pending** : The interconnect has been approved, and is being initialized.
- **Available** : The network link is up, and the interconnect is ready for use.
- **Down** : The network link is down.
- **Deleted** : The interconnect has been deleted.
- **region** (*string*) –

The AWS region where the connection is located.

Example: us-east-1

Default: None

- **location** (*string*) –

Where the connection is located.

Example: EqSV5

Default: None

- **bandwidth** (*string*) –

Bandwidth of the connection.

Example: 1Gbps

Default: None

describe_locations()

Returns the list of AWS Direct Connect locations in the current AWS region. These are the locations that may be selected when calling `CreateConnection` or `CreateInterconnect`.

Request Syntax

```
response = client.describe_locations()
```

Return type dict

Returns

Response Syntax

```
{
    'locations': [
        {
            'locationCode': 'string',
            'locationName': 'string'
        },
    ]
}
```

Response Structure

- (*dict*) –

– **locations** (*list*) –

* (*dict*) –

An AWS Direct Connect location where connections and interconnects can be requested.

· **locationCode** (*string*) –

The code used to indicate the AWS Direct Connect location.

· **locationName** (*string*) –

The name of the AWS Direct Connect location. The name includes the colocation partner name and the physical site of the lit building.

describe_virtual_gateways ()

Returns a list of virtual private gateways owned by the AWS account.

You can create one or more AWS Direct Connect private virtual interfaces linking to a virtual private gateway. A virtual private gateway can be managed via Amazon Virtual Private Cloud (VPC) console or the [EC2 CreateVpnGateway](#) action.

Request Syntax

```
response = client.describe_virtual_gateways()
```

Return type dict

Returns

Response Syntax

```
{
  'virtualGateways': [
    {
      'virtualGatewayId': 'string',
      'virtualGatewayState': 'string'
    },
  ]
}
```

Response Structure

• (*dict*) –

A structure containing a list of virtual private gateways.

– **virtualGateways** (*list*) –

A list of virtual private gateways.

* (*dict*) –

You can create one or more AWS Direct Connect private virtual interfaces linking to your virtual private gateway.

Virtual private gateways can be managed using the Amazon Virtual Private Cloud (Amazon VPC) console or the [Amazon EC2 CreateVpnGateway](#) action .

· **virtualGatewayId** (*string*) –

The ID of the virtual private gateway to a VPC. This only applies to private virtual interfaces.

Example: vgw-123er56

- **virtualGatewayState** (*string*) – State of the virtual private gateway.
- **Pending** : This is the initial state after calling *CreateVpnGateway* .
- **Available** : Ready for use by a private virtual interface.
- **Deleting** : This is the initial state after calling *DeleteVpnGateway* .
- **Deleted** : In this state, a private virtual interface is unable to send traffic over this gateway.

describe_virtual_interfaces (***kwargs*)

Displays all virtual interfaces for an AWS account. Virtual interfaces deleted fewer than 15 minutes before *DescribeVirtualInterfaces* is called are also returned. If a connection ID is included then only virtual interfaces associated with this connection will be returned. If a virtual interface ID is included then only a single virtual interface will be returned.

A virtual interface (VLAN) transmits the traffic between the AWS Direct Connect location and the customer.

If a connection ID is provided, only virtual interfaces provisioned on the specified connection will be returned. If a virtual interface ID is provided, only this particular virtual interface will be returned.

Request Syntax

```
response = client.describe_virtual_interfaces(
    connectionId='string',
    virtualInterfaceId='string'
)
```

Parameters

- **connectionId** (*string*) – ID of the connection.
Example: dxcon-fg5678gh
Default: None
- **virtualInterfaceId** (*string*) – ID of the virtual interface.
Example: dxvif-123dfg56
Default: None

Return type dict

Returns

Response Syntax

```
{
  'virtualInterfaces': [
    {
      'ownerAccount': 'string',
      'virtualInterfaceId': 'string',
      'location': 'string',
      'connectionId': 'string',
      'virtualInterfaceType': 'string',
      'virtualInterfaceName': 'string',
      'vlan': 123,
      'asn': 123,
      'authKey': 'string',
      'amazonAddress': 'string',
      'customerAddress': 'string',
      'virtualInterfaceState': 'confirming'|'verifying'|'pending'|'available'
```

```

        'customerRouterConfig': 'string',
        'virtualGatewayId': 'string',
        'routeFilterPrefixes': [
            {
                'cidr': 'string'
            },
        ]
    },
]
}

```

Response Structure

- (*dict*) –

A structure containing a list of virtual interfaces.

- **virtualInterfaces** (*list*) –

A list of virtual interfaces.

- * (*dict*) –

A virtual interface (VLAN) transmits the traffic between the AWS Direct Connect location and the customer.

- **ownerAccount** (*string*) –
- **virtualInterfaceId** (*string*) –

ID of the virtual interface.

Example: dxvif-123dfg56

Default: None

- **location** (*string*) –

Where the connection is located.

Example: EqSV5

Default: None

- **connectionId** (*string*) –

ID of the connection.

Example: dxcon-fg5678gh

Default: None

- **virtualInterfaceType** (*string*) –

The type of virtual interface.

Example: private (Amazon VPC) or public (Amazon S3, Amazon DynamoDB, and so on.)

- **virtualInterfaceName** (*string*) –

The name of the virtual interface assigned by the customer.

Example: “My VPC”

- **vlan** (*integer*) –

The VLAN ID.

Example: 101

- **asn** (*integer*) –

Autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.

Example: 65000

- **authKey** (*string*) –

Authentication key for BGP configuration.

Example: asdf34example

- **amazonAddress** (*string*) –

IP address assigned to the Amazon interface.

Example: 192.168.1.1/30

- **customerAddress** (*string*) –

IP address assigned to the customer interface.

Example: 192.168.1.2/30

- **virtualInterfaceState** (*string*) – State of the virtual interface.
- **Confirming** : The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.
- **Verifying** : This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.
- **Pending** : A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.
- **Available** : A virtual interface that is able to forward traffic.
- **Deleting** : A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.
- **Deleted** : A virtual interface that cannot forward traffic.
- **Rejected** : The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the ‘Confirming’ state is deleted by the virtual interface owner, the virtual interface will enter the ‘Rejected’ state.
- **customerRouterConfig** (*string*) –

Information for generating the customer router configuration.

- **virtualGatewayId** (*string*) –

The ID of the virtual private gateway to a VPC. This only applies to private virtual interfaces.

Example: vgw-123er56

- **routeFilterPrefixes** (*list*) –

A list of routes to be advertised to the AWS network in this region (public virtual interface) or your VPC (private virtual interface).

- (*dict*) –

A route filter prefix that the customer can advertise through Border Gateway Protocol (BGP) over a public virtual interface.

- **cidr** (*string*) –

CIDR notation for the advertised route. Multiple routes are separated by commas.

Example: 10.10.10.0/24,10.10.11.0/24

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to *ClientMethod*.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_waiter (*waiter_name*)

DirectoryService

Table of Contents

- [DirectoryService](#)
 - [Client](#)

Client

class `DirectoryService.Client`

A low-level client representing AWS Directory Service:

```
import boto3
```

```
client = boto3.client('ds')
```

These are the available methods:

- `can_paginate()`
- `connect_directory()`
- `create_alias()`
- `create_computer()`
- `create_directory()`
- `create_snapshot()`

- `delete_directory()`
- `delete_snapshot()`
- `describe_directories()`
- `describe_snapshots()`
- `disable_radius()`
- `disable_sso()`
- `enable_radius()`
- `enable_sso()`
- `generate_presigned_url()`
- `get_directory_limits()`
- `get_paginator()`
- `get_snapshot_limits()`
- `get_waiter()`
- `restore_from_snapshot()`
- `update_radius()`

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

connect_directory (***kwargs*)

Creates an AD Connector to connect an on-premises directory.

Request Syntax

```
response = client.connect_directory(
    Name='string',
    ShortName='string',
    Password='string',
    Description='string',
    Size='Small'|'Large',
    ConnectSettings={
        'VpcId': 'string',
        'SubnetIds': [
            'string',
        ],
        'CustomerDnsIps': [
            'string',
        ],
        'CustomerUserName': 'string'
    }
)
```

Parameters

- **Name** (*string*) – **[REQUIRED]**

The fully-qualified name of the on-premises directory, such as `corp.example.com`.

- **ShortName** (*string*) – The NetBIOS name of the on-premises directory, such as `CORP`.
- **Password** (*string*) – **[REQUIRED]**

The password for the on-premises user account.

- **Description** (*string*) – A textual description for the directory.
- **Size** (*string*) – [REQUIRED]

The size of the directory.

- **ConnectSettings** (*dict*) – [REQUIRED]

A DirectoryConnectSettings object that contains additional information for the operation.

- **VpcId** (*string*) – [REQUIRED]

The identifier of the VPC that the AD Connector is created in.

- **SubnetIds** (*list*) – [REQUIRED]

A list of subnet identifiers in the VPC that the AD Connector is created in.

* (*string*) –

- **CustomerDnsIps** (*list*) – [REQUIRED]

A list of one or more IP addresses of DNS servers or domain controllers in the on-premises directory.

* (*string*) –

- **CustomerUserName** (*string*) – [REQUIRED]

The username of an account in the on-premises directory that is used to connect to the directory. This account must have the following privileges:

- * Read users and groups
- * Create computer objects
- * Join computers to the domain

Return type dict

Returns

Response Syntax

```
{
    'DirectoryId': 'string'
}
```

Response Structure

- (*dict*) –

Contains the results of the ConnectDirectory operation.

- **DirectoryId** (*string*) –

The identifier of the new directory.

create_alias (***kwargs*)

Creates an alias for a directory and assigns the alias to the directory. The alias is used to construct the access URL for the directory, such as `http://alias.awsapps.com`.

Warning: After an alias has been created, it cannot be deleted or reused, so this operation should only be used when absolutely necessary.

Request Syntax

```
response = client.create_alias(
    DirectoryId='string',
    Alias='string'
)
```

Parameters

- **DirectoryId** (*string*) – [REQUIRED]

The identifier of the directory to create the alias for.

- **Alias** (*string*) – [REQUIRED]

The requested alias.

The alias must be unique amongst all aliases in AWS. This operation will throw an `EntityAlreadyExistsException` if this alias already exists.

Return type dict

Returns

Response Syntax

```
{
    'DirectoryId': 'string',
    'Alias': 'string'
}
```

Response Structure

- (*dict*) –

Contains the results of the CreateAlias operation.

- **DirectoryId** (*string*) –

The identifier of the directory.

- **Alias** (*string*) –

The alias for the directory.

create_computer (***kwargs*)

Creates a computer account in the specified directory, and joins the computer to the directory.

Request Syntax

```
response = client.create_computer(
    DirectoryId='string',
    ComputerName='string',
    Password='string',
    OrganizationalUnitDistinguishedName='string',
    ComputerAttributes=[
        {
            'Name': 'string',
            'Value': 'string'
        },
    ],
)
```

Parameters

- **DirectoryId** (*string*) – [REQUIRED]

The identifier of the directory to create the computer account in.

- **ComputerName** (*string*) – [REQUIRED]

The name of the computer account.

- **Password** (*string*) – [REQUIRED]

A one-time password that is used to join the computer to the directory. You should generate a random, strong password to use for this parameter.

- **OrganizationalUnitDistinguishedName** (*string*) – The fully-qualified distinguished name of the organizational unit to place the computer account in.

- **ComputerAttributes** (*list*) – An array of Attribute objects that contain any LDAP attributes to apply to the computer account.

– (*dict*) –

Represents a named directory attribute.

* **Name** (*string*) –

The name of the attribute.

* **Value** (*string*) –

The value of the attribute.

Return type dict

Returns

Response Syntax

```
{
  'Computer': {
    'ComputerId': 'string',
    'ComputerName': 'string',
    'ComputerAttributes': [
      {
        'Name': 'string',
        'Value': 'string'
      },
    ]
  }
}
```

Response Structure

- (*dict*) –

Contains the results for the CreateComputer operation.

– **Computer** (*dict*) –

A Computer object the represents the computer account.

* **ComputerId** (*string*) –

The identifier of the computer.

* **ComputerName** (*string*) –

The computer name.

* **ComputerAttributes** (*list*) –

An array of Attribute objects that contain the LDAP attributes that belong to the computer account.

· (*dict*) –

Represents a named directory attribute.

· **Name** (*string*) –

The name of the attribute.

· **Value** (*string*) –

The value of the attribute.

create_directory (***kwargs*)

Creates a Simple AD directory.

Request Syntax

```
response = client.create_directory(
    Name='string',
    ShortName='string',
    Password='string',
    Description='string',
    Size='Small'|'Large',
    VpcSettings={
        'VpcId': 'string',
        'SubnetIds': [
            'string',
        ]
    }
)
```

Parameters

- **Name** (*string*) – **[REQUIRED]**

The fully qualified name for the directory, such as `corp.example.com`.

- **ShortName** (*string*) – The short name of the directory, such as `CORP`.
- **Password** (*string*) – **[REQUIRED]**

The password for the directory administrator. The directory creation process creates a directory administrator account with the username `Administrator` and this password.

- **Description** (*string*) – A textual description for the directory.
- **Size** (*string*) – **[REQUIRED]**

The size of the directory.

- **VpcSettings** (*dict*) – A `DirectoryVpcSettings` object that contains additional information for the operation.

- **VpcId** (*string*) – **[REQUIRED]**

The identifier of the VPC to create the Simple AD directory in.

- **SubnetIds** (*list*) – **[REQUIRED]**

The identifiers of the subnets for the directory servers. The two subnets must be in different Availability Zones. AWS Directory Service creates a directory server and a DNS server in each of these subnets.

* (*string*) –

Return type `dict`

Returns

Response Syntax

```
{
    'DirectoryId': 'string'
}
```

Response Structure

- (*dict*) –

Contains the results of the `CreateDirectory` operation.

- **DirectoryId** (*string*) –

The identifier of the directory that was created.

create_snapshot (***kwargs*)

Creates a snapshot of an existing directory.

You cannot take snapshots of extended or connected directories.

Request Syntax

```
response = client.create_snapshot(
    DirectoryId='string',
    Name='string'
)
```

Parameters

- **DirectoryId** (*string*) – [REQUIRED]
The identifier of the directory to take a snapshot of.
- **Name** (*string*) – The descriptive name to apply to the snapshot.

Return type dict

Returns

Response Syntax

```
{
    'SnapshotId': 'string'
}
```

Response Structure

- (*dict*) –
Contains the results of the CreateSnapshot operation.
 - **SnapshotId** (*string*) –
The identifier of the snapshot that was created.

delete_directory (***kwargs*)

Deletes an AWS Directory Service directory.

Request Syntax

```
response = client.delete_directory(
    DirectoryId='string'
)
```

Parameters **DirectoryId** (*string*) – [REQUIRED]

The identifier of the directory to delete.

Return type dict

Returns

Response Syntax

```
{
    'DirectoryId': 'string'
}
```

Response Structure

- (*dict*) –
Contains the results of the DeleteDirectory operation.
 - **DirectoryId** (*string*) –
The directory identifier.

delete_snapshot (**kwargs)
Deletes a directory snapshot.

Request Syntax

```
response = client.delete_snapshot(  
    SnapshotId='string'  
)
```

Parameters **SnapshotId** (*string*) – [REQUIRED]

The identifier of the directory snapshot to be deleted.

Return type dict

Returns

Response Syntax

```
{  
    'SnapshotId': 'string'  
}
```

Response Structure

- (*dict*) –

Contains the results of the DeleteSnapshot operation.

- **SnapshotId** (*string*) –

The identifier of the directory snapshot that was deleted.

describe_directories (**kwargs)

Obtains information about the directories that belong to this account.

You can retrieve information about specific directories by passing the directory identifiers in the *DirectoryIds* parameter. Otherwise, all directories that belong to the current account are returned.

This operation supports pagination with the use of the *NextToken* request and response parameters. If more results are available, the *DescribeDirectoriesResult.NextToken* member contains a token that you pass in the next call to *DescribeDirectories* to retrieve the next set of items.

You can also specify a maximum number of return results with the *Limit* parameter.

Request Syntax

```
response = client.describe_directories(  
    DirectoryIds=[  
        'string',  
    ],  
    NextToken='string',  
    Limit=123  
)
```

Parameters

- **DirectoryIds** (*list*) – A list of identifiers of the directories to obtain the information for. If this member is null, all directories that belong to the current account are returned.

An empty list results in an *InvalidParameterException* being thrown.

- (*string*) –

- **NextToken** (*string*) – The *DescribeDirectoriesResult.NextToken* value from a previous call to *DescribeDirectories* . Pass null if this is the first call.

- **Limit** (*integer*) – The maximum number of items to return. If this value is zero, the maximum number of items is specified by the limitations of the operation.

Return type dict

Returns

Response Syntax

```
{
  'DirectoryDescriptions': [
    {
      'DirectoryId': 'string',
      'Name': 'string',
      'ShortName': 'string',
      'Size': 'Small'|'Large',
      'Alias': 'string',
      'AccessUrl': 'string',
      'Description': 'string',
      'DnsIpAddrs': [
        'string',
      ],
      'Stage': 'Requested'|'Creating'|'Created'|'Active'|'Inoperable'|'Impai
      'LaunchTime': datetime(2015, 1, 1),
      'StageLastUpdatedDateTime': datetime(2015, 1, 1),
      'Type': 'SimpleAD'|'ADConnector',
      'VpcSettings': {
        'VpcId': 'string',
        'SubnetIds': [
          'string',
        ],
        'SecurityGroupId': 'string',
        'AvailabilityZones': [
          'string',
        ]
      },
      'ConnectSettings': {
        'VpcId': 'string',
        'SubnetIds': [
          'string',
        ],
        'CustomerUserName': 'string',
        'SecurityGroupId': 'string',
        'AvailabilityZones': [
          'string',
        ],
        'ConnectIps': [
          'string',
        ]
      },
      'RadiusSettings': {
        'RadiusServers': [
          'string',
        ],
        'RadiusPort': 123,
        'RadiusTimeout': 123,
        'RadiusRetries': 123,
        'SharedSecret': 'string',
        'AuthenticationProtocol': 'PAP'|'CHAP'|'MS-CHAPv1'|'MS-CHAPv2',
        'DisplayLabel': 'string',
      },
    },
  ],
}
```

```
        'UseSameUsername': True|False
    },
    'RadiusStatus': 'Creating'|'Completed'|'Failed',
    'StageReason': 'string',
    'SsoEnabled': True|False
},
],
'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the results of the DescribeDirectories operation.

- **DirectoryDescriptions** (*list*) –

The list of DirectoryDescription objects that were retrieved.

It is possible that this list contains less than the number of items specified in the *Limit* member of the request. This occurs if there are less than the requested number of items left to retrieve, or if the limitations of the operation have been exceeded.

- * (*dict*) –

Contains information about an AWS Directory Service directory.

- **DirectoryId** (*string*) –

The directory identifier.

- **Name** (*string*) –

The fully-qualified name of the directory.

- **ShortName** (*string*) –

The short name of the directory.

- **Size** (*string*) –

The directory size.

- **Alias** (*string*) –

The alias for the directory.

- **AccessUrl** (*string*) –

The access URL for the directory, such as `http://alias.awsapps.com`.

- **Description** (*string*) –

The textual description for the directory.

- **DnsIpAddrs** (*list*) –

The IP addresses of the DNS servers for the directory. For a Simple AD directory, these are the IP addresses of the Simple AD directory servers. For an AD Connector directory, these are the IP addresses of the DNS servers or domain controllers in the on-premises directory that the AD Connector is connected to.

- (*string*) –

- **Stage** (*string*) –

The current stage of the directory.

- **LaunchTime** (*datetime*) –

- Specifies when the directory was created.
- **StageLastUpdatedDateTime** (*datetime*) –
The date and time that the stage was last updated.
- **Type** (*string*) –
The directory size.
- **VpcSettings** (*dict*) –
A DirectoryVpcSettingsDescription object that contains additional information about a Simple AD directory. This member is only present if the directory is a Simple AD directory.
- **VpcId** (*string*) –
The identifier of the VPC that the directory is in.
- **SubnetIds** (*list*) –
The identifiers of the subnets for the directory servers.
- (*string*) –
- **SecurityGroupId** (*string*) –
The security group identifier for the directory.
- **AvailabilityZones** (*list*) –
The list of Availability Zones that the directory is in.
- (*string*) –
- **ConnectSettings** (*dict*) –
A DirectoryConnectSettingsDescription object that contains additional information about an AD Connector directory. This member is only present if the directory is an AD Connector directory.
- **VpcId** (*string*) –
The identifier of the VPC that the AD Connector is in.
- **SubnetIds** (*list*) –
A list of subnet identifiers in the VPC that the AD connector is in.
- (*string*) –
- **CustomerUserName** (*string*) –
The username of the service account in the on-premises directory.
- **SecurityGroupId** (*string*) –
The security group identifier for the AD Connector directory.
- **AvailabilityZones** (*list*) –
A list of the Availability Zones that the directory is in.
- (*string*) –
- **ConnectIps** (*list*) –
The IP addresses of the AD Connector servers.
- (*string*) –
- **RadiusSettings** (*dict*) –
A RadiusSettings object that contains information about the RADIUS server configured for this directory.
- **RadiusServers** (*list*) –

An array of strings that contains the IP addresses of the RADIUS server endpoints, or the IP addresses of your RADIUS server load balancer.

- *(string)* –
- **RadiusPort** (*integer*) –

The port that your RADIUS server is using for communications. Your on-premises network must allow inbound traffic over this port from the AWS Directory Service servers.

- **RadiusTimeout** (*integer*) –

The amount of time, in seconds, to wait for the RADIUS server to respond.

- **RadiusRetries** (*integer*) –

The maximum number of times that communication with the RADIUS server is attempted.

- **SharedSecret** (*string*) –

The shared secret code that was specified when your RADIUS endpoints were created.

- **AuthenticationProtocol** (*string*) –

The protocol specified for your RADIUS endpoints.

- **DisplayLabel** (*string*) –

Not currently used.

- **UseSameUsername** (*boolean*) –

Not currently used.

- **RadiusStatus** (*string*) –

The status of the RADIUS MFA server connection.

- **StageReason** (*string*) –

Additional information about the directory stage.

- **SsoEnabled** (*boolean*) –

Indicates if single-sign on is enabled for the directory. For more information, see `EnableSso` and `DisableSso`.

- **NextToken** (*string*) –

If not null, more results are available. Pass this value for the *NextToken* parameter in a subsequent call to `DescribeDirectories` to retrieve the next set of items.

describe_snapshots (***kwargs*)

Obtains information about the directory snapshots that belong to this account.

This operation supports pagination with the use of the *NextToken* request and response parameters. If more results are available, the *DescribeSnapshots.NextToken* member contains a token that you pass in the next call to `DescribeSnapshots` to retrieve the next set of items.

You can also specify a maximum number of return results with the *Limit* parameter.

Request Syntax

```
response = client.describe_snapshots(  
    DirectoryId='string',  
    SnapshotIds=[  
        'string',
```

```

],
NextToken='string',
Limit=123
)

```

Parameters

- **DirectoryId** (*string*) – The identifier of the directory to retrieve snapshot information for.
- **SnapshotIds** (*list*) – A list of identifiers of the snapshots to obtain the information for. If this member is null or empty, all snapshots are returned using the *Limit* and *NextToken* members.
 - (*string*) –
- **NextToken** (*string*) – The *DescribeSnapshotsResult.NextToken* value from a previous call to *DescribeSnapshots*. Pass null if this is the first call.
- **Limit** (*integer*) – The maximum number of objects to return.

Return type dict

Returns

Response Syntax

```

{
  'Snapshots': [
    {
      'DirectoryId': 'string',
      'SnapshotId': 'string',
      'Type': 'Auto'|'Manual',
      'Name': 'string',
      'Status': 'Creating'|'Completed'|'Failed',
      'StartTime': datetime(2015, 1, 1)
    },
  ],
  'NextToken': 'string'
}

```

Response Structure

- (*dict*) –

Contains the results of the *DescribeSnapshots* operation.

- **Snapshots** (*list*) –

The list of Snapshot objects that were retrieved.

It is possible that this list contains less than the number of items specified in the *Limit* member of the request. This occurs if there are less than the requested number of items left to retrieve, or if the limitations of the operation have been exceeded.

- * (*dict*) –

Describes a directory snapshot.

- **DirectoryId** (*string*) –

The directory identifier.

- **SnapshotId** (*string*) –

The snapshot identifier.

- **Type** (*string*) –

The snapshot type.

- **Name** (*string*) –
The descriptive name of the snapshot.
- **Status** (*string*) –
The snapshot status.
- **StartTime** (*datetime*) –
The date and time that the snapshot was taken.
- **NextToken** (*string*) –
If not null, more results are available. Pass this value in the *NextToken* member of a subsequent call to *DescribeSnapshots*.

disable_radius (***kwargs*)

Disables multi-factor authentication (MFA) with Remote Authentication Dial In User Service (RADIUS) for an AD Connector directory.

Request Syntax

```
response = client.disable_radius(  
    DirectoryId='string'  
)
```

Parameters **DirectoryId** (*string*) – [REQUIRED]

The identifier of the directory to disable MFA for.

Return type dict

Returns

Response Syntax

```
{}
```

Response Structure

- (*dict*) –

Contains the results of the *DisableRadius* operation.

disable_sso (***kwargs*)

Disables single-sign on for a directory.

Request Syntax

```
response = client.disable_sso(  
    DirectoryId='string',  
    UserName='string',  
    Password='string'  
)
```

Parameters

- **DirectoryId** (*string*) – [REQUIRED]

The identifier of the directory to disable single-sign on for.

- **UserName** (*string*) – The username of an alternate account to use to disable single-sign on. This is only used for AD Connector directories. This account must have privileges to remove a service principle name.

If the AD Connector service account does not have privileges to remove a service principle name, you can specify an alternate account with the *UserName* and *Password* parameters. These credentials are only used to disable single sign-on

and are not stored by the service. The AD Connector service account is not changed.

- **Password** (*string*) – The password of an alternate account to use to disable single-sign on. This is only used for AD Connector directories. See the *UserName* parameter for more information.

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –
Contains the results of the DisableSso operation.

enable_radius (**kwargs)

Enables multi-factor authentication (MFA) with Remote Authentication Dial In User Service (RADIUS) for an AD Connector directory.

Request Syntax

```
response = client.enable_radius(
    DirectoryId='string',
    RadiusSettings={
        'RadiusServers': [
            'string',
        ],
        'RadiusPort': 123,
        'RadiusTimeout': 123,
        'RadiusRetries': 123,
        'SharedSecret': 'string',
        'AuthenticationProtocol': 'PAP' | 'CHAP' | 'MS-CHAPv1' | 'MS-CHAPv2',
        'DisplayLabel': 'string',
        'UseSameUsername': True|False
    }
)
```

Parameters

- **DirectoryId** (*string*) – [REQUIRED]

The identifier of the directory to enable MFA for.

- **RadiusSettings** (*dict*) – [REQUIRED]

A RadiusSettings object that contains information about the RADIUS server.

- **RadiusServers** (*list*) –

An array of strings that contains the IP addresses of the RADIUS server endpoints, or the IP addresses of your RADIUS server load balancer.

* (*string*) –

- **RadiusPort** (*integer*) –

The port that your RADIUS server is using for communications. Your on-premises network must allow inbound traffic over this port from the AWS Directory Service servers.

- **RadiusTimeout** (*integer*) –

The amount of time, in seconds, to wait for the RADIUS server to respond.

- **RadiusRetries** (*integer*) –

The maximum number of times that communication with the RADIUS server is attempted.

- **SharedSecret** (*string*) –

The shared secret code that was specified when your RADIUS endpoints were created.

- **AuthenticationProtocol** (*string*) –

The protocol specified for your RADIUS endpoints.

- **DisplayLabel** (*string*) –

Not currently used.

- **UseSameUsername** (*boolean*) –

Not currently used.

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –

Contains the results of the EnableRadius operation.

enable_sso (***kwargs*)

Enables single-sign on for a directory.

Request Syntax

```
response = client.enable_sso(  
    DirectoryId='string',  
    UserName='string',  
    Password='string'  
)
```

Parameters

- **DirectoryId** (*string*) – [REQUIRED]

The identifier of the directory to enable single-sign on for.

- **UserName** (*string*) – The username of an alternate account to use to enable single-sign on. This is only used for AD Connector directories. This account must have privileges to add a service principle name.

If the AD Connector service account does not have privileges to add a service principle name, you can specify an alternate account with the *UserName* and *Password* parameters. These credentials are only used to enable single sign-on and are not stored by the service. The AD Connector service account is not changed.

- **Password** (*string*) – The password of an alternate account to use to enable single-sign on. This is only used for AD Connector directories. See the *UserName* parameter for more information.

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –

Contains the results of the EnableSso operation.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to *ClientMethod*.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_directory_limits ()

Obtains directory limit information for the current region.

Request Syntax

```
response = client.get_directory_limits()
```

Return type dict

Returns

Response Syntax

```
{
  'DirectoryLimits': {
    'CloudOnlyDirectoriesLimit': 123,
    'CloudOnlyDirectoriesCurrentCount': 123,
    'CloudOnlyDirectoriesLimitReached': True|False,
    'ConnectedDirectoriesLimit': 123,
    'ConnectedDirectoriesCurrentCount': 123,
    'ConnectedDirectoriesLimitReached': True|False
  }
}
```

Response Structure

- (*dict*) –

Contains the results of the GetDirectoryLimits operation.

- **DirectoryLimits** (*dict*) –

A DirectoryLimits object that contains the directory limits for the current region.

- * **CloudOnlyDirectoriesLimit** (*integer*) –

The maximum number of cloud directories allowed in the region.

- * **CloudOnlyDirectoriesCurrentCount** (*integer*) –

The current number of cloud directories in the region.

- * **CloudOnlyDirectoriesLimitReached** (*boolean*) –

Indicates if the cloud directory limit has been reached.

- * **ConnectedDirectoriesLimit** (*integer*) –

The maximum number of connected directories allowed in the region.

- * **ConnectedDirectoriesCurrentCount** (*integer*) –

The current number of connected directories in the region.

- * **ConnectedDirectoriesLimitReached** (*boolean*) –

Indicates if the connected directory limit has been reached.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_snapshot_limits (***kwargs*)

Obtains the manual snapshot limits for a directory.

Request Syntax

```
response = client.get_snapshot_limits(
    DirectoryId='string'
)
```

Parameters **DirectoryId** (*string*) – [REQUIRED]

Contains the identifier of the directory to obtain the limits for.

Return type `dict`

Returns

Response Syntax

```
{
  'SnapshotLimits': {
    'ManualSnapshotsLimit': 123,
    'ManualSnapshotsCurrentCount': 123,
    'ManualSnapshotsLimitReached': True|False
  }
}
```

Response Structure

- (*dict*) –

Contains the results of the `GetSnapshotLimits` operation.

- **SnapshotLimits** (*dict*) –

A `SnapshotLimits` object that contains the manual snapshot limits for the specified directory.

- * **ManualSnapshotsLimit** (*integer*) –

The maximum number of manual snapshots allowed.

- * **ManualSnapshotsCurrentCount** (*integer*) –
The current number of manual snapshots of the directory.
- * **ManualSnapshotsLimitReached** (*boolean*) –
Indicates if the manual snapshot limit has been reached.

get_waiter (*waiter_name*)

restore_from_snapshot (**kwargs)

Restores a directory using an existing directory snapshot.

When you restore a directory from a snapshot, any changes made to the directory after the snapshot date are overwritten.

This action returns as soon as the restore operation is initiated. You can monitor the progress of the restore operation by calling the DescribeDirectories operation with the directory identifier. When the **DirectoryDescription.Stage** value changes to `Active`, the restore operation is complete.

Request Syntax

```
response = client.restore_from_snapshot(
    SnapshotId='string'
)
```

Parameters **SnapshotId** (*string*) – [REQUIRED]

The identifier of the snapshot to restore from.

Return type dict

Returns

Response Syntax

```
{}
```

Response Structure

- (*dict*) –

Contains the results of the RestoreFromSnapshot operation.

update_radius (**kwargs)

Updates the Remote Authentication Dial In User Service (RADIUS) server information for an AD Connector directory.

Request Syntax

```
response = client.update_radius(
    DirectoryId='string',
    RadiusSettings={
        'RadiusServers': [
            'string',
        ],
        'RadiusPort': 123,
        'RadiusTimeout': 123,
        'RadiusRetries': 123,
        'SharedSecret': 'string',
        'AuthenticationProtocol': 'PAP' | 'CHAP' | 'MS-CHAPv1' | 'MS-CHAPv2',
        'DisplayLabel': 'string',
        'UseSameUsername': True|False
    }
)
```

Parameters

- **DirectoryId** (*string*) – [REQUIRED]

The identifier of the directory to update the RADIUS server information for.

- **RadiusSettings** (*dict*) – [REQUIRED]

A RadiusSettings object that contains information about the RADIUS server.

- **RadiusServers** (*list*) –

An array of strings that contains the IP addresses of the RADIUS server endpoints, or the IP addresses of your RADIUS server load balancer.

* (*string*) –

- **RadiusPort** (*integer*) –

The port that your RADIUS server is using for communications. Your on-premises network must allow inbound traffic over this port from the AWS Directory Service servers.

- **RadiusTimeout** (*integer*) –

The amount of time, in seconds, to wait for the RADIUS server to respond.

- **RadiusRetries** (*integer*) –

The maximum number of times that communication with the RADIUS server is attempted.

- **SharedSecret** (*string*) –

The shared secret code that was specified when your RADIUS endpoints were created.

- **AuthenticationProtocol** (*string*) –

The protocol specified for your RADIUS endpoints.

- **DisplayLabel** (*string*) –

Not currently used.

- **UseSameUsername** (*boolean*) –

Not currently used.

Return type dict

Returns**Response Syntax**

```
{ }
```

Response Structure

- (*dict*) –

Contains the results of the UpdateRadius operation.

DynamoDB

Table of Contents

- **DynamoDB**
 - Client
 - Paginators
 - Waiters
 - Service Resource
 - Table

Client**class** `DynamoDB.Client`

A low-level client representing Amazon DynamoDB:

```
import boto3
```

```
client = boto3.client('dynamodb')
```

These are the available methods:

- `batch_get_item()`
- `batch_write_item()`
- `can_paginate()`
- `create_table()`
- `delete_item()`
- `delete_table()`
- `describe_table()`
- `generate_presigned_url()`
- `get_item()`
- `get_paginator()`
- `get_waiter()`
- `list_tables()`
- `put_item()`
- `query()`
- `scan()`
- `update_item()`
- `update_table()`

batch_get_item (***kwargs*)

The *BatchGetItem* operation returns the attributes of one or more items from one or more tables. You identify requested items by primary key.

A single operation can retrieve up to 16 MB of data, which can contain as many as 100 items. *BatchGetItem* will return a partial result if the response size limit is exceeded, the table's provisioned throughput is exceeded, or an internal processing failure occurs. If a partial result is returned, the operation returns a value for *UnprocessedKeys*. You can use this value to retry the operation starting with the next item to get.

Warning: If you request more than 100 items *BatchGetItem* will return a *ValidationException* with the message “Too many items requested for the BatchGetItem call”.

For example, if you ask to retrieve 100 items, but each individual item is 300 KB in size, the system returns 52 items (so as not to exceed the 16 MB limit). It also returns an appropriate *UnprocessedKeys* value so you can get the next page of results. If desired, your application can include its own logic to assemble the pages of results into one data set.

If *none* of the items can be processed due to insufficient provisioned throughput on all of the tables in the request, then *BatchGetItem* will return a *ProvisionedThroughputExceededException* . If *at least one* of the items is successfully processed, then *BatchGetItem* completes successfully, while returning the keys of the unread items in *UnprocessedKeys* .

Warning: If DynamoDB returns any unprocessed items, you should retry the batch operation on those items. However, *we strongly recommend that you use an exponential backoff algorithm* . If you retry the batch operation immediately, the underlying read or write requests can still fail due to throttling on the individual tables. If you delay the batch operation using exponential backoff, the individual requests in the batch are much more likely to succeed.

For more information, see [Batch Operations and Error Handling](#) in the *Amazon DynamoDB Developer Guide* .

By default, *BatchGetItem* performs eventually consistent reads on every table in the request. If you want strongly consistent reads instead, you can set *ConsistentRead* to `true` for any or all tables.

In order to minimize response latency, *BatchGetItem* retrieves items in parallel.

When designing your application, keep in mind that DynamoDB does not return attributes in any particular order. To help parse the response by item, include the primary key values for the items in your request in the *AttributesToGet* parameter.

If a requested item does not exist, it is not returned in the result. Requests for nonexistent items consume the minimum read capacity units according to the type of read. For more information, see [Capacity Units Calculations](#) in the *Amazon DynamoDB Developer Guide* .

Request Syntax

```
response = client.batch_get_item(
    RequestItems={
        'string': {
            'Keys': [
                {
                    'string': {
                        'S': 'string',
                        'N': 'string',
                        'B': b'bytes',
                        'SS': [
                            'string',
                        ],
                        'NS': [
                            'string',
                        ],
                        'BS': [
                            b'bytes',
                        ],
                        'M': {
                            'string': {'... recursive ...'}
                        },
                        'L': [
                            {'... recursive ...'},
                        ],
                        'NULL': True|False,
                        'BOOL': True|False
                    }
                },
            ],
        },
        'AttributesToGet': [
```

```

        'string',
    ],
    'ConsistentRead': True|False,
    'ProjectionExpression': 'string',
    'ExpressionAttributeNames': {
        'string': 'string'
    }
},
ReturnConsumedCapacity='INDEXES'|'TOTAL'|'NONE'
)

```

Parameters

- **RequestItems** (*dict*) – [REQUIRED]

A map of one or more table names and, for each table, a map that describes one or more items to retrieve from that table. Each table name can be used only once per *BatchGetItem* request.

Each element in the map of items to retrieve consists of the following:

- *ConsistentRead* - If `true`, a strongly consistent read is used; if `false` (the default), an eventually consistent read is used.
- *ExpressionAttributeNames* - One or more substitution tokens for attribute names in the *ProjectionExpression* parameter. The following are some use cases for using *ExpressionAttributeNames*:
 - * To access an attribute whose name conflicts with a DynamoDB reserved word.
 - To create a placeholder for repeating occurrences of an attribute name in an expression.
 - To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the `#` character in an expression to dereference an attribute name. For example, consider the following attribute name:

- `Percentile`

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see [Reserved Words](#) in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for *ExpressionAttributeNames*:

- `{"#P": "Percentile"}`

You could then use this substitution in an expression, as in this example:

- `#P = :val`

Note: Tokens that begin with the `:` character are *expression attribute values*, which are placeholders for the actual value at runtime.

For more information on expression attribute names, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

- *Keys* - An array of primary key attribute values that define specific items in the table. For each primary key, you must provide *all* of the key attributes. For example, with a hash type primary key, you only need to provide the hash attribute. For a hash-and-range type primary key, you must provide *both* the hash attribute and the range attribute.
- *ProjectionExpression* - A string that identifies one or more attributes to retrieve from the table. These attributes can include scalars, sets, or elements of a JSON document. The attributes in the expression must be separated by commas. If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not

appear in the result. For more information, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

- *AttributesToGet* - .. warning:: This is a legacy parameter, for backward compatibility. New applications should use *ProjectionExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception. This parameter allows you to retrieve attributes of type List or Map; however, it cannot retrieve individual elements within a List or a Map. The names of one or more attributes to retrieve. If no attribute names are provided, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result. Note that *AttributesToGet* has no effect on provisioned throughput consumption. DynamoDB determines capacity units consumed based on item size, not on the amount of data that is returned to an application.

- *(string)* -
 - * *(dict)* -

Represents a set of primary keys and, for each key, the attributes to retrieve from the table.

For each primary key, you must provide *all* of the key attributes. For example, with a hash type primary key, you only need to provide the hash attribute. For a hash-and-range type primary key, you must provide *both* the hash attribute and the range attribute.

- **Keys** (*list*) - **[REQUIRED]**

The primary key attribute values that define the items and the attributes associated with the items.

- *(dict)* -
- *(string)* -
- *(dict)* -

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) -

A String data type.

- **N** (*string*) -

A Number data type.

- **B** (*bytes*) -

A Binary data type.

- **SS** (*list*) -

A String Set data type.

- *(string)* -
- **NS** (*list*) -

A Number Set data type.

- *(string)* -
- **BS** (*list*) -

A Binary Set data type.

- *(bytes)* –
- **M** (*dict*) –

A Map of attribute values.

- *(string)* –
- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **AttributesToGet** (*list*) –

One or more attributes to retrieve from the table or index. If no attribute names are specified then all attributes will be returned. If any of the specified attributes are not found, they will not appear in the result.

- *(string)* –
- **ConsistentRead** (*boolean*) –

The consistency of a read operation. If set to `true`, then a strongly consistent read is used; otherwise, an eventually consistent read is used.

- **ProjectionExpression** (*string*) –

A string that identifies one or more attributes to retrieve from the table. These attributes can include scalars, sets, or elements of a JSON document. The attributes in the *ProjectionExpression* must be separated by commas.

If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

For more information, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

Note: *ProjectionExpression* replaces the legacy *AttributesToGet* parameter.

- **ExpressionAttributeNames** (*dict*) –

One or more substitution tokens for attribute names in an expression. The following are some use cases for using *ExpressionAttributeNames* :

- To access an attribute whose name conflicts with a DynamoDB reserved word.
- To create a placeholder for repeating occurrences of an attribute name in an expression.
- To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the # character in an expression to dereference an attribute name. For example, consider the following attribute name:

- **Percentile**

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see [Reserved Words](#) in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for *ExpressionAttributeNames* :

- { "#P": "Percentile" }

You could then use this substitution in an expression, as in this example:

- #P = :val
-

Note: Tokens that begin with the : character are *expression attribute values* , which are placeholders for the actual value at runtime.

For more information on expression attribute names, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide* .

- (*string*) –
- (*string*) –

- **ReturnConsumedCapacity** (*string*) – Determines the level of detail about provisioned throughput consumption that is returned in the response:
 - *INDEXES* - The response includes the aggregate *ConsumedCapacity* for the operation, together with *ConsumedCapacity* for each table and secondary index that was accessed. Note that some operations, such as *GetItem* and *BatchGetItem* , do not access any indexes at all. In these cases, specifying *INDEXES* will only return *ConsumedCapacity* information for table(s).
 - *TOTAL* - The response includes only the aggregate *ConsumedCapacity* for the operation.
 - *NONE* - No *ConsumedCapacity* details are included in the response.

Return type dict

Returns

Response Syntax

```
{
```



```

'Responses': {
  'string': [
    {
      'string': {
        'S': 'string',
        'N': 'string',
        'B': b'bytes',
        'SS': [
          'string',
        ],
        'NS': [
          'string',
        ],
        'BS': [
          b'bytes',
        ],
        'M': {
          'string': {'... recursive ...'}
        },
        'L': [
          {'... recursive ...'},
        ],
        'NULL': True|False,
        'BOOL': True|False
      }
    },
  ],
},
'UnprocessedKeys': {
  'string': {
    'Keys': [
      {
        'string': {
          'S': 'string',
          'N': 'string',
          'B': b'bytes',
          'SS': [
            'string',
          ],
          'NS': [
            'string',
          ],
          'BS': [
            b'bytes',
          ],
          'M': {
            'string': {'... recursive ...'}
          },
          'L': [
            {'... recursive ...'},
          ],
          'NULL': True|False,
          'BOOL': True|False
        }
      },
    ],
  },
  'AttributesToGet': [
    'string',
  ],
}

```

```
    ],
    'ConsistentRead': True|False,
    'ProjectionExpression': 'string',
    'ExpressionAttributeNames': {
        'string': 'string'
    }
}
},
'ConsumedCapacity': [
    {
        'TableName': 'string',
        'CapacityUnits': 123.0,
        'Table': {
            'CapacityUnits': 123.0
        },
        'LocalSecondaryIndexes': {
            'string': {
                'CapacityUnits': 123.0
            }
        },
        'GlobalSecondaryIndexes': {
            'string': {
                'CapacityUnits': 123.0
            }
        }
    },
],
]
```

Response Structure

- (*dict*) –

Represents the output of a *BatchGetItem* operation.

- **Responses** (*dict*) –

A map of table name to a list of items. Each object in *Responses* consists of a table name, along with a map of attribute data consisting of the data type and attribute value.

- * (*string*) –

- (*list*) –
- (*dict*) –
- (*string*) –
- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –
- **NS** (*list*) –

A Number Set data type.

- (*string*) –
- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –
- **M** (*dict*) –

A Map of attribute values.

- (*string*) –
- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

– **UnprocessedKeys** (*dict*) –

A map of tables and their respective keys that were not processed with the current response. The *UnprocessedKeys* value is in the same form as *RequestItems*, so the value can be provided directly to a subsequent *BatchGetItem* operation. For more information, see *RequestItems* in the Request Parameters section.

Each element consists of:

- * *Keys* - An array of primary key attribute values that define specific items in the table.
- * *AttributesToGet* - One or more attributes to be retrieved from the table or index. By default, all attributes are returned. If a requested attribute is not found, it does not appear in the result.

- * *ConsistentRead* - The consistency of a read operation. If set to `true`, then a strongly consistent read is used; otherwise, an eventually consistent read is used.

If there are no unprocessed keys remaining, the response contains an empty *UnprocessedKeys* map.

- * (*string*) –

- (*dict*) –

Represents a set of primary keys and, for each key, the attributes to retrieve from the table.

For each primary key, you must provide *all* of the key attributes. For example, with a hash type primary key, you only need to provide the hash attribute. For a hash-and-range type primary key, you must provide *both* the hash attribute and the range attribute.

- **Keys** (*list*) –

The primary key attribute values that define the items and the attributes associated with the items.

- (*dict*) –

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –

- **M** (*dict*) –

A Map of attribute values.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **AttributesToGet** (*list*) –

One or more attributes to retrieve from the table or index. If no attribute names are specified then all attributes will be returned. If any of the specified attributes are not found, they will not appear in the result.

- (*string*) –
- **ConsistentRead** (*boolean*) –

The consistency of a read operation. If set to `true`, then a strongly consistent read is used; otherwise, an eventually consistent read is used.

- **ProjectionExpression** (*string*) –

A string that identifies one or more attributes to retrieve from the table. These attributes can include scalars, sets, or elements of a JSON document. The attributes in the *ProjectionExpression* must be separated by commas.

If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

For more information, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

Note: *ProjectionExpression* replaces the legacy *AttributesToGet* parameter.

- **ExpressionAttributeNames** (*dict*) –

One or more substitution tokens for attribute names in an expression. The following are some use cases for using *Expres-*

sionAttributeNames :

- To access an attribute whose name conflicts with a DynamoDB reserved word.
- To create a placeholder for repeating occurrences of an attribute name in an expression.
- To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the # character in an expression to dereference an attribute name. For example, consider the following attribute name:

- Percentile

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see [Reserved Words](#) in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for *ExpressionAttributeNames* :

- { "#P": "Percentile" }

You could then use this substitution in an expression, as in this example:

- #P = :val

Note: Tokens that begin with the : character are *expression attribute values* , which are placeholders for the actual value at runtime.

For more information on expression attribute names, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide* .

- (string) –
- (string) –

– ConsumedCapacity (list) –

The read capacity units consumed by the operation.

Each element consists of:

- * *TableName* - The table that consumed the provisioned throughput.
- * *CapacityUnits* - The total number of capacity units consumed.
- * (dict) –

The capacity units consumed by an operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. *ConsumedCapacity* is only returned if the request asked for it. For more information, see [Provisioned Throughput](#) in the *Amazon DynamoDB Developer Guide* .

- **TableName** (string) –

The name of the table that was affected by the operation.

- **CapacityUnits** (float) –

The total number of capacity units consumed by the operation.

- **Table** (dict) –

The amount of throughput consumed on the table affected by the operation.

- **CapacityUnits** (float) –

The total number of capacity units consumed on a table or an index.

- **LocalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each local index affected by the operation.

- (*string*) –
- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

- **GlobalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each global index affected by the operation.

- (*string*) –
- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

batch_write_item (***kwargs*)

The *BatchWriteItem* operation puts or deletes multiple items in one or more tables. A single call to *BatchWriteItem* can write up to 16 MB of data, which can comprise as many as 25 put or delete requests. Individual items to be written can be as large as 400 KB.

Note: *BatchWriteItem* cannot update items. To update items, use the *UpdateItem* API.

The individual *PutItem* and *DeleteItem* operations specified in *BatchWriteItem* are atomic; however *BatchWriteItem* as a whole is not. If any requested operations fail because the table's provisioned throughput is exceeded or an internal processing failure occurs, the failed operations are returned in the *UnprocessedItems* response parameter. You can investigate and optionally resend the requests. Typically, you would call *BatchWriteItem* in a loop. Each iteration would check for unprocessed items and submit a new *BatchWriteItem* request with those unprocessed items until all items have been processed.

Note that if *none* of the items can be processed due to insufficient provisioned throughput on all of the tables in the request, then *BatchWriteItem* will return a *ProvisionedThroughputExceededException*.

Warning: If DynamoDB returns any unprocessed items, you should retry the batch operation on those items. However, *we strongly recommend that you use an exponential backoff algorithm*. If you retry the batch operation immediately, the underlying read or write requests can still fail due to throttling on the individual tables. If you delay the batch operation using exponential backoff, the individual requests in the batch are much more likely to succeed.

For more information, see [Batch Operations and Error Handling](#) in the *Amazon DynamoDB Developer Guide*.

With *BatchWriteItem*, you can efficiently write or delete large amounts of data, such as from Amazon Elastic MapReduce (EMR), or copy data from another database into DynamoDB. In order to improve performance with these large-scale operations, *BatchWriteItem* does not behave in the same way as indi-

vidual *PutItem* and *DeleteItem* calls would. For example, you cannot specify conditions on individual put and delete requests, and *BatchWriteItem* does not return deleted items in the response.

If you use a programming language that supports concurrency, you can use threads to write items in parallel. Your application must include the necessary logic to manage the threads. With languages that don't support threading, you must update or delete the specified items one at a time. In both situations, *BatchWriteItem* provides an alternative where the API performs the specified put and delete operations in parallel, giving you the power of the thread pool approach without having to introduce complexity into your application.

Parallel processing reduces latency, but each specified put and delete request consumes the same number of write capacity units whether it is processed in parallel or not. Delete operations on nonexistent items consume one write capacity unit.

If one or more of the following is true, DynamoDB rejects the entire batch write operation:

- One or more tables specified in the *BatchWriteItem* request does not exist.
- Primary key attributes specified on an item in the request do not match those in the corresponding table's primary key schema.
- You try to perform multiple operations on the same item in the same *BatchWriteItem* request. For example, you cannot put and delete the same item in the same *BatchWriteItem* request.
- There are more than 25 requests in the batch.
- Any individual item in a batch exceeds 400 KB.
- The total request size exceeds 16 MB.

Request Syntax

```
response = client.batch_write_item(
    RequestItems={
        'string': [
            {
                'PutRequest': {
                    'Item': {
                        'string': {
                            'S': 'string',
                            'N': 'string',
                            'B': b'bytes',
                            'SS': [
                                'string',
                            ],
                            'NS': [
                                'string',
                            ],
                            'BS': [
                                b'bytes',
                            ],
                            'M': {
                                'string': {'... recursive ...'}
                            },
                            'L': [
                                {'... recursive ...'},
                            ],
                            'NULL': True|False,
                            'BOOL': True|False
                        }
                    }
                },
            ],
        'DeleteRequest': {
            'Key': {
                'string': {
```



```

        'S': 'string',
        'N': 'string',
        'B': b'bytes',
        'SS': [
            'string',
        ],
        'NS': [
            'string',
        ],
        'BS': [
            b'bytes',
        ],
        'M': {
            'string': {'... recursive ...'}
        },
        'L': [
            {'... recursive ...'},
        ],
        'NULL': True|False,
        'BOOL': True|False
    }
}
},
]
},
ReturnConsumedCapacity='INDEXES'|'TOTAL'|'NONE',
ReturnItemCollectionMetrics='SIZE'|'NONE'
)

```

Parameters

- **RequestItems** (*dict*) – [REQUIRED]

A map of one or more table names and, for each table, a list of operations to be performed (*DeleteRequest* or *PutRequest*). Each element in the map consists of the following:

- *DeleteRequest* - Perform a *DeleteItem* operation on the specified item. The item to be deleted is identified by a *Key* subelement:
 - * *Key* - A map of primary key attribute values that uniquely identify the ! item. Each entry in this map consists of an attribute name and an attribute value. For each primary key, you must provide *all* of the key attributes. For example, with a hash type primary key, you only need to provide the hash attribute. For a hash-and-range type primary key, you must provide *both* the hash attribute and the range attribute.
- *PutRequest* - Perform a *PutItem* operation on the specified item. The item to be put is identified by an *Item* subelement:
 - * *Item* - A map of attributes and their values. Each entry in this map consists of an attribute name and an attribute value. Attribute values must not be null; string and binary type attributes must have lengths greater than zero; and set type attributes must not be empty. Requests that contain empty values will be rejected with a *ValidationException* exception. If you specify any attributes that are part of an index key, then the data types for those attributes must match those of the schema in the table's attribute definition.
- (*string*) –
 - * (*list*) –
 - (*dict*) –

Represents an operation to perform - either *DeleteItem* or *PutItem* . You can only request one of these operations, not both, in a single *WriteRequest* . If you do need to perform both of these operations, you will need to provide two separate *WriteRequest* objects.

- **PutRequest** (*dict*) –

A request to perform a *PutItem* operation.

- **Item** (*dict*) – **[REQUIRED]**

A map of attribute name to attribute values, representing the primary key of an item to be processed by *PutItem* . All of the table's primary key attributes must be specified, and their data types must match those of the table's key schema. If any attributes are present in the item which are part of an index key schema for the table, their types must match the index key schema.

- (*string*) –
- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –

- **M** (*dict*) –

A Map of attribute values.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book

item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **DeleteRequest** (*dict*) –

A request to perform a *DeleteItem* operation.

- **Key** (*dict*) – **[REQUIRED]**

A map of attribute name to attribute values, representing the primary key of the item to delete. All of the table's primary key attributes must be specified, and their data types must match those of the table's key schema.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- *(bytes)* –
- **M** (*dict*) –

A Map of attribute values.

- *(string)* –
- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **ReturnConsumedCapacity** (*string*) – Determines the level of detail about provisioned throughput consumption that is returned in the response:
 - **INDEXES** - The response includes the aggregate *ConsumedCapacity* for the operation, together with *ConsumedCapacity* for each table and secondary index that was accessed. Note that some operations, such as *GetItem* and *BatchGetItem* , do not access any indexes at all. In these cases, specifying **INDEXES** will only return *ConsumedCapacity* information for table(s).
 - **TOTAL** - The response includes only the aggregate *ConsumedCapacity* for the operation.
 - **NONE** - No *ConsumedCapacity* details are included in the response.
- **ReturnItemCollectionMetrics** (*string*) – Determines whether item collection metrics are returned. If set to **SIZE** , the response includes statistics about item collections, if any, that were modified during the operation are returned in the response. If set to **NONE** (the default), no statistics are returned.

Return type dict

Returns

Response Syntax

```
{
  'UnprocessedItems': {
    'string': [
```

```

{
    'PutRequest': {
        'Item': {
            'string': {
                'S': 'string',
                'N': 'string',
                'B': b'bytes',
                'SS': [
                    'string',
                ],
                'NS': [
                    'string',
                ],
                'BS': [
                    b'bytes',
                ],
                'M': {
                    'string': {'... recursive ...'}
                },
                'L': [
                    {'... recursive ...'},
                ],
                'NULL': True|False,
                'BOOL': True|False
            }
        }
    },
    'DeleteRequest': {
        'Key': {
            'string': {
                'S': 'string',
                'N': 'string',
                'B': b'bytes',
                'SS': [
                    'string',
                ],
                'NS': [
                    'string',
                ],
                'BS': [
                    b'bytes',
                ],
                'M': {
                    'string': {'... recursive ...'}
                },
                'L': [
                    {'... recursive ...'},
                ],
                'NULL': True|False,
                'BOOL': True|False
            }
        }
    },
    'ItemCollectionMetrics': {
        'string': [

```

```
{
  'ItemCollectionKey': {
    'string': {
      'S': 'string',
      'N': 'string',
      'B': b'bytes',
      'SS': [
        'string',
      ],
      'NS': [
        'string',
      ],
      'BS': [
        b'bytes',
      ],
      'M': {
        'string': {'... recursive ...'}
      },
      'L': [
        {'... recursive ...'},
      ],
      'NULL': True|False,
      'BOOL': True|False
    }
  },
  'SizeEstimateRangeGB': [
    123.0,
  ]
},
]
},
'ConsumedCapacity': [
  {
    'TableName': 'string',
    'CapacityUnits': 123.0,
    'Table': {
      'CapacityUnits': 123.0
    },
    'LocalSecondaryIndexes': {
      'string': {
        'CapacityUnits': 123.0
      }
    },
    'GlobalSecondaryIndexes': {
      'string': {
        'CapacityUnits': 123.0
      }
    }
  }
],
]
```

Response Structure

- (*dict*) –

Represents the output of a *BatchWriteItem* operation.

- **UnprocessedItems** (*dict*) –

A map of tables and requests against those tables that were not processed.

The *UnprocessedItems* value is in the same form as *RequestItems*, so you can provide this value directly to a subsequent *BatchGetItem* operation. For more information, see *RequestItems* in the Request Parameters section.

Each *UnprocessedItems* entry consists of a table name and, for that table, a list of operations to perform (*DeleteRequest* or *PutRequest*).

- * *DeleteRequest* - Perform a *DeleteItem* operation on the specified item. The item to be deleted is identified by a *Key* subelement:
 - * *Key* - A map of primary key attribute values that uniquely identify the item. Each entry in this map consists of an attribute name and an attribute value.
- * *PutRequest* - Perform a *PutItem* operation on the specified item. The item to be put is identified by an *Item* subelement:
 - * *Item* - A map of attributes and their values. Each entry in this map consists of an attribute name and an attribute value. Attribute values must not be null; string and binary type attributes must have lengths greater than zero; and set type attributes must not be empty. Requests that contain empty values will be rejected with a *ValidationException* exception. If you specify any attributes that are part of an index key, then the data types for those attributes must match those of the schema in the table's attribute definition.

If there are no unprocessed items remaining, the response contains an empty *UnprocessedItems* map.

- * (*string*) –
 - (*list*) –
 - (*dict*) –

Represents an operation to perform - either *DeleteItem* or *PutItem*. You can only request one of these operations, not both, in a single *WriteRequest*. If you do need to perform both of these operations, you will need to provide two separate *WriteRequest* objects.

- **PutRequest** (*dict*) –

A request to perform a *PutItem* operation.

- **Item** (*dict*) –

A map of attribute name to attribute values, representing the primary key of an item to be processed by *PutItem*. All of the table's primary key attributes must be specified, and their data types must match those of the table's key schema. If any attributes are present in the item which are part of an index key schema for the table, their types must match the index key schema.

- (*string*) –
- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –
- **NS** (*list*) –

A Number Set data type.

- (*string*) –
- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –
- **M** (*dict*) –

A Map of attribute values.

- (*string*) –
- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **DeleteRequest** (*dict*) –

A request to perform a *DeleteItem* operation.

- **Key** (*dict*) –

A map of attribute name to attribute values, representing the primary key of the item to delete. All of the table's primary key attributes must be specified, and their data types must match those of the table's key schema.

- *(string)* –
- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** *(string)* –

A String data type.

- **N** *(string)* –

A Number data type.

- **B** *(bytes)* –

A Binary data type.

- **SS** *(list)* –

A String Set data type.

- *(string)* –

- **NS** *(list)* –

A Number Set data type.

- *(string)* –

- **BS** *(list)* –

A Binary Set data type.

- *(bytes)* –

- **M** *(dict)* –

A Map of attribute values.

- *(string)* –

- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** *(list)* –

A List of attribute values.

- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** *(boolean)* –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **ItemCollectionMetrics** (*dict*) –

A list of tables that were processed by *BatchWriteItem* and, for each table, information about any item collections that were affected by individual *DeleteItem* or *PutItem* operations.

Each entry consists of the following subelements:

- * *ItemCollectionKey* - The hash key value of the item collection. This is the same as the hash key of the item.
- * *SizeEstimateRange* - An estimate of item collection size, expressed in GB. This is a two-element array containing a lower bound and an upper bound for the estimate. The estimate includes the size of all the items in the table, plus the size of all attributes projected into all of the local secondary indexes on the table. Use this estimate to measure whether a local secondary index is approaching its size limit. The estimate is subject to change over time; therefore, do not rely on the precision or accuracy of the estimate.

- * (*string*) –

- (*list*) –

- (*dict*) –

Information about item collections, if any, that were affected by the operation. *ItemCollectionMetrics* is only returned if the request asked for it. If the table does not have any local secondary indexes, this information is not returned in the response.

- **ItemCollectionKey** (*dict*) –

The hash key value of the item collection. This value is the same as the hash key of the item.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –

- **M** (*dict*) –

A Map of attribute values.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **SizeEstimateRangeGB** (*list*) –

An estimate of item collection size, in gigabytes. This value is a two-element array containing a lower bound and an upper bound for the estimate. The estimate includes the size of all the items in the table, plus the size of all attributes projected into all of the local secondary indexes on that table. Use this estimate to measure whether a local secondary index is approaching its size limit.

The estimate is subject to change over time; therefore, do not rely on the precision or accuracy of the estimate.

- (*float*) –

- **ConsumedCapacity** (*list*) –

The capacity units consumed by the operation.

Each element consists of:

- * *TableName* - The table that consumed the provisioned throughput.
- * *CapacityUnits* - The total number of capacity units consumed.

* (*dict*) –

The capacity units consumed by an operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. *ConsumedCapacity* is only returned if the request asked for it. For more information, see [Provisioned Throughput](#) in the *Amazon DynamoDB Developer Guide*.

- **TableName** (*string*) –

The name of the table that was affected by the operation.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed by the operation.

- **Table** (*dict*) –

The amount of throughput consumed on the table affected by the operation.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

- **LocalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each local index affected by the operation.

- (*string*) –

- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

- **GlobalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each global index affected by the operation.

- (*string*) –

- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

create_table (**kwargs)

The *CreateTable* operation adds a new table to your account. In an AWS account, table names must be unique within each region. That is, you can have two tables with same name if you create the tables in different regions.

CreateTable is an asynchronous operation. Upon receiving a *CreateTable* request, DynamoDB immediately returns a response with a *TableStatus* of *CREATING*. After the table is created, DynamoDB sets the *TableStatus* to *ACTIVE*. You can perform read and write operations only on an *ACTIVE* table.

You can optionally define secondary indexes on the new table, as part of the *CreateTable* operation. If you want to create multiple tables with secondary indexes on them, you must create the tables sequentially. Only one table with secondary indexes can be in the *CREATING* state at any given time.

You can use the *DescribeTable* API to check the table status.

Request Syntax

```
response = client.create_table(
    AttributeDefinitions=[
        {
            'AttributeName': 'string',
            'AttributeType': 'S'|'N'|'B'
        },
    ],
    TableName='string',
    KeySchema=[
        {
            'AttributeName': 'string',
            'KeyType': 'HASH'|'RANGE'
        },
    ],
    LocalSecondaryIndexes=[
        {
            'IndexName': 'string',
            'KeySchema': [
                {
                    'AttributeName': 'string',
                    'KeyType': 'HASH'|'RANGE'
                },
            ],
            'Projection': {
                'ProjectionType': 'ALL'|'KEYS_ONLY'|'INCLUDE',
                'NonKeyAttributes': [
                    'string',
                ]
            }
        },
    ],
    GlobalSecondaryIndexes=[
        {
            'IndexName': 'string',
            'KeySchema': [
                {
                    'AttributeName': 'string',
                    'KeyType': 'HASH'|'RANGE'
                },
            ],
            'Projection': {
                'ProjectionType': 'ALL'|'KEYS_ONLY'|'INCLUDE',
```

```
        'NonKeyAttributes': [
            'string',
        ]
    },
    'ProvisionedThroughput': {
        'ReadCapacityUnits': 123,
        'WriteCapacityUnits': 123
    }
},
ProvisionedThroughput={
    'ReadCapacityUnits': 123,
    'WriteCapacityUnits': 123
},
StreamSpecification={
    'StreamEnabled': True|False,
    'StreamViewType': 'NEW_IMAGE' | 'OLD_IMAGE' | 'NEW_AND_OLD_IMAGES' | 'KEYS_ONLY'
}
)
```

Parameters

- **AttributeDefinitions** (*list*) – [REQUIRED]

An array of attributes that describe the key schema for the table and indexes.

– (*dict*) –

Represents an attribute for describing the key schema for the table and indexes.

- * **AttributeName** (*string*) – [REQUIRED]

A name for the attribute.

- * **AttributeType** (*string*) – [REQUIRED]

The data type for the attribute.

- **TableName** (*string*) – [REQUIRED]

The name of the table to create.

- **KeySchema** (*list*) – [REQUIRED]

Specifies the attributes that make up the primary key for a table or an index. The attributes in *KeySchema* must also be defined in the *AttributeDefinitions* array. For more information, see [Data Model](#) in the *Amazon DynamoDB Developer Guide*.

Each *KeySchemaElement* in the array is composed of:

- *AttributeName* - The name of this key attribute.
- *KeyType* - Determines whether the key attribute is `HASH` or `RANGE`.

For a primary key that consists of a hash attribute, you must provide exactly one element with a *KeyType* of `HASH`.

For a primary key that consists of hash and range attributes, you must provide exactly two elements, in this order: The first element must have a *KeyType* of `HASH`, and the second element must have a *KeyType* of `RANGE`.

For more information, see [Specifying the Primary Key](#) in the *Amazon DynamoDB Developer Guide*.

– (*dict*) –

Represents a *single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of

an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement*. A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.

* **AttributeName** (*string*) – [REQUIRED]

The name of a key attribute.

* **KeyType** (*string*) – [REQUIRED]

The attribute data, consisting of the data type and the attribute value itself.

- **LocalSecondaryIndexes** (*list*) – One or more local secondary indexes (the maximum is five) to be created on the table. Each index is scoped to a given hash key value. There is a 10 GB size limit per hash key; otherwise, the size of a local secondary index is unconstrained.

Each local secondary index in the array includes the following:

- *IndexName* - The name of the local secondary index. Must be unique only for this table.
- *KeySchema* - Specifies the key schema for the local secondary index. The key schema must begin with the same hash key attribute as the table.
- *Projection* - Specifies attributes that are copied (projected) from the table into the index. These are in addition to the primary key attributes and index key attributes, which are automatically projected. Each attribute specification is composed of:
 - * *ProjectionType* - One of the following:
 - * KEYS_ONLY - Only the index and primary keys are projected into the index.
 - * INCLUDE - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes*.
 - * ALL - All of the table attributes are projected into the index.
 - * *NonKeyAttributes* - A list of one or more non-key attribute names that are projected into the secondary index. The total count of attributes provided in *NonKeyAttributes*, summed across all of the secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.
- (*dict*) –

Represents the properties of a local secondary index.

* **IndexName** (*string*) – [REQUIRED]

The name of the local secondary index. The name must be unique among all other indexes on this table.

* **KeySchema** (*list*) – [REQUIRED]

The complete key schema for the local secondary index, consisting of one or more pairs of attribute names and key types (HASH or RANGE).

· (*dict*) –

Represents a *single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be

represented by one *KeySchemaElement* . A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.

- **AttributeName** (*string*) – [REQUIRED]

The name of a key attribute.

- **KeyType** (*string*) – [REQUIRED]

The attribute data, consisting of the data type and the attribute value itself.

- * **Projection** (*dict*) – [REQUIRED]

Represents attributes that are copied (projected) from the table into an index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

- **ProjectionType** (*string*) –

The set of attributes that are projected into the index:

- KEYS_ONLY - Only the index and primary keys are projected into the index.
- INCLUDE - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes* .
- ALL - All of the table attributes are projected into the index.
- **NonKeyAttributes** (*list*) –

Represents the non-key attribute names which will be projected into the index.

For local secondary indexes, the total count of *NonKeyAttributes* summed across all of the local secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.

- (*string*) –

- **GlobalSecondaryIndexes** (*list*) – One or more global secondary indexes (the maximum is five) to be created on the table. Each global secondary index in the array includes the following:

- *IndexName* - The name of the global secondary index. Must be unique only for this table.
- *KeySchema* - Specifies the key schema for the global secondary index.
- *Projection* - Specifies attributes that are copied (projected) from the table into the index. These are in addition to the primary key attributes and index key attributes, which are automatically projected. Each attribute specification is composed of: * *ProjectionType* - One of the following:
 - * KEYS_ONLY - Only the index and primary keys are projected into the index.
 - * INCLUDE - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes* .
 - * ALL - All of the table attributes are projected into the index.
- *NonKeyAttributes* - A list of one or more non-key attribute names that are projected into the secondary index. The total count of attributes provided in *NonKeyAttributes* , summed across all of the secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.

- *ProvisionedThroughput* - The provisioned throughput settings for the global secondary index, consisting of read and write capacity units.
- (*dict*) –

Represents the properties of a global secondary index.

* **IndexName** (*string*) – [REQUIRED]

The name of the global secondary index. The name must be unique among all other indexes on this table.

* **KeySchema** (*list*) – [REQUIRED]

The complete key schema for a global secondary index, which consists of one or more pairs of attribute names and key types (HASH or RANGE).

· (*dict*) –

Represents a *single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement* . A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.

· **AttributeName** (*string*) – [REQUIRED]

The name of a key attribute.

· **KeyType** (*string*) – [REQUIRED]

The attribute data, consisting of the data type and the attribute value itself.

* **Projection** (*dict*) – [REQUIRED]

Represents attributes that are copied (projected) from the table into an index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

· **ProjectionType** (*string*) –

The set of attributes that are projected into the index:

- KEYS_ONLY - Only the index and primary keys are projected into the index.
- INCLUDE - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes* .
- ALL - All of the table attributes are projected into the index.
- **NonKeyAttributes** (*list*) –

Represents the non-key attribute names which will be projected into the index.

For local secondary indexes, the total count of *NonKeyAttributes* summed across all of the local secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.

· (*string*) –

* **ProvisionedThroughput** (*dict*) – [REQUIRED]

Represents the provisioned throughput settings for a specified table or index. The settings can be modified using the *UpdateTable* operation.

For current minimum and maximum provisioned throughput values, see [Limits](#) in the *Amazon DynamoDB Developer Guide* .

· **ReadCapacityUnits** (*integer*) – [REQUIRED]

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a *ThrottlingException* . For more information, see [Specifying Read and Write Requirements](#) in the *Amazon DynamoDB Developer Guide* .

· **WriteCapacityUnits** (*integer*) – [REQUIRED]

The maximum number of writes consumed per second before DynamoDB returns a *ThrottlingException* . For more information, see [Specifying Read and Write Requirements](#) in the *Amazon DynamoDB Developer Guide* .

• **ProvisionedThroughput** (*dict*) – [REQUIRED]

Represents the provisioned throughput settings for a specified table or index. The settings can be modified using the *UpdateTable* operation.

For current minimum and maximum provisioned throughput values, see [Limits](#) in the *Amazon DynamoDB Developer Guide* .

– **ReadCapacityUnits** (*integer*) – [REQUIRED]

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a *ThrottlingException* . For more information, see [Specifying Read and Write Requirements](#) in the *Amazon DynamoDB Developer Guide* .

– **WriteCapacityUnits** (*integer*) – [REQUIRED]

The maximum number of writes consumed per second before DynamoDB returns a *ThrottlingException* . For more information, see [Specifying Read and Write Requirements](#) in the *Amazon DynamoDB Developer Guide* .

• **StreamSpecification** (*dict*) – The settings for DynamoDB Streams on the table. These settings consist of:

- *StreamEnabled* - Indicates whether Streams is to be enabled (true) or disabled (false).
- *StreamViewType* - When an item in the table is modified, *StreamViewType* determines what information is written to the table's stream. Valid values for *StreamViewType* are: * *KEYS_ONLY* - Only the key attributes of the modified item are written to the stream.
- *NEW_IMAGE* - The entire item, as it appears after it was modified, is written to the stream.
- *OLD_IMAGE* - The entire item, as it appeared before it was modified, is written to the stream.
- *NEW_AND_OLD_IMAGES* - Both the new and the old item images of the item are written to the stream.

– **StreamEnabled** (*boolean*) –

Indicates whether DynamoDB Streams is enabled (true) or disabled (false) on the table.

– **StreamViewType** (*string*) –

The DynamoDB Streams settings for the table. These settings consist of:

- * *StreamEnabled* - Indicates whether DynamoDB Streams is enabled (true) or disabled (false) on the table.
- * *StreamViewType* - When an item in the table is modified, *StreamViewType* determines what information is written to the stream for this table. Valid values for *StreamViewType* are:
 - * *KEYS_ONLY* - Only the key attributes of the modified item are written to the stream.
 - * *NEW_IMAGE* - The entire item, as it appears after it was modified, is written to the stream.
 - * *OLD_IMAGE* - The entire item, as it appeared before it was modified, is written to the stream.
 - * *NEW_AND_OLD_IMAGES* - Both the new and the old item images of the item are written to the stream.

Return type dict

Returns

Response Syntax

```
{
  'TableDescription': {
    'AttributeDefinitions': [
      {
        'AttributeName': 'string',
        'AttributeType': 'S'|'N'|'B'
      },
    ],
    'TableName': 'string',
    'KeySchema': [
      {
        'AttributeName': 'string',
        'KeyType': 'HASH'|'RANGE'
      },
    ],
    'TableStatus': 'CREATING'|'UPDATING'|'DELETING'|'ACTIVE',
    'CreationDateTime': datetime(2015, 1, 1),
    'ProvisionedThroughput': {
      'LastIncreaseDateTime': datetime(2015, 1, 1),
      'LastDecreaseDateTime': datetime(2015, 1, 1),
      'NumberOfDecreasesToday': 123,
      'ReadCapacityUnits': 123,
      'WriteCapacityUnits': 123
    },
    'TableSizeBytes': 123,
    'ItemCount': 123,
    'TableArn': 'string',
    'LocalSecondaryIndexes': [
      {
        'IndexName': 'string',
        'KeySchema': [
          {
            'AttributeName': 'string',
            'KeyType': 'HASH'|'RANGE'
          },
        ],
        'Projection': {
          'ProjectionType': 'ALL'|'KEYS_ONLY'|'INCLUDE',
```

```
        'NonKeyAttributes': [
            'string',
        ],
        'IndexSizeBytes': 123,
        'ItemCount': 123,
        'IndexArn': 'string'
    },
],
'GlobalSecondaryIndexes': [
    {
        'IndexName': 'string',
        'KeySchema': [
            {
                'AttributeName': 'string',
                'KeyType': 'HASH' | 'RANGE'
            },
        ],
        'Projection': {
            'ProjectionType': 'ALL' | 'KEYS_ONLY' | 'INCLUDE',
            'NonKeyAttributes': [
                'string',
            ]
        },
        'IndexStatus': 'CREATING' | 'UPDATING' | 'DELETING' | 'ACTIVE',
        'Backfilling': True | False,
        'ProvisionedThroughput': {
            'LastIncreaseDateTime': datetime(2015, 1, 1),
            'LastDecreaseDateTime': datetime(2015, 1, 1),
            'NumberOfDecreasesToday': 123,
            'ReadCapacityUnits': 123,
            'WriteCapacityUnits': 123
        },
        'IndexSizeBytes': 123,
        'ItemCount': 123,
        'IndexArn': 'string'
    },
],
'StreamSpecification': {
    'StreamEnabled': True | False,
    'StreamViewType': 'NEW_IMAGE' | 'OLD_IMAGE' | 'NEW_AND_OLD_IMAGES' | 'KEYS_O
},
'LatestStreamLabel': 'string',
'LatestStreamArn': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a *CreateTable* operation.

- **TableDescription** (*dict*) –

Represents the properties of a table.

- * **AttributeDefinitions** (*list*) –

An array of *AttributeDefinition* objects. Each of these objects describes one attribute in the table and index key schema.

Each *AttributeDefinition* object in this array is composed of:

- *AttributeName* - The name of the attribute.
- *AttributeType* - The data type for the attribute.
- (*dict*) –

Represents an attribute for describing the key schema for the table and indexes.

- **AttributeName** (*string*) –

A name for the attribute.

- **AttributeType** (*string*) –

The data type for the attribute.

- * **TableName** (*string*) –

The name of the table.

- * **KeySchema** (*list*) –

The primary key structure for the table. Each *KeySchemaElement* consists of:

- *AttributeName* - The name of the attribute.
- *KeyType* - The key type for the attribute. Can be either `HASH` or `RANGE`.

For more information about primary keys, see [Primary Key](#) in the *Amazon DynamoDB Developer Guide*.

- (*dict*) –

Represents a *single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement*. A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.

- **AttributeName** (*string*) –

The name of a key attribute.

- **KeyType** (*string*) –

The attribute data, consisting of the data type and the attribute value itself.

- * **TableStatus** (*string*) –

The current state of the table:

- *CREATING* - The table is being created.
- *UPDATING* - The table is being updated.
- *DELETING* - The table is being deleted.
- *ACTIVE* - The table is ready for use.

- * **CreationDateTime** (*datetime*) –

The date and time when the table was created, in [UNIX epoch time](#) format.

- * **ProvisionedThroughput** (*dict*) –

The provisioned throughput settings for the table, consisting of read

and write capacity units, along with data about increases and decreases.

- **LastIncreaseDateTime** (*datetime*) –

The date and time of the last provisioned throughput increase for this table.

- **LastDecreaseDateTime** (*datetime*) –

The date and time of the last provisioned throughput decrease for this table.

- **NumberOfDecreasesToday** (*integer*) –

The number of provisioned throughput decreases for this table during this UTC calendar day. For current maximums on provisioned throughput decreases, see [Limits](#) in the *Amazon DynamoDB Developer Guide*.

- **ReadCapacityUnits** (*integer*) –

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a *ThrottlingException*. Eventually consistent reads require less effort than strongly consistent reads, so a setting of 50 *ReadCapacityUnits* per second provides 100 eventually consistent *ReadCapacityUnits* per second.

- **WriteCapacityUnits** (*integer*) –

The maximum number of writes consumed per second before DynamoDB returns a *ThrottlingException*.

- * **TableSizeBytes** (*integer*) –

The total size of the specified table, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- * **ItemCount** (*integer*) –

The number of items in the specified table. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- * **TableArn** (*string*) –

The Amazon Resource Name (ARN) that uniquely identifies the table.

- * **LocalSecondaryIndexes** (*list*) –

Represents one or more local secondary indexes on the table. Each index is scoped to a given hash key value. Tables with one or more local secondary indexes are subject to an item collection size limit, where the amount of data within a given item collection cannot exceed 10 GB. Each element is composed of:

- *IndexName* - The name of the local secondary index.
- *KeySchema* - Specifies the complete index key schema. The attribute names in the key schema must be between 1 and 255 characters (inclusive). The key schema must begin with the same hash key attribute as the table.
- *Projection* - Specifies attributes that are copied (projected) from the table into the index. These are in addition to the primary key attributes and index key attributes, which are automatically projected. Each attribute specification is composed

- of: * *ProjectionType* - One of the following: * `KEYS_ONLY` - Only the index and primary keys are projected into the index.
- `INCLUDE` - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes*.
- `ALL` - All of the table attributes are projected into the index.
- *NonKeyAttributes* - A list of one or more non-key attribute names that are projected into the secondary index. The total count of attributes provided in *NonKeyAttributes*, summed across all of the secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.
- *IndexSizeBytes* - Represents the total size of the index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.
- *ItemCount* - Represents the number of items in the index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

If the table is in the `DELETING` state, no information about indexes will be returned.

- *(dict)* –
Represents the properties of a local secondary index.
- **IndexName** (*string*) –
Represents the name of the local secondary index.
- **KeySchema** (*list*) –
The complete index key schema, which consists of one or more pairs of attribute names and key types (`HASH` or `RANGE`).
- *(dict)* –
Represents a *single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement*. A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.
- **AttributeName** (*string*) –
The name of a key attribute.
- **KeyType** (*string*) –
The attribute data, consisting of the data type and the attribute value itself.
- **Projection** (*dict*) –
Represents attributes that are copied (projected) from the table into an index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.
- **ProjectionType** (*string*) –

The set of attributes that are projected into the index:

- **KEYS_ONLY** - Only the index and primary keys are projected into the index.
- **INCLUDE** - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes*.
- **ALL** - All of the table attributes are projected into the index.
- **NonKeyAttributes** (*list*) –

Represents the non-key attribute names which will be projected into the index.

For local secondary indexes, the total count of *NonKeyAttributes* summed across all of the local secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.

- (*string*) –
- **IndexSizeBytes** (*integer*) –

The total size of the specified index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- **ItemCount** (*integer*) –

The number of items in the specified index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- **IndexArn** (*string*) –

The Amazon Resource Name (ARN) that uniquely identifies the index.

* **GlobalSecondaryIndexes** (*list*) –

The global secondary indexes, if any, on the table. Each index is scoped to a given hash key value. Each element is composed of:

- *Backfilling* - If true, then the index is currently in the backfilling phase. Backfilling occurs only when a new global secondary index is added to the table; it is the process by which DynamoDB populates the new index with data from the table. (This attribute does not appear for indexes that were created during a *CreateTable* operation.)
- *IndexName* - The name of the global secondary index.
- *IndexSizeBytes* - The total size of the global secondary index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.
- *IndexStatus* - The current status of the global secondary index:
 - * *CREATING* - The index is being created.
 - * *UPDATING* - The index is being updated.
 - * *DELETING* - The index is being deleted.
 - * *ACTIVE* - The index is ready for use.
- *ItemCount* - The number of items in the global secondary index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.
- *KeySchema* - Specifies the complete index key schema. The attribute names in the key schema must be between 1 and 255 characters (inclusive). The key schema must begin with the

same hash key attribute as the table.

- *Projection* - Specifies attributes that are copied (projected) from the table into the index. These are in addition to the primary key attributes and index key attributes, which are automatically projected. Each attribute specification is composed of:
 - * *ProjectionType* - One of the following:
 - * **KEYS_ONLY** - Only the index and primary keys are projected into the index.
 - * **INCLUDE** - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes*.
 - * **ALL** - All of the table attributes are projected into the index.
- *NonKeyAttributes* - A list of one or more non-key attribute names that are projected into the secondary index. The total count of attributes provided in *NonKeyAttributes*, summed across all of the secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.
- *ProvisionedThroughput* - The provisioned throughput settings for the global secondary index, consisting of read and write capacity units, along with data about increases and decreases.

If the table is in the **DELETING** state, no information about indexes will be returned.

- *(dict)* –

Represents the properties of a global secondary index.

- **IndexName** (*string*) –

The name of the global secondary index.

- **KeySchema** (*list*) –

The complete key schema for the global secondary index, consisting of one or more pairs of attribute names and key types (**HASH** or **RANGE**).

- *(dict)* –

Represents a *single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement*. A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.

- **AttributeName** (*string*) –

The name of a key attribute.

- **KeyType** (*string*) –

The attribute data, consisting of the data type and the attribute value itself.

- **Projection** (*dict*) –

Represents attributes that are copied (projected) from the table into an index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

- **ProjectionType** (*string*) –

The set of attributes that are projected into the index:

- **KEYS_ONLY** - Only the index and primary keys are projected into the index.
- **INCLUDE** - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes*.
- **ALL** - All of the table attributes are projected into the index.
- **NonKeyAttributes** (*list*) –

Represents the non-key attribute names which will be projected into the index.

For local secondary indexes, the total count of *NonKeyAttributes* summed across all of the local secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.

- (*string*) –

- **IndexStatus** (*string*) –

The current state of the global secondary index:

- **CREATING** - The index is being created.
- **UPDATING** - The index is being updated.
- **DELETING** - The index is being deleted.
- **ACTIVE** - The index is ready for use.
- **Backfilling** (*boolean*) –

Indicates whether the index is currently backfilling. *Backfilling* is the process of reading items from the table and determining whether they can be added to the index. (Not all items will qualify: For example, a hash key attribute cannot have any duplicates.) If an item can be added to the index, DynamoDB will do so. After all items have been processed, the backfilling operation is complete and *Backfilling* is false.

Note: For indexes that were created during a *CreateTable* operation, the *Backfilling* attribute does not appear in the *DescribeTable* output.

- **ProvisionedThroughput** (*dict*) –

Represents the provisioned throughput settings for the table, consisting of read and write capacity units, along with data about increases and decreases.

- **LastIncreaseDateTime** (*datetime*) –

The date and time of the last provisioned throughput increase for this table.

- **LastDecreaseDateTime** (*datetime*) –

The date and time of the last provisioned throughput decrease for this table.

- **NumberOfDecreasesToday** (*integer*) –

The number of provisioned throughput decreases for this table during this UTC calendar day. For current maximums on

provisioned throughput decreases, see [Limits](#) in the *Amazon DynamoDB Developer Guide*.

- **ReadCapacityUnits** (*integer*) –

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a *ThrottlingException*. Eventually consistent reads require less effort than strongly consistent reads, so a setting of 50 *ReadCapacityUnits* per second provides 100 eventually consistent *ReadCapacityUnits* per second.

- **WriteCapacityUnits** (*integer*) –

The maximum number of writes consumed per second before DynamoDB returns a *ThrottlingException*.

- **IndexSizeBytes** (*integer*) –

The total size of the specified index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- **ItemCount** (*integer*) –

The number of items in the specified index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- **IndexArn** (*string*) –

The Amazon Resource Name (ARN) that uniquely identifies the index.

- * **StreamSpecification** (*dict*) –

The current DynamoDB Streams configuration for the table.

- **StreamEnabled** (*boolean*) –

Indicates whether DynamoDB Streams is enabled (true) or disabled (false) on the table.

- **StreamViewType** (*string*) –

The DynamoDB Streams settings for the table. These settings consist of:

- *StreamEnabled* - Indicates whether DynamoDB Streams is enabled (true) or disabled (false) on the table.
- *StreamViewType* - When an item in the table is modified, *StreamViewType* determines what information is written to the stream for this table. Valid values for *StreamViewType* are:
 - * *KEYS_ONLY* - Only the key attributes of the modified item are written to the stream.
 - * *NEW_IMAGE* - The entire item, as it appears after it was modified, is written to the stream.
 - * *OLD_IMAGE* - The entire item, as it appeared before it was modified, is written to the stream.
 - * *NEW_AND_OLD_IMAGES* - Both the new and the old item images of the item are written to the stream.

- * **LatestStreamLabel** (*string*) –

A timestamp, in ISO 8601 format, for this stream.

Note that *LatestStreamLabel* is not a unique identifier for the stream, because it is possible that a stream from another table might have the

same timestamp. However, the combination of the following three elements is guaranteed to be unique:

- the AWS customer ID.
- the table name.
- the *StreamLabel*.

* **LatestStreamArn** (*string*) –

The Amazon Resource Name (ARN) that uniquely identifies the latest stream for this table.

delete_item (**kwargs)

Deletes a single item in a table by primary key. You can perform a conditional delete operation that deletes the item if it exists, or if it has an expected attribute value.

In addition to deleting an item, you can also return the item's attribute values in the same operation, using the *ReturnValues* parameter.

Unless you specify conditions, the *DeleteItem* is an idempotent operation; running it multiple times on the same item or attribute does *not* result in an error response.

Conditional deletes are useful for deleting items only if specific conditions are met. If those conditions are met, DynamoDB performs the delete. Otherwise, the item is not deleted.

Request Syntax

```
response = client.delete_item(
    TableName='string',
    Key={
        'string': {
            'S': 'string',
            'N': 'string',
            'B': b'bytes',
            'SS': [
                'string',
            ],
            'NS': [
                'string',
            ],
            'BS': [
                b'bytes',
            ],
            'M': {
                'string': {'... recursive ...'}
            },
            'L': [
                {'... recursive ...'},
            ],
            'NULL': True|False,
            'BOOL': True|False
        }
    },
    Expected={
        'string': {
            'Value': {
                'S': 'string',
                'N': 'string',
                'B': b'bytes',
                'SS': [
                    'string',
```

```

        ],
        'NS': [
            'string',
        ],
        'BS': [
            b'bytes',
        ],
        'M': {
            'string': {'... recursive ...'}
        },
        'L': [
            {'... recursive ...'},
        ],
        'NULL': True|False,
        'BOOL': True|False
    },
    'Exists': True|False,
    'ComparisonOperator': 'EQ' | 'NE' | 'IN' | 'LE' | 'LT' | 'GE' | 'GT' | 'BETWEEN' | 'NOT_NULL' | '...',
    'AttributeValueList': [
        {
            'S': 'string',
            'N': 'string',
            'B': b'bytes',
            'SS': [
                'string',
            ],
            'NS': [
                'string',
            ],
            'BS': [
                b'bytes',
            ],
            'M': {
                'string': {'... recursive ...'}
            },
            'L': [
                {'... recursive ...'},
            ],
            'NULL': True|False,
            'BOOL': True|False
        },
    ],
    ],
    },
    ConditionalOperator='AND' | 'OR',
    ReturnValues='NONE' | 'ALL_OLD' | 'UPDATED_OLD' | 'ALL_NEW' | 'UPDATED_NEW',
    ReturnConsumedCapacity='INDEXES' | 'TOTAL' | 'NONE',
    ReturnItemCollectionMetrics='SIZE' | 'NONE',
    ConditionExpression='string',
    ExpressionAttributeNames={
        'string': 'string'
    },
    ExpressionAttributeValues={
        'string': {
            'S': 'string',
            'N': 'string',
            'B': b'bytes',
            'SS': [

```

```
        'string',
    ],
    'NS': [
        'string',
    ],
    'BS': [
        b'bytes',
    ],
    'M': {
        'string': {'... recursive ...'}
    },
    'L': [
        {'... recursive ...'},
    ],
    'NULL': True|False,
    'BOOL': True|False
}
)
```

Parameters

- **TableName** (*string*) – **[REQUIRED]**

The name of the table from which to delete the item.

- **Key** (*dict*) – **[REQUIRED]**

A map of attribute names to *AttributeValue* objects, representing the primary key of the item to delete.

For the primary key, you must provide all of the attributes. For example, with a hash type primary key, you only need to provide the hash attribute. For a hash-and-range type primary key, you must provide both the hash attribute and the range attribute.

- (*string*) –
 - * (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- *(string)* –
- **BS** (*list*) –

A Binary Set data type.

- *(bytes)* –
- **M** (*dict*) –

A Map of attribute values.

- *(string)* –
- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **Expected** (*dict*) –

Warning: This is a legacy parameter, for backward compatibility. New applications should use *ConditionExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception.

A map of attribute/condition pairs. *Expected* provides a conditional block for the *DeleteItem* operation.

Each element of *Expected* consists of an attribute name, a comparison operator, and one or more values. DynamoDB compares the attribute with the value(s) you supplied, using the comparison operator. For each *Expected* element, the result of the evaluation is either true or false.

If you specify more than one element in the *Expected* map, then by default all of the conditions must evaluate to true. In other words, the conditions are ANDed together. (You can use the *ConditionalOperator* parameter to OR the conditions instead. If you do this, then at least one of the conditions must evaluate to true, rather than all of them.)

If the *Expected* map evaluates to true, then the conditional operation succeeds; otherwise, it fails.

Expected contains the following:

- *AttributeValueList* - One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *ComparisonOperator* being used. For type Number, value comparisons are numeric. String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A , and a is greater than B . For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters . For type Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.
- *ComparisonOperator* - A comparator for evaluating attributes in the *AttributeValueList* . When performing the comparison, DynamoDB uses strongly consistent reads. The following comparison operators are available: EQ | NE | LE | LT | GE | GT | NOT_NULL | NULL | CONTAINS | NOT_CONTAINS | BEGINS_WITH | IN | BETWEEN The following are descriptions of each comparison operator. * EQ : Equal. EQ is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" } . Also, { "N" : "6" } does not equal { "NS" : ["6", "2", "1"] } .
- NE : Not equal. NE is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" } . Also, { "N" : "6" } does not equal { "NS" : ["6", "2", "1"] } .
- LE : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" } . Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] } .
- LT : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" } . Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] } .
- GE : Greater than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" } . Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] } .
- GT : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" } . Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] } .

- NOT_NULL : The attribute exists. NOT_NULL is supported for all datatypes, including lists and maps. .. note::This operator tests for the existence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using NOT_NULL , the result is a Boolean *true* . This result is because the attribute “a ” exists; its data type is not relevant to the NOT_NULL comparison operator.
- NULL : The attribute does not exist. NULL is supported for all datatypes, including lists and maps. .. note::This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using NULL , the result is a Boolean *false* . This is because the attribute “a ” exists; its data type is not relevant to the NULL comparison operator.
- CONTAINS : Checks for a subsequence, or value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS ”, “NS ”, or “BS ”), then the operator evaluates to true if it finds an exact match with any member of the set. CONTAINS is supported for lists: When evaluating “a CONTAINS b”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
- NOT_CONTAINS : Checks for absence of a subsequence, or absence of a value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS ”, “NS ”, or “BS ”), then the operator evaluates to true if it *does not* find an exact match with any member of the set. NOT_CONTAINS is supported for lists: When evaluating “a NOT CONTAINS b”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
- BEGINS_WITH : Checks for a prefix. *AttributeValueList* can contain only one *AttributeValue* of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).
- IN : Checks for matching elements within two sets. *AttributeValueList* can contain one or more *AttributeValue* elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
- BETWEEN : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not compare to { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] }

For usage examples of *AttributeValueList* and *ComparisonOperator* , see [Legacy Conditional Parameters](#) in the *Amazon DynamoDB Developer Guide* .

For backward compatibility with previous DynamoDB releases, the following parameters can be used instead of *AttributeValueList* and *ComparisonOperator* :

- *Value* - A value for DynamoDB to compare with an attribute.
- *Exists* - A Boolean value that causes DynamoDB to evaluate the value before attempting the conditional operation: * If *Exists* is `true` , DynamoDB will check to see if that attribute value already exists in the table. If it is found, then the condition evaluates to true; otherwise the condition evaluate to false.
- If *Exists* is `false` , DynamoDB assumes that the attribute value does *not* exist in the table. If in fact the value does not exist, then the assumption is valid and the condition evaluates to true. If the value is found, despite the assumption that it does not exist, the condition evaluates to false.

Note that the default value for *Exists* is `true` .

The *Value* and *Exists* parameters are incompatible with *AttributeValueList* and *ComparisonOperator* . Note that if you use both sets of parameters at once, DynamoDB will return a *ValidationException* exception.

Note: This parameter does not support attributes of type List or Map.

- *(string)* -
- * *(dict)* -

Represents a condition to be compared with an attribute value. This condition can be used with *DeleteItem* , *PutItem* or *UpdateItem* operations; if the comparison evaluates to true, the operation succeeds; if not, the operation fails. You can use *ExpectedAttributeValue* in one of two different ways:

- Use *AttributeValueList* to specify one or more values to compare against an attribute. Use *ComparisonOperator* to specify how you want to perform the comparison. If the comparison evaluates to true, then the conditional operation succeeds.
- Use *Value* to specify a value that DynamoDB will compare against an attribute. If the values match, then *ExpectedAttributeValue* evaluates to true and the conditional operation succeeds. Optionally, you can also set *Exists* to false, indicating that you *do not* expect to find the attribute value in the table. In this case, the conditional operation succeeds only if the comparison evaluates to false.

Value and *Exists* are incompatible with *AttributeValueList* and *ComparisonOperator* . Note that if you use both sets of parameters at once, DynamoDB will return a *ValidationException* exception.

- **Value** (*dict*) -

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) -
- A String data type.
- **N** (*string*) -

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –
- **NS** (*list*) –

A Number Set data type.

- (*string*) –
- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –
- **M** (*dict*) –

A Map of attribute values.

- (*string*) –
- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **Exists** (*boolean*) –

Causes DynamoDB to evaluate the value before attempting a conditional operation:

- If *Exists* is `true`, DynamoDB will check to see if that attribute value already exists in the table. If it is found, then the operation succeeds. If it is not found, the operation fails with a *ConditionalCheckFailedException*.
- If *Exists* is `false`, DynamoDB assumes that the attribute value does not exist in the table. If in fact the value does not exist, then the assumption is valid and the operation succeeds.

If the value is found, despite the assumption that it does not exist, the operation fails with a *ConditionalCheckFailedException*.

The default setting for *Exists* is `true`. If you supply a *Value* all by itself, DynamoDB assumes the attribute exists: You don't have to set *Exists* to `true`, because it is implied.

DynamoDB returns a *ValidationException* if:

- *Exists* is `true` but there is no *Value* to check. (You expect a value to exist, but don't specify what that value is.)
- *Exists* is `false` but you also provide a *Value*. (You cannot expect an attribute to have a value, while also expecting it not to exist.)
- **ComparisonOperator** (*string*) –

A comparator for evaluating attributes in the *AttributeValueList*. For example, equals, greater than, less than, etc.

The following comparison operators are available:

```
EQ | NE | LE | LT | GE | GT | NOT_NULL
| NULL | CONTAINS | NOT_CONTAINS |
BEGINS_WITH | IN | BETWEEN
```

The following are descriptions of each comparison operator.

- **EQ** : Equal. **EQ** is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not equal `{"NS": ["6", "2", "1"]}`.
- **NE** : Not equal. **NE** is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not equal `{"NS": ["6", "2", "1"]}`.
- **LE** : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
- **LT** : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
- **GE** : Greater than or equal. *AttributeValueList* can contain

only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .

- GT : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- NOT_NULL : The attribute exists. NOT_NULL is supported for all datatypes, including lists and maps. .. note::This operator tests for the existence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using NOT_NULL , the result is a Boolean *true* . This result is because the attribute “a ” exists; its data type is not relevant to the NOT_NULL comparison operator.
- NULL : The attribute does not exist. NULL is supported for all datatypes, including lists and maps. .. note::This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using NULL , the result is a Boolean *false* . This is because the attribute “a ” exists; its data type is not relevant to the NULL comparison operator.
- CONTAINS : Checks for a subsequence, or value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS ”, “NS ”, or “BS ”), then the operator evaluates to true if it finds an exact match with any member of the set. CONTAINS is supported for lists: When evaluating “a CONTAINS b ”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
- NOT_CONTAINS : Checks for absence of a subsequence, or absence of a value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS ”, “NS ”, or “BS ”), then the operator evaluates to true if it *does not* find an exact match with any member of the set. NOT_CONTAINS is supported for lists: When evaluating “a NOT CONTAINS b ”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
- BEGINS_WITH : Checks for a prefix. *AttributeValueList* can

contain only one *AttributeValue* of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).

- **IN** : Checks for matching elements within two sets. *AttributeValueList* can contain one or more *AttributeValue* elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
- **BETWEEN** : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not compare to { "N" : "6" } . Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }

- **AttributeValueList** (*list*) –

One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *ComparisonOperator* being used.

For type Number, value comparisons are numeric.

String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A , and a is greater than B . For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters .

For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.

For information on specifying data types in JSON, see [JSON Data Format](#) in the *Amazon DynamoDB Developer Guide* .

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –

- **M** (*dict*) –

A Map of attribute values.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **ConditionalOperator** (*string*) –

Warning: This is a legacy parameter, for backward compatibility applications should use *ConditionExpression* instead. Do not combine parameters and expression parameters in a single API call; otherwise namoDB will return a *ValidationException* exception.

A logical operator to apply to the conditions in the *Expected* map:

- AND - If all of the conditions evaluate to true, then the entire map evaluates to true.
- OR - If at least one of the conditions evaluate to true, then the entire map evaluates to true.

If you omit *ConditionalOperator*, then AND is the default.

The operation will succeed only if the entire map evaluates to true.

Note: This parameter does not support attributes of type List or Map.

- **ReturnValues** (*string*) – Use *ReturnValues* if you want to get the item attributes as they appeared before they were deleted. For *DeleteItem*, the valid values are:
 - NONE - If *ReturnValues* is not specified, or if its value is NONE, then nothing is returned. (This setting is the default for *ReturnValues*.)
 - ALL_OLD - The content of the old item is returned.
- **ReturnConsumedCapacity** (*string*) – Determines the level of detail about provisioned throughput consumption that is returned in the response:
 - INDEXES - The response includes the aggregate *ConsumedCapacity* for the operation, together with *ConsumedCapacity* for each table and secondary index that was accessed. Note that some operations, such as *GetItem* and *BatchGetItem*, do not access any indexes at all. In these cases, specifying *INDEXES* will only return *ConsumedCapacity* information for table(s).
 - TOTAL - The response includes only the aggregate *ConsumedCapacity* for the operation.
 - NONE - No *ConsumedCapacity* details are included in the response.
- **ReturnItemCollectionMetrics** (*string*) – Determines whether item collection metrics are returned. If set to *SIZE*, the response includes statistics about item collections, if any, that were modified during the operation are returned in the response. If set to *NONE* (the default), no statistics are returned.
- **ConditionExpression** (*string*) – A condition that must be satisfied in order for a conditional *DeleteItem* to succeed.

An expression can contain any of the following:

- Functions: `attribute_exists` | `attribute_not_exists` | `attribute_type` | `contains` | `begins_with` | `size`
These function names are case-sensitive.
- Comparison operators: `=` | `|` | `<` | `>` | `=` | `=` | `BETWEEN` | `IN`
- Logical operators: `AND` | `OR` | `NOT`

For more information on condition expressions, see [Specifying Conditions](#) in the *Amazon DynamoDB Developer Guide*.

Note: *ConditionExpression* replaces the legacy *ConditionalOperator* and *Expected* parameters.

-
- **ExpressionAttributeNames** (*dict*) – One or more substitution tokens for attribute names in an expression. The following are some use cases for using *ExpressionAttributeNames*:
 - To access an attribute whose name conflicts with a DynamoDB reserved word.
 - To create a placeholder for repeating occurrences of an attribute name in an expression.
 - To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the `#` character in an expression to dereference an attribute name. For example, consider the following attribute name:

- Percentile

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see [Reserved Words](#) in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for *ExpressionAttributeNames*:

- `{"#P": "Percentile"}`

You could then use this substitution in an expression, as in this example:

– #P = :val

Note: Tokens that begin with the `:` character are *expression attribute values*, which are placeholders for the actual value at runtime.

For more information on expression attribute names, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

– (string) –
 * (string) –

- **ExpressionAttributeValues** (*dict*) – One or more values that can be substituted in an expression.

Use the `:` (colon) character in an expression to dereference an attribute value. For example, suppose that you wanted to check whether the value of the *ProductStatus* attribute was one of the following:

```
Available | Backordered | Discontinued
```

You would first need to specify *ExpressionAttributeValues* as follows:

```
{ ":avail":{"S":"Available"}, ":back":{"S":"Backordered"},
  ":disc":{"S":"Discontinued"} }
```

You could then use these values in an expression, such as this:

```
ProductStatus IN (:avail, :back, :disc)
```

For more information on expression attribute values, see [Specifying Conditions](#) in the *Amazon DynamoDB Developer Guide*.

– (string) –
 * (dict) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –
 A String data type.
- **N** (*string*) –
 A Number data type.
- **B** (*bytes*) –
 A Binary data type.
- **SS** (*list*) –
 A String Set data type.
- (*string*) –
- **NS** (*list*) –
 A Number Set data type.
- (*string*) –
- **BS** (*list*) –
 A Binary Set data type.
- (*bytes*) –

- **M** (*dict*) –

A Map of attribute values.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

Return type dict

Returns

Response Syntax

```
{
  'Attributes': {
    'string': {
      'S': 'string',
      'N': 'string',
      'B': b'bytes',
      'SS': [
        'string',
      ],
      'NS': [
        'string',
      ],
      'BS': [
        b'bytes',
      ],
      'M': {
        'string': {'... recursive ...'}
      },
      'L': [
        {'... recursive ...'},
      ],
      'NULL': True|False,
```

```

        'BOOL': True|False
    },
    'ConsumedCapacity': {
        'TableName': 'string',
        'CapacityUnits': 123.0,
        'Table': {
            'CapacityUnits': 123.0
        },
        'LocalSecondaryIndexes': {
            'string': {
                'CapacityUnits': 123.0
            }
        },
        'GlobalSecondaryIndexes': {
            'string': {
                'CapacityUnits': 123.0
            }
        }
    },
    'ItemCollectionMetrics': {
        'ItemCollectionKey': {
            'string': {
                'S': 'string',
                'N': 'string',
                'B': b'bytes',
                'SS': [
                    'string',
                ],
                'NS': [
                    'string',
                ],
                'BS': [
                    b'bytes',
                ],
                'M': {
                    'string': {'... recursive ...'}
                },
                'L': [
                    {'... recursive ...'},
                ],
                'NULL': True|False,
                'BOOL': True|False
            }
        },
        'SizeEstimateRangeGB': [
            123.0,
        ]
    }
}

```

Response Structure

- (*dict*) –

Represents the output of a *DeleteItem* operation.

- **Attributes** (*dict*) –

A map of attribute names to *AttributeValue* objects, representing the item as it appeared before the *DeleteItem* operation. This map appears in the

response only if *ReturnValues* was specified as `ALL_OLD` in the request.

- * *(string)* –

- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** *(string)* –

A String data type.

- **N** *(string)* –

A Number data type.

- **B** *(bytes)* –

A Binary data type.

- **SS** *(list)* –

A String Set data type.

- *(string)* –

- **NS** *(list)* –

A Number Set data type.

- *(string)* –

- **BS** *(list)* –

A Binary Set data type.

- *(bytes)* –

- **M** *(dict)* –

A Map of attribute values.

- *(string)* –

- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** *(list)* –

A List of attribute values.

- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

– **ConsumedCapacity** (*dict*) –

The capacity units consumed by an operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. *ConsumedCapacity* is only returned if the request asked for it. For more information, see [Provisioned Throughput](#) in the *Amazon DynamoDB Developer Guide*.

- * **TableName** (*string*) –

The name of the table that was affected by the operation.

- * **CapacityUnits** (*float*) –

The total number of capacity units consumed by the operation.

- * **Table** (*dict*) –

The amount of throughput consumed on the table affected by the operation.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

- * **LocalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each local index affected by the operation.

- (*string*) –

- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

- * **GlobalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each global index affected by the operation.

- (*string*) –

- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

– **ItemCollectionMetrics** (*dict*) –

Information about item collections, if any, that were affected by the operation. *ItemCollectionMetrics* is only returned if the request asked for it. If the table does not have any local secondary indexes, this information is not returned in the response.

Each *ItemCollectionMetrics* element consists of:

- * *ItemCollectionKey* - The hash key value of the item collection. This is the same as the hash key of the item.
- * *SizeEstimateRange* - An estimate of item collection size, in gigabytes. This value is a two-element array containing a lower bound and an upper bound for the estimate. The estimate includes the size of all the items in the table, plus the size of all attributes projected into all of the local secondary indexes on that table. Use this estimate to measure whether a local secondary index is approaching its size limit. The estimate is subject to change over time; therefore, do not rely on the precision or accuracy of the estimate.

* **ItemCollectionKey** (*dict*) –

The hash key value of the item collection. This value is the same as the hash key of the item.

- (*string*) –
- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –

- **M** (*dict*) –

A Map of attribute values.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one

title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- * **SizeEstimateRangeGB** (*list*) –

An estimate of item collection size, in gigabytes. This value is a two-element array containing a lower bound and an upper bound for the estimate. The estimate includes the size of all the items in the table, plus the size of all attributes projected into all of the local secondary indexes on that table. Use this estimate to measure whether a local secondary index is approaching its size limit.

The estimate is subject to change over time; therefore, do not rely on the precision or accuracy of the estimate.

- (*float*) –

delete_table (***kwargs*)

The *DeleteTable* operation deletes a table and all of its items. After a *DeleteTable* request, the specified table is in the `DELETING` state until DynamoDB completes the deletion. If the table is in the `ACTIVE` state, you can delete it. If a table is in `CREATING` or `UPDATING` states, then DynamoDB returns a *ResourceInUseException*. If the specified table does not exist, DynamoDB returns a *ResourceNotFoundException*. If table is already in the `DELETING` state, no error is returned.

Note: DynamoDB might continue to accept data read and write operations, such as *GetItem* and *PutItem*, on a table in the `DELETING` state until the table deletion is complete.

When you delete a table, any indexes on that table are also deleted.

If you have DynamoDB Streams enabled on the table, then the corresponding stream on that table goes into the `DISABLED` state, and the stream is automatically deleted after 24 hours.

Use the *DescribeTable* API to check the status of the table.

Request Syntax

```
response = client.delete_table(
    TableName='string'
)
```

Parameters **TableName** (*string*) – [REQUIRED]

The name of the table to delete.

Return type dict

Returns

Response Syntax

```
{
  'TableDescription': {
    'AttributeDefinitions': [
      {
        'AttributeName': 'string',
        'AttributeType': 'S'|'N'|'B'
      },
    ],
    'TableName': 'string',
    'KeySchema': [
      {
        'AttributeName': 'string',
        'KeyType': 'HASH'|'RANGE'
      },
    ],
    'TableStatus': 'CREATING'|'UPDATING'|'DELETING'|'ACTIVE',
    'CreationDateTime': datetime(2015, 1, 1),
    'ProvisionedThroughput': {
      'LastIncreaseDateTime': datetime(2015, 1, 1),
      'LastDecreaseDateTime': datetime(2015, 1, 1),
      'NumberOfDecreasesToday': 123,
      'ReadCapacityUnits': 123,
      'WriteCapacityUnits': 123
    },
    'TableSizeBytes': 123,
    'ItemCount': 123,
    'TableArn': 'string',
    'LocalSecondaryIndexes': [
      {
        'IndexName': 'string',
        'KeySchema': [
          {
            'AttributeName': 'string',
            'KeyType': 'HASH'|'RANGE'
          },
        ],
        'Projection': {
          'ProjectionType': 'ALL'|'KEYS_ONLY'|'INCLUDE',
          'NonKeyAttributes': [
            'string',
          ]
        },
        'IndexSizeBytes': 123,
        'ItemCount': 123,
        'IndexArn': 'string'
      },
    ],
    'GlobalSecondaryIndexes': [
      {
        'IndexName': 'string',
        'KeySchema': [
          {
            'AttributeName': 'string',
            'KeyType': 'HASH'|'RANGE'
          },
        ],
        'Projection': {
          'ProjectionType': 'ALL'|'KEYS_ONLY'|'INCLUDE',
          'NonKeyAttributes': [
            'string',
          ]
        },
        'IndexSizeBytes': 123,
        'ItemCount': 123,
        'IndexArn': 'string'
      },
    ],
  },
}
```



```

        },
    ],
    'Projection': {
        'ProjectionType': 'ALL'|'KEYS_ONLY'|'INCLUDE',
        'NonKeyAttributes': [
            'string',
        ]
    },
    'IndexStatus': 'CREATING'|'UPDATING'|'DELETING'|'ACTIVE',
    'Backfilling': True|False,
    'ProvisionedThroughput': {
        'LastIncreaseDateTime': datetime(2015, 1, 1),
        'LastDecreaseDateTime': datetime(2015, 1, 1),
        'NumberOfDecreasesToday': 123,
        'ReadCapacityUnits': 123,
        'WriteCapacityUnits': 123
    },
    'IndexSizeBytes': 123,
    'ItemCount': 123,
    'IndexArn': 'string'
},
],
'StreamSpecification': {
    'StreamEnabled': True|False,
    'StreamViewType': 'NEW_IMAGE'|'OLD_IMAGE'|'NEW_AND_OLD_IMAGES'|'KEYS_ONLY',
},
'LatestStreamLabel': 'string',
'LatestStreamArn': 'string'
}
}

```

Response Structure

- (*dict*) –

Represents the output of a *DeleteTable* operation.

- **TableDescription** (*dict*) –

Represents the properties of a table.

- * **AttributeDefinitions** (*list*) –

An array of *AttributeDefinition* objects. Each of these objects describes one attribute in the table and index key schema.

Each *AttributeDefinition* object in this array is composed of:

- *AttributeName* - The name of the attribute.
- *AttributeType* - The data type for the attribute.

- (*dict*) –

Represents an attribute for describing the key schema for the table and indexes.

- **AttributeName** (*string*) –

A name for the attribute.

- **AttributeType** (*string*) –

The data type for the attribute.

- * **TableName** (*string*) –

The name of the table.

* **KeySchema** (*list*) –

The primary key structure for the table. Each *KeySchemaElement* consists of:

- *AttributeName* - The name of the attribute.
- *KeyType* - The key type for the attribute. Can be either HASH or RANGE .

For more information about primary keys, see [Primary Key](#) in the *Amazon DynamoDB Developer Guide* .

- (*dict*) –

Represents *a single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement* . A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.

- **AttributeName** (*string*) –

The name of a key attribute.

- **KeyType** (*string*) –

The attribute data, consisting of the data type and the attribute value itself.

* **TableStatus** (*string*) –

The current state of the table:

- *CREATING* - The table is being created.
- *UPDATING* - The table is being updated.
- *DELETING* - The table is being deleted.
- *ACTIVE* - The table is ready for use.

* **CreationDateTime** (*datetime*) –

The date and time when the table was created, in [UNIX epoch time](#) format.

* **ProvisionedThroughput** (*dict*) –

The provisioned throughput settings for the table, consisting of read and write capacity units, along with data about increases and decreases.

- **LastIncreaseDateTime** (*datetime*) –

The date and time of the last provisioned throughput increase for this table.

- **LastDecreaseDateTime** (*datetime*) –

The date and time of the last provisioned throughput decrease for this table.

- **NumberOfDecreasesToday** (*integer*) –

The number of provisioned throughput decreases for this table during this UTC calendar day. For current maximums on provisioned throughput decreases, see [Limits](#) in the *Amazon DynamoDB Developer Guide* .

- **ReadCapacityUnits** (*integer*) –

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a *ThrottlingException*. Eventually consistent reads require less effort than strongly consistent reads, so a setting of 50 *ReadCapacityUnits* per second provides 100 eventually consistent *ReadCapacityUnits* per second.

- **WriteCapacityUnits** (*integer*) –

The maximum number of writes consumed per second before DynamoDB returns a *ThrottlingException*.

- * **TableSizeBytes** (*integer*) –

The total size of the specified table, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- * **ItemCount** (*integer*) –

The number of items in the specified table. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- * **TableArn** (*string*) –

The Amazon Resource Name (ARN) that uniquely identifies the table.

- * **LocalSecondaryIndexes** (*list*) –

Represents one or more local secondary indexes on the table. Each index is scoped to a given hash key value. Tables with one or more local secondary indexes are subject to an item collection size limit, where the amount of data within a given item collection cannot exceed 10 GB. Each element is composed of:

- *IndexName* - The name of the local secondary index.
- *KeySchema* - Specifies the complete index key schema. The attribute names in the key schema must be between 1 and 255 characters (inclusive). The key schema must begin with the same hash key attribute as the table.
- *Projection* - Specifies attributes that are copied (projected) from the table into the index. These are in addition to the primary key attributes and index key attributes, which are automatically projected. Each attribute specification is composed of:
 - * *ProjectionType* - One of the following:
 - * **KEYS_ONLY** - Only the index and primary keys are projected into the index.
 - * **INCLUDE** - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes*.
 - * **ALL** - All of the table attributes are projected into the index.
 - * *NonKeyAttributes* - A list of one or more non-key attribute names that are projected into the secondary index. The total count of attributes provided in *NonKeyAttributes*, summed across all of the secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.
- *IndexSizeBytes* - Represents the total size of the index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.
- *ItemCount* - Represents the number of items in the index. Dy-

namoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

If the table is in the `DELETING` state, no information about indexes will be returned.

- *(dict)* –

Represents the properties of a local secondary index.

- **IndexName** (*string*) –

Represents the name of the local secondary index.

- **KeySchema** (*list*) –

The complete index key schema, which consists of one or more pairs of attribute names and key types (`HASH` or `RANGE`).

- *(dict)* –

Represents *a single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement*. A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.

- **AttributeName** (*string*) –

The name of a key attribute.

- **KeyType** (*string*) –

The attribute data, consisting of the data type and the attribute value itself.

- **Projection** (*dict*) –

Represents attributes that are copied (projected) from the table into an index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

- **ProjectionType** (*string*) –

The set of attributes that are projected into the index:

- `KEYS_ONLY` - Only the index and primary keys are projected into the index.

- `INCLUDE` - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes*.

- `ALL` - All of the table attributes are projected into the index.

- **NonKeyAttributes** (*list*) –

Represents the non-key attribute names which will be projected into the index.

For local secondary indexes, the total count of *NonKeyAttributes* summed across all of the local secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.

- (*string*) –
- **IndexSizeBytes** (*integer*) –

The total size of the specified index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- **ItemCount** (*integer*) –

The number of items in the specified index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- **IndexArn** (*string*) –

The Amazon Resource Name (ARN) that uniquely identifies the index.

- * **GlobalSecondaryIndexes** (*list*) –

The global secondary indexes, if any, on the table. Each index is scoped to a given hash key value. Each element is composed of:

- *Backfilling* - If true, then the index is currently in the backfilling phase. Backfilling occurs only when a new global secondary index is added to the table; it is the process by which DynamoDB populates the new index with data from the table. (This attribute does not appear for indexes that were created during a *CreateTable* operation.)
- *IndexName* - The name of the global secondary index.
- *IndexSizeBytes* - The total size of the global secondary index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.
- *IndexStatus* - The current status of the global secondary index:
 - * *CREATING* - The index is being created.
 - * *UPDATING* - The index is being updated.
 - * *DELETING* - The index is being deleted.
 - * *ACTIVE* - The index is ready for use.
- *ItemCount* - The number of items in the global secondary index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.
- *KeySchema* - Specifies the complete index key schema. The attribute names in the key schema must be between 1 and 255 characters (inclusive). The key schema must begin with the same hash key attribute as the table.
- *Projection* - Specifies attributes that are copied (projected) from the table into the index. These are in addition to the primary key attributes and index key attributes, which are automatically projected. Each attribute specification is composed of:
 - * *ProjectionType* - One of the following:
 - * *KEYS_ONLY* - Only the index and primary keys are projected into the index.
 - * *INCLUDE* - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes*.
 - * *ALL* - All of the table attributes are projected into the index.
- *NonKeyAttributes* - A list of one or more non-key attribute names that are projected into the secondary index. The total count of attributes provided in *NonKeyAttributes*, summed across all of the secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this

counts as two distinct attributes when determining the total.

- *ProvisionedThroughput* - The provisioned throughput settings for the global secondary index, consisting of read and write capacity units, along with data about increases and decreases.

If the table is in the `DELETING` state, no information about indexes will be returned.

- (*dict*) –

Represents the properties of a global secondary index.

- **IndexName** (*string*) –

The name of the global secondary index.

- **KeySchema** (*list*) –

The complete key schema for the global secondary index, consisting of one or more pairs of attribute names and key types (`HASH` or `RANGE`).

- (*dict*) –

Represents *a single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement*. A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.

- **AttributeName** (*string*) –

The name of a key attribute.

- **KeyType** (*string*) –

The attribute data, consisting of the data type and the attribute value itself.

- **Projection** (*dict*) –

Represents attributes that are copied (projected) from the table into an index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

- **ProjectionType** (*string*) –

The set of attributes that are projected into the index:

- `KEYS_ONLY` - Only the index and primary keys are projected into the index.
- `INCLUDE` - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes*.
- `ALL` - All of the table attributes are projected into the index.
- **NonKeyAttributes** (*list*) –

Represents the non-key attribute names which will be projected into the index.

For local secondary indexes, the total count of *NonKeyAttributes* summed across all of the local secondary indexes, must not exceed 20. If you project the same attribute into two

different indexes, this counts as two distinct attributes when determining the total.

- *(string)* –
- **IndexStatus** (*string*) –

The current state of the global secondary index:

- *CREATING* - The index is being created.
- *UPDATING* - The index is being updated.
- *DELETING* - The index is being deleted.
- *ACTIVE* - The index is ready for use.
- **Backfilling** (*boolean*) –

Indicates whether the index is currently backfilling. *Backfilling* is the process of reading items from the table and determining whether they can be added to the index. (Not all items will qualify: For example, a hash key attribute cannot have any duplicates.) If an item can be added to the index, DynamoDB will do so. After all items have been processed, the backfilling operation is complete and *Backfilling* is false.

Note: For indexes that were created during a *CreateTable* operation, the *Backfilling* attribute does not appear in the *DescribeTable* output.

- **ProvisionedThroughput** (*dict*) –

Represents the provisioned throughput settings for the table, consisting of read and write capacity units, along with data about increases and decreases.

- **LastIncreaseDateTime** (*datetime*) –

The date and time of the last provisioned throughput increase for this table.

- **LastDecreaseDateTime** (*datetime*) –

The date and time of the last provisioned throughput decrease for this table.

- **NumberOfDecreasesToday** (*integer*) –

The number of provisioned throughput decreases for this table during this UTC calendar day. For current maximums on provisioned throughput decreases, see [Limits](#) in the *Amazon DynamoDB Developer Guide*.

- **ReadCapacityUnits** (*integer*) –

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a *ThrottlingException*. Eventually consistent reads require less effort than strongly consistent reads, so a setting of 50 *ReadCapacityUnits* per second provides 100 eventually consistent *ReadCapacityUnits* per second.

- **WriteCapacityUnits** (*integer*) –

The maximum number of writes consumed per second before DynamoDB returns a *ThrottlingException*.

- **IndexSizeBytes** (*integer*) –

The total size of the specified index, in bytes. DynamoDB updates this value approximately every six hours. Recent

changes might not be reflected in this value.

- **ItemCount** (*integer*) –

The number of items in the specified index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- **IndexArn** (*string*) –

The Amazon Resource Name (ARN) that uniquely identifies the index.

- * **StreamSpecification** (*dict*) –

The current DynamoDB Streams configuration for the table.

- **StreamEnabled** (*boolean*) –

Indicates whether DynamoDB Streams is enabled (true) or disabled (false) on the table.

- **StreamViewType** (*string*) –

The DynamoDB Streams settings for the table. These settings consist of:

- *StreamEnabled* - Indicates whether DynamoDB Streams is enabled (true) or disabled (false) on the table.
- *StreamViewType* - When an item in the table is modified, *StreamViewType* determines what information is written to the stream for this table. Valid values for *StreamViewType* are: * *KEYS_ONLY* - Only the key attributes of the modified item are written to the stream.
- *NEW_IMAGE* - The entire item, as it appears after it was modified, is written to the stream.
- *OLD_IMAGE* - The entire item, as it appeared before it was modified, is written to the stream.
- *NEW_AND_OLD_IMAGES* - Both the new and the old item images of the item are written to the stream.

- * **LatestStreamLabel** (*string*) –

A timestamp, in ISO 8601 format, for this stream.

Note that *LatestStreamLabel* is not a unique identifier for the stream, because it is possible that a stream from another table might have the same timestamp. However, the combination of the following three elements is guaranteed to be unique:

- the AWS customer ID.
- the table name.
- the *StreamLabel*.

- * **LatestStreamArn** (*string*) –

The Amazon Resource Name (ARN) that uniquely identifies the latest stream for this table.

describe_table (***kwargs*)

Returns information about the table, including the current status of the table, when it was created, the primary key schema, and any indexes on the table.

Note: If you issue a DescribeTable request immediately after a CreateTable request, DynamoDB might return a ResourceNotFoundException. This is because DescribeTable uses an eventually consistent query, and the metadata for your table might not be available at that moment. Wait for a few seconds, and then try the DescribeTable request again.

Request Syntax

```
response = client.describe_table(
    TableName='string'
)
```

Parameters *TableName* (*string*) – **[REQUIRED]**

The name of the table to describe.

Return type dict

Returns

Response Syntax

```
{
    'Table': {
        'AttributeDefinitions': [
            {
                'AttributeName': 'string',
                'AttributeType': 'S'|'N'|'B'
            },
        ],
        'TableName': 'string',
        'KeySchema': [
            {
                'AttributeName': 'string',
                'KeyType': 'HASH'|'RANGE'
            },
        ],
        'TableStatus': 'CREATING'|'UPDATING'|'DELETING'|'ACTIVE',
        'CreationDateTime': datetime(2015, 1, 1),
        'ProvisionedThroughput': {
            'LastIncreaseDateTime': datetime(2015, 1, 1),
            'LastDecreaseDateTime': datetime(2015, 1, 1),
            'NumberOfDecreasesToday': 123,
            'ReadCapacityUnits': 123,
            'WriteCapacityUnits': 123
        },
        'TableSizeBytes': 123,
        'ItemCount': 123,
        'TableArn': 'string',
        'LocalSecondaryIndexes': [
            {
                'IndexName': 'string',
                'KeySchema': [
                    {
                        'AttributeName': 'string',
                        'KeyType': 'HASH'|'RANGE'
                    },
                ],
                'Projection': {
                    'ProjectionType': 'ALL'|'KEYS_ONLY'|'INCLUDE',
                    'NonKeyAttributes': [
                        'string',
                    ]
                },
                'IndexSizeBytes': 123,
                'ItemCount': 123,
                'IndexArn': 'string'
            },
        ],
    },
}
```

```
    },
],
'GlobalSecondaryIndexes': [
    {
        'IndexName': 'string',
        'KeySchema': [
            {
                'AttributeName': 'string',
                'KeyType': 'HASH' | 'RANGE'
            },
        ],
        'Projection': {
            'ProjectionType': 'ALL' | 'KEYS_ONLY' | 'INCLUDE',
            'NonKeyAttributes': [
                'string',
            ]
        },
        'IndexStatus': 'CREATING' | 'UPDATING' | 'DELETING' | 'ACTIVE',
        'Backfilling': True | False,
        'ProvisionedThroughput': {
            'LastIncreaseDateTime': datetime(2015, 1, 1),
            'LastDecreaseDateTime': datetime(2015, 1, 1),
            'NumberOfDecreasesToday': 123,
            'ReadCapacityUnits': 123,
            'WriteCapacityUnits': 123
        },
        'IndexSizeBytes': 123,
        'ItemCount': 123,
        'IndexArn': 'string'
    },
],
'StreamSpecification': {
    'StreamEnabled': True | False,
    'StreamViewType': 'NEW_IMAGE' | 'OLD_IMAGE' | 'NEW_AND_OLD_IMAGES' | 'KEYS_ONLY',
},
'LatestStreamLabel': 'string',
'LatestStreamArn': 'string'
}
```

Response Structure

- (dict) –

Represents the output of a *DescribeTable* operation.

- **Table** (dict) –

Represents the properties of a table.

- * **AttributeDefinitions** (list) –

An array of *AttributeDefinition* objects. Each of these objects describes one attribute in the table and index key schema.

Each *AttributeDefinition* object in this array is composed of:

- *AttributeName* - The name of the attribute.
- *AttributeType* - The data type for the attribute.

- (dict) –

Represents an attribute for describing the key schema for the table and indexes.

- **AttributeName** (*string*) –
A name for the attribute.
- **AttributeType** (*string*) –
The data type for the attribute.
- * **TableName** (*string*) –
The name of the table.
- * **KeySchema** (*list*) –
The primary key structure for the table. Each *KeySchemaElement* consists of:
 - *AttributeName* - The name of the attribute.
 - *KeyType* - The key type for the attribute. Can be either `HASH` or `RANGE`.
 For more information about primary keys, see [Primary Key](#) in the *Amazon DynamoDB Developer Guide*.
 - (*dict*) –
Represents a *single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement*. A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.
 - **AttributeName** (*string*) –
The name of a key attribute.
 - **KeyType** (*string*) –
The attribute data, consisting of the data type and the attribute value itself.
- * **TableStatus** (*string*) –
The current state of the table:
 - *CREATING* - The table is being created.
 - *UPDATING* - The table is being updated.
 - *DELETING* - The table is being deleted.
 - *ACTIVE* - The table is ready for use.
- * **CreationDateTime** (*datetime*) –
The date and time when the table was created, in [UNIX epoch time](#) format.
- * **ProvisionedThroughput** (*dict*) –
The provisioned throughput settings for the table, consisting of read and write capacity units, along with data about increases and decreases.
 - **LastIncreaseDateTime** (*datetime*) –
The date and time of the last provisioned throughput increase for this table.
 - **LastDecreaseDateTime** (*datetime*) –

The date and time of the last provisioned throughput decrease for this table.

- **NumberOfDecreasesToday** (*integer*) –

The number of provisioned throughput decreases for this table during this UTC calendar day. For current maximums on provisioned throughput decreases, see [Limits](#) in the *Amazon DynamoDB Developer Guide*.

- **ReadCapacityUnits** (*integer*) –

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a *ThrottlingException*. Eventually consistent reads require less effort than strongly consistent reads, so a setting of 50 *ReadCapacityUnits* per second provides 100 eventually consistent *ReadCapacityUnits* per second.

- **WriteCapacityUnits** (*integer*) –

The maximum number of writes consumed per second before DynamoDB returns a *ThrottlingException*.

- * **TableSizeBytes** (*integer*) –

The total size of the specified table, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- * **ItemCount** (*integer*) –

The number of items in the specified table. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- * **TableArn** (*string*) –

The Amazon Resource Name (ARN) that uniquely identifies the table.

- * **LocalSecondaryIndexes** (*list*) –

Represents one or more local secondary indexes on the table. Each index is scoped to a given hash key value. Tables with one or more local secondary indexes are subject to an item collection size limit, where the amount of data within a given item collection cannot exceed 10 GB. Each element is composed of:

- *IndexName* - The name of the local secondary index.
- *KeySchema* - Specifies the complete index key schema. The attribute names in the key schema must be between 1 and 255 characters (inclusive). The key schema must begin with the same hash key attribute as the table.
- *Projection* - Specifies attributes that are copied (projected) from the table into the index. These are in addition to the primary key attributes and index key attributes, which are automatically projected. Each attribute specification is composed of: * *ProjectionType* - One of the following: * *KEYS_ONLY* - Only the index and primary keys are projected into the index.
- *INCLUDE* - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes*.
- *ALL* - All of the table attributes are projected into the index.
- *NonKeyAttributes* - A list of one or more non-key attribute

names that are projected into the secondary index. The total count of attributes provided in *NonKeyAttributes* , summed across all of the secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.

- *IndexSizeBytes* - Represents the total size of the index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.
- *ItemCount* - Represents the number of items in the index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

If the table is in the `DELETING` state, no information about indexes will be returned.

- *(dict)* –

Represents the properties of a local secondary index.

- **IndexName** (*string*) –

Represents the name of the local secondary index.

- **KeySchema** (*list*) –

The complete index key schema, which consists of one or more pairs of attribute names and key types (`HASH` or `RANGE`).

- *(dict)* –

Represents *a single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement* . A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.

- **AttributeName** (*string*) –

The name of a key attribute.

- **KeyType** (*string*) –

The attribute data, consisting of the data type and the attribute value itself.

- **Projection** (*dict*) –

Represents attributes that are copied (projected) from the table into an index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

- **ProjectionType** (*string*) –

The set of attributes that are projected into the index:

- `KEYS_ONLY` - Only the index and primary keys are projected into the index.
- `INCLUDE` - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes* .
- `ALL` - All of the table attributes are projected into the index.

- **NonKeyAttributes** (*list*) –

Represents the non-key attribute names which will be projected into the index.

For local secondary indexes, the total count of *NonKeyAttributes* summed across all of the local secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.

- (*string*) –

- **IndexSizeBytes** (*integer*) –

The total size of the specified index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- **ItemCount** (*integer*) –

The number of items in the specified index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- **IndexArn** (*string*) –

The Amazon Resource Name (ARN) that uniquely identifies the index.

- * **GlobalSecondaryIndexes** (*list*) –

The global secondary indexes, if any, on the table. Each index is scoped to a given hash key value. Each element is composed of:

- *Backfilling* - If true, then the index is currently in the backfilling phase. Backfilling occurs only when a new global secondary index is added to the table; it is the process by which DynamoDB populates the new index with data from the table. (This attribute does not appear for indexes that were created during a *CreateTable* operation.)
- *IndexName* - The name of the global secondary index.
- *IndexSizeBytes* - The total size of the global secondary index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.
- *IndexStatus* - The current status of the global secondary index:
 - * *CREATING* - The index is being created.
 - * *UPDATING* - The index is being updated.
 - * *DELETING* - The index is being deleted.
 - * *ACTIVE* - The index is ready for use.
- *ItemCount* - The number of items in the global secondary index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.
- *KeySchema* - Specifies the complete index key schema. The attribute names in the key schema must be between 1 and 255 characters (inclusive). The key schema must begin with the same hash key attribute as the table.
- *Projection* - Specifies attributes that are copied (projected) from the table into the index. These are in addition to the primary key attributes and index key attributes, which are automatically projected. Each attribute specification is composed of:
 - * *ProjectionType* - One of the following:
 - * *KEYS_ONLY* - Only the index and primary keys are projected into the index.

- **INCLUDE** - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes*.
- **ALL** - All of the table attributes are projected into the index.
- **NonKeyAttributes** - A list of one or more non-key attribute names that are projected into the secondary index. The total count of attributes provided in *NonKeyAttributes*, summed across all of the secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.
- **ProvisionedThroughput** - The provisioned throughput settings for the global secondary index, consisting of read and write capacity units, along with data about increases and decreases.

If the table is in the `DELETING` state, no information about indexes will be returned.

- *(dict)* –

Represents the properties of a global secondary index.

- **IndexName** (*string*) –

The name of the global secondary index.

- **KeySchema** (*list*) –

The complete key schema for the global secondary index, consisting of one or more pairs of attribute names and key types (`HASH` or `RANGE`).

- *(dict)* –

Represents *a single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement*. A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.

- **AttributeName** (*string*) –

The name of a key attribute.

- **KeyType** (*string*) –

The attribute data, consisting of the data type and the attribute value itself.

- **Projection** (*dict*) –

Represents attributes that are copied (projected) from the table into an index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

- **ProjectionType** (*string*) –

The set of attributes that are projected into the index:

- **KEYS_ONLY** - Only the index and primary keys are projected into the index.
- **INCLUDE** - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKey-*

Attributes .

- **ALL** - All of the table attributes are projected into the index.
- **NonKeyAttributes** (*list*) –

Represents the non-key attribute names which will be projected into the index.

For local secondary indexes, the total count of *NonKeyAttributes* summed across all of the local secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.

- (*string*) –
- **IndexStatus** (*string*) –

The current state of the global secondary index:

- **CREATING** - The index is being created.
- **UPDATING** - The index is being updated.
- **DELETING** - The index is being deleted.
- **ACTIVE** - The index is ready for use.
- **Backfilling** (*boolean*) –

Indicates whether the index is currently backfilling. *Backfilling* is the process of reading items from the table and determining whether they can be added to the index. (Not all items will qualify: For example, a hash key attribute cannot have any duplicates.) If an item can be added to the index, DynamoDB will do so. After all items have been processed, the backfilling operation is complete and *Backfilling* is false.

Note: For indexes that were created during a *CreateTable* operation, the *Backfilling* attribute does not appear in the *DescribeTable* output.

- **ProvisionedThroughput** (*dict*) –

Represents the provisioned throughput settings for the table, consisting of read and write capacity units, along with data about increases and decreases.

- **LastIncreaseDateTime** (*datetime*) –

The date and time of the last provisioned throughput increase for this table.

- **LastDecreaseDateTime** (*datetime*) –

The date and time of the last provisioned throughput decrease for this table.

- **NumberOfDecreasesToday** (*integer*) –

The number of provisioned throughput decreases for this table during this UTC calendar day. For current maximums on provisioned throughput decreases, see [Limits](#) in the *Amazon DynamoDB Developer Guide* .

- **ReadCapacityUnits** (*integer*) –

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a *ThrottlingException* . Eventually consistent reads require less effort than strongly consistent reads, so a setting of 50 *ReadCapacityUnits* per

second provides 100 eventually consistent *ReadCapacityUnits* per second.

- **WriteCapacityUnits** (*integer*) –

The maximum number of writes consumed per second before DynamoDB returns a *ThrottlingException*.

- **IndexSizeBytes** (*integer*) –

The total size of the specified index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- **ItemCount** (*integer*) –

The number of items in the specified index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- **IndexArn** (*string*) –

The Amazon Resource Name (ARN) that uniquely identifies the index.

- * **StreamSpecification** (*dict*) –

The current DynamoDB Streams configuration for the table.

- **StreamEnabled** (*boolean*) –

Indicates whether DynamoDB Streams is enabled (true) or disabled (false) on the table.

- **StreamViewType** (*string*) –

The DynamoDB Streams settings for the table. These settings consist of:

- *StreamEnabled* - Indicates whether DynamoDB Streams is enabled (true) or disabled (false) on the table.
- *StreamViewType* - When an item in the table is modified, *StreamViewType* determines what information is written to the stream for this table. Valid values for *StreamViewType* are: * *KEYS_ONLY* - Only the key attributes of the modified item are written to the stream.
- *NEW_IMAGE* - The entire item, as it appears after it was modified, is written to the stream.
- *OLD_IMAGE* - The entire item, as it appeared before it was modified, is written to the stream.
- *NEW_AND_OLD_IMAGES* - Both the new and the old item images of the item are written to the stream.

- * **LatestStreamLabel** (*string*) –

A timestamp, in ISO 8601 format, for this stream.

Note that *LatestStreamLabel* is not a unique identifier for the stream, because it is possible that a stream from another table might have the same timestamp. However, the combination of the following three elements is guaranteed to be unique:

- the AWS customer ID.
- the table name.
- the *StreamLabel*.

- * **LatestStreamArn** (*string*) –

The Amazon Resource Name (ARN) that uniquely identifies the latest stream for this table.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to *ClientMethod*.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_item (***kwargs*)

The *GetItem* operation returns a set of attributes for the item with the given primary key. If there is no matching item, *GetItem* does not return any data.

GetItem provides an eventually consistent read by default. If your application requires a strongly consistent read, set *ConsistentRead* to `true`. Although a strongly consistent read might take more time than an eventually consistent read, it always returns the last updated value.

Request Syntax

```
response = client.get_item(
    TableName='string',
    Key={
        'string': {
            'S': 'string',
            'N': 'string',
            'B': b'bytes',
            'SS': [
                'string',
            ],
            'NS': [
                'string',
            ],
            'BS': [
                b'bytes',
            ],
            'M': {
                'string': {'... recursive ...'}
            },
            'L': [
                {'... recursive ...'},
            ],
            'NULL': True|False,
            'BOOL': True|False
        }
    },
    AttributesToGet=[
        'string',
    ],
    ConsistentRead=True|False,
    ReturnConsumedCapacity='INDEXES'|'TOTAL'|'NONE',
    ProjectionExpression='string',
    ExpressionAttributeNames={
        'string': 'string'
    }
)
```

Parameters

- **TableName** (*string*) – **[REQUIRED]**

The name of the table containing the requested item.

- **Key** (*dict*) – **[REQUIRED]**

A map of attribute names to *AttributeValue* objects, representing the primary key of the item to retrieve.

For the primary key, you must provide all of the attributes. For example, with a hash type primary key, you only need to provide the hash attribute. For a hash-and-range type primary key, you must provide both the hash attribute and the range attribute.

- (*string*) –
- * (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –

- **M** (*dict*) –

A Map of attribute values.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

Warning: This is a legacy parameter, for backward compatibility. New applications should use *ProjectionExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception.

This parameter allows you to retrieve attributes of type List or Map; however, it cannot retrieve individual elements within a List or a Map.

- **AttributesToGet** (*list*) –

The names of one or more attributes to retrieve. If no attribute names are provided, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

Note that *AttributesToGet* has no effect on provisioned throughput consumption. DynamoDB determines capacity units consumed based on item size, not on the amount of data that is returned to an application.

- (*string*) –

- **ConsistentRead** (*boolean*) – Determines the read consistency model: If set to `true`, then the operation uses strongly consistent reads; otherwise, the operation uses eventually consistent reads.
- **ReturnConsumedCapacity** (*string*) – Determines the level of detail about provisioned throughput consumption that is returned in the response:
 - **INDEXES** - The response includes the aggregate *ConsumedCapacity* for the operation, together with *ConsumedCapacity* for each table and secondary index that was accessed. Note that some operations, such as *GetItem* and *BatchGetItem*, do not access any indexes at all. In these cases, specifying **INDEXES** will only return *ConsumedCapacity* information for table(s).
 - **TOTAL** - The response includes only the aggregate *ConsumedCapacity* for the operation.
 - **NONE** - No *ConsumedCapacity* details are included in the response.
- **ProjectionExpression** (*string*) – A string that identifies one or more attributes to retrieve from the table. These attributes can include scalars, sets, or elements of a JSON document. The attributes in the expression must be separated by commas.

If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

For more information, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

Note: *ProjectionExpression* replaces the legacy *AttributesToGet* parameter.

-
- **ExpressionAttributeNames** (*dict*) – One or more substitution tokens for attribute names in an expression. The following are some use cases for using *ExpressionAttributeNames* :

- To access an attribute whose name conflicts with a DynamoDB reserved word.
- To create a placeholder for repeating occurrences of an attribute name in an expression.
- To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the # character in an expression to dereference an attribute name. For example, consider the following attribute name:

- Percentile

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see [Reserved Words](#) in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for *ExpressionAttributeNames* :

- { "#P": "Percentile" }

You could then use this substitution in an expression, as in this example:

- #P = :val

Note: Tokens that begin with the : character are *expression attribute values* , which are placeholders for the actual value at runtime.

For more information on expression attribute names, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide* .

- (*string*) –
- * (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'Item': {
    'string': {
      'S': 'string',
      'N': 'string',
      'B': b'bytes',
      'SS': [
        'string',
      ],
      'NS': [
        'string',
      ],
      'BS': [
        b'bytes',
      ],
      'M': {
        'string': {'... recursive ...'}
      },
      'L': [
        {'... recursive ...'},
      ],
      'NULL': True|False,
      'BOOL': True|False
    }
  }
```

```
    },
    'ConsumedCapacity': {
        'TableName': 'string',
        'CapacityUnits': 123.0,
        'Table': {
            'CapacityUnits': 123.0
        },
        'LocalSecondaryIndexes': {
            'string': {
                'CapacityUnits': 123.0
            }
        },
        'GlobalSecondaryIndexes': {
            'string': {
                'CapacityUnits': 123.0
            }
        }
    }
}
```

Response Structure

- *(dict)* –

Represents the output of a *GetItem* operation.

- **Item** (*dict*) –

A map of attribute names to *AttributeValue* objects, as specified by *AttributesToGet*.

- * (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –

- **M** (*dict*) –

A Map of attribute values.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **ConsumedCapacity** (*dict*) –

The capacity units consumed by an operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. *ConsumedCapacity* is only returned if the request asked for it. For more information, see [Provisioned Throughput](#) in the *Amazon DynamoDB Developer Guide*.

- * **TableName** (*string*) –

The name of the table that was affected by the operation.

- * **CapacityUnits** (*float*) –

The total number of capacity units consumed by the operation.

- * **Table** (*dict*) –

The amount of throughput consumed on the table affected by the operation.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

- * **LocalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each local index affected by the operation.

- (*string*) –

- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

- * **GlobalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each global index affected by the operation.

- (*string*) –
- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_waiter (*waiter_name*)

list_tables (***kwargs*)

Returns an array of table names associated with the current account and endpoint. The output from *ListTables* is paginated, with each page returning a maximum of 100 table names.

Request Syntax

```
response = client.list_tables(  
    ExclusiveStartTableName='string',  
    Limit=123  
)
```

Parameters

- **ExclusiveStartTableName** (*string*) – The first table name that this operation will evaluate. Use the value that was returned for *LastEvaluatedTableName* in a previous operation, so that you can obtain the next page of results.
- **Limit** (*integer*) – A maximum number of table names to return. If this parameter is not specified, the limit is 100.

Return type `dict`

Returns

Response Syntax

```
{  
    'TableNames': [  
        'string',
```



```

],
  'LastEvaluatedTableName': 'string'
}

```

Response Structure

- (*dict*) –

Represents the output of a *ListTables* operation.

- **TableNames** (*list*) –

The names of the tables associated with the current account at the current endpoint. The maximum size of this array is 100.

If *LastEvaluatedTableName* also appears in the output, you can use this value as the *ExclusiveStartTableName* parameter in a subsequent *ListTables* request and obtain the next page of results.

* (*string*) –

- **LastEvaluatedTableName** (*string*) –

The name of the last table in the current page of results. Use this value as the *ExclusiveStartTableName* in a new request to obtain the next page of results, until all the table names are returned.

If you do not receive a *LastEvaluatedTableName* value in the response, this means that there are no more table names to be retrieved.

put_item (**kwargs)

Creates a new item, or replaces an old item with a new item. If an item that has the same primary key as the new item already exists in the specified table, the new item completely replaces the existing item. You can perform a conditional put operation (add a new item if one with the specified primary key doesn't exist), or replace an existing item if it has certain attribute values.

In addition to putting an item, you can also return the item's attribute values in the same operation, using the *ReturnValues* parameter.

When you add an item, the primary key attribute(s) are the only required attributes. Attribute values cannot be null. String and Binary type attributes must have lengths greater than zero. Set type attributes cannot be empty. Requests with empty values will be rejected with a *ValidationException* exception.

You can request that *PutItem* return either a copy of the original item (before the update) or a copy of the updated item (after the update). For more information, see the *ReturnValues* description below.

Note: To prevent a new item from replacing an existing item, use a conditional put operation with *ComparisonOperator* set to NULL for the primary key attribute, or attributes.

For more information about using this API, see [Working with Items](#) in the *Amazon DynamoDB Developer Guide*.

Request Syntax

```

response = client.put_item(
    TableName='string',
    Item={
        'string': {
            'S': 'string',
            'N': 'string',
            'B': b'bytes',
            'SS': [
                'string',

```

```
    ],
    'NS': [
        'string',
    ],
    'BS': [
        b'bytes',
    ],
    'M': {
        'string': {'... recursive ...'}
    },
    'L': [
        {'... recursive ...'},
    ],
    'NULL': True|False,
    'BOOL': True|False
}
},
Expected={
    'string': {
        'Value': {
            'S': 'string',
            'N': 'string',
            'B': b'bytes',
            'SS': [
                'string',
            ],
            'NS': [
                'string',
            ],
            'BS': [
                b'bytes',
            ],
            'M': {
                'string': {'... recursive ...'}
            },
            'L': [
                {'... recursive ...'},
            ],
            'NULL': True|False,
            'BOOL': True|False
        },
        'Exists': True|False,
        'ComparisonOperator': 'EQ' | 'NE' | 'IN' | 'LE' | 'LT' | 'GE' | 'GT' | 'BETWEEN' | 'NOT_NULL' | '...',
        'AttributeValueList': [
            {
                'S': 'string',
                'N': 'string',
                'B': b'bytes',
                'SS': [
                    'string',
                ],
                'NS': [
                    'string',
                ],
                'BS': [
                    b'bytes',
                ],
                'M': {
```

```

        'string': {'... recursive ...'}
    },
    'L': [
        {'... recursive ...'},
    ],
    'NULL': True|False,
    'BOOL': True|False
},
]
}
},
ReturnValues='NONE'|'ALL_OLD'|'UPDATED_OLD'|'ALL_NEW'|'UPDATED_NEW',
ReturnConsumedCapacity='INDEXES'|'TOTAL'|'NONE',
ReturnItemCollectionMetrics='SIZE'|'NONE',
ConditionalOperator='AND'|'OR',
ConditionExpression='string',
ExpressionAttributeNames={
    'string': 'string'
},
ExpressionAttributeValues={
    'string': {
        'S': 'string',
        'N': 'string',
        'B': b'bytes',
        'SS': [
            'string',
        ],
        'NS': [
            'string',
        ],
        'BS': [
            b'bytes',
        ],
        'M': {
            'string': {'... recursive ...'}
        },
        'L': [
            {'... recursive ...'},
        ],
        'NULL': True|False,
        'BOOL': True|False
    }
}
)

```

Parameters

- **TableName** (*string*) – [REQUIRED]

The name of the table to contain the item.

- **Item** (*dict*) – [REQUIRED]

A map of attribute name/value pairs, one for each attribute. Only the primary key attributes are required; you can optionally provide other attribute name-value pairs for the item.

You must provide all of the attributes for the primary key. For example, with a hash type primary key, you only need to provide the hash attribute. For a hash-and-range type primary key, you must provide both the hash attribute and the range attribute.

If you specify any attributes that are part of an index key, then the data types for those attributes must match those of the schema in the table's attribute definition.

For more information about primary keys, see [Primary Key](#) in the *Amazon DynamoDB Developer Guide*.

Each element in the *Item* map is an *AttributeValue* object.

- (*string*) -
- * (*dict*) -

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) -

A String data type.

- **N** (*string*) -

A Number data type.

- **B** (*bytes*) -

A Binary data type.

- **SS** (*list*) -

A String Set data type.

- (*string*) -

- **NS** (*list*) -

A Number Set data type.

- (*string*) -

- **BS** (*list*) -

A Binary Set data type.

- (*bytes*) -

- **M** (*dict*) -

A Map of attribute values.

- (*string*) -

- (*dict*) -

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) -

A List of attribute values.

- (*dict*) -

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **Expected** (*dict*) –

Warning: This is a legacy parameter, for backward compatibility. New applications should use *ConditionExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception.

A map of attribute/condition pairs. *Expected* provides a conditional block for the *PutItem* operation.

Note: This parameter does not support attributes of type List or Map.

Each element of *Expected* consists of an attribute name, a comparison operator, and one or more values. DynamoDB compares the attribute with the value(s) you supplied, using the comparison operator. For each *Expected* element, the result of the evaluation is either true or false.

If you specify more than one element in the *Expected* map, then by default all of the conditions must evaluate to true. In other words, the conditions are ANDed together. (You can use the *ConditionalOperator* parameter to OR the conditions instead. If you do this, then at least one of the conditions must evaluate to true, rather than all of them.)

If the *Expected* map evaluates to true, then the conditional operation succeeds; otherwise, it fails.

Expected contains the following:

- *AttributeValueList* - One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *ComparisonOperator* being used. For type Number, value comparisons are numeric. String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A, and a is greater than B. For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters. For type Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.
- *ComparisonOperator* - A comparator for evaluating attributes in the *AttributeValueList*. When performing the comparison, DynamoDB uses strongly consistent reads. The following comparison operators are available: EQ | NE | LE | LT | GE | GT | NOT_NULL | NULL | CONTAINS | NOT_CONTAINS | BEGINS_WITH | IN | BETWEEN. The following are descriptions of each comparison operator. * EQ : Equal. EQ is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* element of a different type than

the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not equal `{"NS": ["6", "2", "1"]}`.

- **NE** : Not equal. **NE** is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not equal `{"NS": ["6", "2", "1"]}`.
- **LE** : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
- **LT** : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
- **GE** : Greater than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
- **GT** : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
- **NOT_NULL** : The attribute exists. **NOT_NULL** is supported for all datatypes, including lists and maps. .. note::This operator tests for the existence of an attribute, not its data type. If the data type of attribute “a” is null, and you evaluate it using **NOT_NULL**, the result is a Boolean *true*. This result is because the attribute “a” exists; its data type is not relevant to the **NOT_NULL** comparison operator.
- **NULL** : The attribute does not exist. **NULL** is supported for all datatypes, including lists and maps. .. note::This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute “a” is null, and you evaluate it using **NULL**, the result is a Boolean *false*. This is because the attribute “a” exists; its data type is not relevant to the **NULL** comparison operator.
- **CONTAINS** : Checks for a subsequence, or value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS”, “NS”, or “BS”), then the operator evaluates to true if it finds an exact match with any member of the set.

CONTAINS is supported for lists: When evaluating “a CONTAINS b”, “a” can be a list; however, “b” cannot be a set, a map, or a list.

- NOT_CONTAINS : Checks for absence of a subsequence, or absence of a value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS”, “NS”, or “BS”), then the operator evaluates to true if it *does not* find an exact match with any member of the set. NOT_CONTAINS is supported for lists: When evaluating “a NOT CONTAINS b”, “a” can be a list; however, “b” cannot be a set, a map, or a list.
- BEGINS_WITH : Checks for a prefix. *AttributeValueList* can contain only one *AttributeValue* of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).
- IN : Checks for matching elements within two sets. *AttributeValueList* can contain one or more *AttributeValue* elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
- BETWEEN : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, {“S”: “6”} does not compare to {“N”: “6”}. Also, {“N”: “6”} does not compare to {“NS”: [“6”, “2”, “1”]}

For usage examples of *AttributeValueList* and *ComparisonOperator*, see [Legacy Conditional Parameters](#) in the *Amazon DynamoDB Developer Guide*.

For backward compatibility with previous DynamoDB releases, the following parameters can be used instead of *AttributeValueList* and *ComparisonOperator*:

- *Value* - A value for DynamoDB to compare with an attribute.
- *Exists* - A Boolean value that causes DynamoDB to evaluate the value before attempting the conditional operation: * If *Exists* is `true`, DynamoDB will check to see if that attribute value already exists in the table. If it is found, then the condition evaluates to true; otherwise the condition evaluates to false.
- If *Exists* is `false`, DynamoDB assumes that the attribute value does *not* exist in the table. If in fact the value does not exist, then the assumption is valid and the condition evaluates to true. If the value is found, despite the assumption that it does not exist, the condition evaluates to false.

Note that the default value for *Exists* is `true`.

The *Value* and *Exists* parameters are incompatible with *AttributeValueList* and *ComparisonOperator*. Note that if you use both sets of parameters at once, DynamoDB will return a *ValidationException* exception.

- (string) –
- * (dict) –

Represents a condition to be compared with an attribute value. This

condition can be used with *DeleteItem* , *PutItem* or *UpdateItem* operations; if the comparison evaluates to true, the operation succeeds; if not, the operation fails. You can use *ExpectedAttributeValue* in one of two different ways:

- Use *AttributeValueList* to specify one or more values to compare against an attribute. Use *ComparisonOperator* to specify how you want to perform the comparison. If the comparison evaluates to true, then the conditional operation succeeds.
- Use *Value* to specify a value that DynamoDB will compare against an attribute. If the values match, then *ExpectedAttributeValue* evaluates to true and the conditional operation succeeds. Optionally, you can also set *Exists* to false, indicating that you *do not* expect to find the attribute value in the table. In this case, the conditional operation succeeds only if the comparison evaluates to false.

Value and *Exists* are incompatible with *AttributeValueList* and *ComparisonOperator* . Note that if you use both sets of parameters at once, DynamoDB will return a *ValidationException* exception.

- **Value** (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –

- **M** (*dict*) –

A Map of attribute values.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute

can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **Exists** (*boolean*) –

Causes DynamoDB to evaluate the value before attempting a conditional operation:

- If *Exists* is `true`, DynamoDB will check to see if that attribute value already exists in the table. If it is found, then the operation succeeds. If it is not found, the operation fails with a *ConditionalCheckFailedException*.
- If *Exists* is `false`, DynamoDB assumes that the attribute value does not exist in the table. If in fact the value does not exist, then the assumption is valid and the operation succeeds. If the value is found, despite the assumption that it does not exist, the operation fails with a *ConditionalCheckFailedException*.

The default setting for *Exists* is `true`. If you supply a *Value* all by itself, DynamoDB assumes the attribute exists: You don't have to set *Exists* to `true`, because it is implied.

DynamoDB returns a *ValidationException* if:

- *Exists* is `true` but there is no *Value* to check. (You expect a value to exist, but don't specify what that value is.)
- *Exists* is `false` but you also provide a *Value*. (You cannot expect an attribute to have a value, while also expecting it not to exist.)
- **ComparisonOperator** (*string*) –

A comparator for evaluating attributes in the *AttributeValueList*. For example, equals, greater than, less than, etc.

The following comparison operators are available:

```
EQ | NE | LE | LT | GE | GT | NOT_NULL
| NULL | CONTAINS | NOT_CONTAINS |
BEGINS_WITH | IN | BETWEEN
```

The following are descriptions of each comparison operator.

- EQ : Equal. EQ is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] } .
- NE : Not equal. NE is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] } .
- LE : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- LT : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- GE : Greater than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- GT : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- NOT_NULL : The attribute exists. NOT_NULL is supported for all datatypes, including lists and maps. .. note::This operator tests for the existence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using NOT_NULL , the result is a Boolean *true* . This result is because the attribute “a ” exists; its data type is not relevant to the NOT_NULL comparison operator.
- NULL : The attribute does not exist. NULL is supported for all datatypes, including lists and maps. .. note::This operator tests for the nonexistence of an attribute, not its data type.

If the data type of attribute “a ” is null, and you evaluate it using `NULL`, the result is a Boolean *false*. This is because the attribute “a ” exists; its data type is not relevant to the `NULL` comparison operator.

- **CONTAINS** : Checks for a subsequence, or value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS”, “NS”, or “BS”), then the operator evaluates to true if it finds an exact match with any member of the set. **CONTAINS** is supported for lists: When evaluating “a CONTAINS b”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
- **NOT_CONTAINS** : Checks for absence of a subsequence, or absence of a value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS”, “NS”, or “BS”), then the operator evaluates to true if it *does not* find an exact match with any member of the set. **NOT_CONTAINS** is supported for lists: When evaluating “a NOT CONTAINS b”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
- **BEGINS_WITH** : Checks for a prefix. *AttributeValueList* can contain only one *AttributeValue* of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).
- **IN** : Checks for matching elements within two sets. *AttributeValueList* can contain one or more *AttributeValue* elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
- **BETWEEN** : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not compare to { "N" : "6" }. Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }.
- **AttributeValueList (list)** –

One or more values to evaluate against the supplied attribute.

The number of values in the list depends on the *Comparison-Operator* being used.

For type Number, value comparisons are numeric.

String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, `a` is greater than `A`, and `a` is greater than `B`. For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters.

For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.

For information on specifying data types in JSON, see [JSON Data Format](#) in the *Amazon DynamoDB Developer Guide*.

· **(dict)** –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

· **S (string)** –

A String data type.

· **N (string)** –

A Number data type.

· **B (bytes)** –

A Binary data type.

· **SS (list)** –

A String Set data type.

· **(string)** –

· **NS (list)** –

A Number Set data type.

· **(string)** –

· **BS (list)** –

A Binary Set data type.

· **(bytes)** –

· **M (dict)** –

A Map of attribute values.

· **(string)** –

· **(dict)** –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one

title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **ReturnValues** (*string*) – Use *ReturnValues* if you want to get the item attributes as they appeared before they were updated with the *PutItem* request. For *PutItem*, the valid values are:
 - **NONE** - If *ReturnValues* is not specified, or if its value is **NONE**, then nothing is returned. (This setting is the default for *ReturnValues*.)
 - **ALL_OLD** - If *PutItem* overwrote an attribute name-value pair, then the content of the old item is returned.

Note: Other “Valid Values” are not relevant to *PutItem*.

- **ReturnConsumedCapacity** (*string*) – Determines the level of detail about provisioned throughput consumption that is returned in the response:
 - **INDEXES** - The response includes the aggregate *ConsumedCapacity* for the operation, together with *ConsumedCapacity* for each table and secondary index that was accessed. Note that some operations, such as *GetItem* and *BatchGetItem*, do not access any indexes at all. In these cases, specifying **INDEXES** will only return *ConsumedCapacity* information for table(s).
 - **TOTAL** - The response includes only the aggregate *ConsumedCapacity* for the operation.
 - **NONE** - No *ConsumedCapacity* details are included in the response.
- **ReturnItemCollectionMetrics** (*string*) – Determines whether item collection metrics are returned. If set to **SIZE**, the response includes statistics about item collections, if any, that were modified during the operation are returned in the response. If set to **NONE** (the default), no statistics are returned.

Warning: This is a legacy parameter, for backward compatibility applications should use *ConditionExpression* instead. Do not combine parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception.

- **ConditionalOperator** (*string*) –

A logical operator to apply to the conditions in the *Expected* map:

- **AND** - If all of the conditions evaluate to true, then the entire map evaluates to true.
- **OR** - If at least one of the conditions evaluate to true, then the entire map evaluates to true.

If you omit *ConditionalOperator* , then AND is the default.

The operation will succeed only if the entire map evaluates to true.

Note: This parameter does not support attributes of type List or Map.

- **ConditionExpression** (*string*) – A condition that must be satisfied in order for a conditional *PutItem* operation to succeed.

An expression can contain any of the following:

- Functions: `attribute_exists` | `attribute_not_exists` | `attribute_type` | `contains` | `begins_with` | `size`
These function names are case-sensitive.
- Comparison operators: `=` | `|` | `|` | `=` | `=` | `BETWEEN` | `IN`
- Logical operators: `AND` | `OR` | `NOT`

For more information on condition expressions, see [Specifying Conditions](#) in the *Amazon DynamoDB Developer Guide* .

Note: *ConditionExpression* replaces the legacy *ConditionalOperator* and *Expected* parameters.

- **ExpressionAttributeNames** (*dict*) – One or more substitution tokens for attribute names in an expression. The following are some use cases for using *ExpressionAttributeNames* :
 - To access an attribute whose name conflicts with a DynamoDB reserved word.
 - To create a placeholder for repeating occurrences of an attribute name in an expression.
 - To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the `#` character in an expression to dereference an attribute name. For example, consider the following attribute name:

- `Percentile`

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see [Reserved Words](#) in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for *ExpressionAttributeNames* :

- `{"#P": "Percentile"}`

You could then use this substitution in an expression, as in this example:

- `#P = :val`

Note: Tokens that begin with the `:` character are *expression attribute values* , which are placeholders for the actual value at runtime.

For more information on expression attribute names, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide* .

- (*string*) –
 * (*string*) –

- **ExpressionAttributeValues** (*dict*) – One or more values that can be substituted in an expression.

Use the `:` (colon) character in an expression to dereference an attribute value. For example, suppose that you wanted to check whether the value of the *ProductStatus* attribute was one of the following:

Available | Backordered | Discontinued

You would first need to specify *ExpressionAttributeValues* as follows:

```
{ ":avail":{"S":"Available"}, ":back":{"S":"Backordered"},
  ":disc":{"S":"Discontinued"} }
```

You could then use these values in an expression, such as this:

```
ProductStatus IN (:avail, :back, :disc)
```

For more information on expression attribute values, see [Specifying Conditions in the Amazon DynamoDB Developer Guide](#).

- *(string)* -
- * *(dict)* -

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** *(string)* -
A String data type.
- **N** *(string)* -
A Number data type.
- **B** *(bytes)* -
A Binary data type.
- **SS** *(list)* -
A String Set data type.
- *(string)* -
- **NS** *(list)* -
A Number Set data type.
- *(string)* -
- **BS** *(list)* -
A Binary Set data type.
- *(bytes)* -
- **M** *(dict)* -
A Map of attribute values.
- *(string)* -
- *(dict)* -

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** *(list)* -
A List of attribute values.
- *(dict)* -

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

Return type dict

Returns

Response Syntax

```
{
  'Attributes': {
    'string': {
      'S': 'string',
      'N': 'string',
      'B': b'bytes',
      'SS': [
        'string',
      ],
      'NS': [
        'string',
      ],
      'BS': [
        b'bytes',
      ],
      'M': {
        'string': {'... recursive ...'}
      },
      'L': [
        {'... recursive ...'},
      ],
      'NULL': True|False,
      'BOOL': True|False
    }
  },
  'ConsumedCapacity': {
    'TableName': 'string',
    'CapacityUnits': 123.0,
    'Table': {
      'CapacityUnits': 123.0
    },
    'LocalSecondaryIndexes': {
      'string': {
        'CapacityUnits': 123.0
      }
    },
    'GlobalSecondaryIndexes': {
      'string': {
        'CapacityUnits': 123.0
      }
    }
  }
}
```



```

    }
  },
  'ItemCollectionMetrics': {
    'ItemCollectionKey': {
      'string': {
        'S': 'string',
        'N': 'string',
        'B': b'bytes',
        'SS': [
          'string',
        ],
        'NS': [
          'string',
        ],
        'BS': [
          b'bytes',
        ],
        'M': {
          'string': {'... recursive ...'}
        },
        'L': [
          {'... recursive ...'},
        ],
        'NULL': True|False,
        'BOOL': True|False
      }
    },
    'SizeEstimateRangeGB': [
      123.0,
    ]
  }
}

```

Response Structure

- (*dict*) –

Represents the output of a *PutItem* operation.

- **Attributes** (*dict*) –

The attribute values as they appeared before the *PutItem* operation, but only if *ReturnValues* is specified as `ALL_OLD` in the request. Each element consists of an attribute name and an attribute value.

- * (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –

- **M** (*dict*) –

A Map of attribute values.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **ConsumedCapacity** (*dict*) –

The capacity units consumed by an operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. *ConsumedCapacity* is only returned if the request asked for it. For more information, see [Provisioned Throughput](#) in the *Amazon DynamoDB Developer Guide* .

- * **TableName** (*string*) –

The name of the table that was affected by the operation.

- * **CapacityUnits** (*float*) –

The total number of capacity units consumed by the operation.

* **Table** (*dict*) –

The amount of throughput consumed on the table affected by the operation.

· **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

* **LocalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each local index affected by the operation.

· (*string*) –

· (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

· **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

* **GlobalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each global index affected by the operation.

· (*string*) –

· (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

· **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

– **ItemCollectionMetrics** (*dict*) –

Information about item collections, if any, that were affected by the operation. *ItemCollectionMetrics* is only returned if the request asked for it. If the table does not have any local secondary indexes, this information is not returned in the response.

Each *ItemCollectionMetrics* element consists of:

* **ItemCollectionKey** - The hash key value of the item collection. This is the same as the hash key of the item.

* **SizeEstimateRange** - An estimate of item collection size, in gigabytes. This value is a two-element array containing a lower bound and an upper bound for the estimate. The estimate includes the size of all the items in the table, plus the size of all attributes projected into all of the local secondary indexes on that table. Use this estimate to measure whether a local secondary index is approaching its size limit. The estimate is subject to change over time; therefore, do not rely on the precision or accuracy of the estimate.

* **ItemCollectionKey** (*dict*) –

The hash key value of the item collection. This value is the same as the hash key of the item.

· (*string*) –

· (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –

- **M** (*dict*) –

A Map of attribute values.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –


```
        'NS': [
            'string',
        ],
        'BS': [
            b'bytes',
        ],
        'M': {
            'string': {'... recursive ...'}
        },
        'L': [
            {'... recursive ...'},
        ],
        'NULL': True|False,
        'BOOL': True|False
    },
],
'ComparisonOperator': 'EQ'|'NE'|'IN'|'LE'|'LT'|'GE'|'GT'|'BETWEEN'|'NOT_NULL'|'...'
},
QueryFilter={
    'string': {
        'AttributeValueList': [
            {
                'S': 'string',
                'N': 'string',
                'B': b'bytes',
                'SS': [
                    'string',
                ],
                'NS': [
                    'string',
                ],
                'BS': [
                    b'bytes',
                ],
                'M': {
                    'string': {'... recursive ...'}
                },
                'L': [
                    {'... recursive ...'},
                ],
                'NULL': True|False,
                'BOOL': True|False
            },
        ],
        'ComparisonOperator': 'EQ'|'NE'|'IN'|'LE'|'LT'|'GE'|'GT'|'BETWEEN'|'NOT_NULL'|'...'
    },
},
ConditionalOperator='AND'|'OR',
ScanIndexForward=True|False,
ExclusiveStartKey={
    'string': {
        'S': 'string',
        'N': 'string',
        'B': b'bytes',
        'SS': [
            'string',
        ],
    },
}
```

```

        'NS': [
            'string',
        ],
        'BS': [
            b'bytes',
        ],
        'M': {
            'string': {'... recursive ...'}
        },
        'L': [
            {'... recursive ...'},
        ],
        'NULL': True|False,
        'BOOL': True|False
    }
},
ReturnConsumedCapacity='INDEXES'|'TOTAL'|'NONE',
ProjectionExpression='string',
FilterExpression='string',
KeyConditionExpression='string',
ExpressionAttributeNames={
    'string': 'string'
},
ExpressionAttributeValues={
    'string': {
        'S': 'string',
        'N': 'string',
        'B': b'bytes',
        'SS': [
            'string',
        ],
        'NS': [
            'string',
        ],
        'BS': [
            b'bytes',
        ],
        'M': {
            'string': {'... recursive ...'}
        },
        'L': [
            {'... recursive ...'},
        ],
        'NULL': True|False,
        'BOOL': True|False
    }
}
)

```

Parameters

- **TableName** (*string*) – [REQUIRED]

The name of the table containing the requested items.

- **IndexName** (*string*) – The name of an index to query. This index can be any local secondary index or global secondary index on the table. Note that if you use the *IndexName* parameter, you must also provide *TableName*.
- **Select** (*string*) – The attributes to be returned in the result. You can retrieve all item attributes, specific item attributes, the count of matching items, or in the

case of an index, some or all of the attributes projected into the index.

- `ALL_ATTRIBUTES` - Returns all of the item attributes from the specified table or index. If you query a local secondary index, then for each matching item in the index DynamoDB will fetch the entire item from the parent table. If the index is configured to project all item attributes, then all of the data can be obtained from the local secondary index, and no fetching is required.
- `ALL_PROJECTED_ATTRIBUTES` - Allowed only when querying an index. Retrieves all attributes that have been projected into the index. If the index is configured to project all attributes, this return value is equivalent to specifying `ALL_ATTRIBUTES`.
- `COUNT` - Returns the number of matching items, rather than the matching items themselves.
- `SPECIFIC_ATTRIBUTES` - Returns only the attributes listed in *AttributesToGet*. This return value is equivalent to specifying *AttributesToGet* without specifying any value for *Select*. If you query a local secondary index and request only attributes that are projected into that index, the operation will read only the index and not the table. If any of the requested attributes are not projected into the local secondary index, DynamoDB will fetch each of these attributes from the parent table. This extra fetching incurs additional throughput cost and latency. If you query a global secondary index, you can only request attributes that are projected into the index. Global secondary index queries cannot fetch attributes from the parent table.

If neither *Select* nor *AttributesToGet* are specified, DynamoDB defaults to `ALL_ATTRIBUTES` when accessing a table, and `ALL_PROJECTED_ATTRIBUTES` when accessing an index. You cannot use both *Select* and *AttributesToGet* together in a single request, unless the value for *Select* is `SPECIFIC_ATTRIBUTES`. (This usage is equivalent to specifying *AttributesToGet* without any value for *Select*.)

Note: If you use the *ProjectionExpression* parameter, then the value for *Select* can only be `SPECIFIC_ATTRIBUTES`. Any other value for *Select* will return an error.

• **AttributesToGet** (*list*) –

Warning: This is a legacy parameter, for backward compatibility. New applications should use *ProjectionExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception. This parameter allows you to retrieve attributes of type List or Map; however, it cannot retrieve individual elements within a List or a Map.

The names of one or more attributes to retrieve. If no attribute names are provided, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

Note that *AttributesToGet* has no effect on provisioned throughput consumption. DynamoDB determines capacity units consumed based on item size, not on the amount of data that is returned to an application.

You cannot use both *AttributesToGet* and *Select* together in a *Query* request, unless the value for *Select* is `SPECIFIC_ATTRIBUTES`. (This usage is equivalent to specifying *AttributesToGet* without any value for *Select*.)

If you query a local secondary index and request only attributes that are pro-

jected into that index, the operation will read only the index and not the table. If any of the requested attributes are not projected into the local secondary index, DynamoDB will fetch each of these attributes from the parent table. This extra fetching incurs additional throughput cost and latency.

If you query a global secondary index, you can only request attributes that are projected into the index. Global secondary index queries cannot fetch attributes from the parent table.

– (*string*) –

- **Limit** (*integer*) – The maximum number of items to evaluate (not necessarily the number of matching items). If DynamoDB processes the number of items up to the limit while processing the results, it stops the operation and returns the matching values up to that point, and a key in *LastEvaluatedKey* to apply in a subsequent operation, so that you can pick up where you left off. Also, if the processed data set size exceeds 1 MB before DynamoDB reaches this limit, it stops the operation and returns the matching values up to the limit, and a key in *LastEvaluatedKey* to apply in a subsequent operation to continue the operation. For more information, see [Query and Scan](#) in the *Amazon DynamoDB Developer Guide*.
- **ConsistentRead** (*boolean*) – Determines the read consistency model: If set to `true`, then the operation uses strongly consistent reads; otherwise, the operation uses eventually consistent reads.

Strongly consistent reads are not supported on global secondary indexes. If you query a global secondary index with *ConsistentRead* set to `true`, you will receive a *ValidationException*.

Warning: This is a legacy parameter, for backward compatibility. New applications should use *KeyConditionExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception.

- **KeyConditions** (*dict*) –

The selection criteria for the query. For a query on a table, you can have conditions only on the table primary key attributes. You must provide the hash key attribute name and value as an EQ condition. You can optionally provide a second condition, referring to the range key attribute.

Note: If you don't provide a range key condition, all of the items that match the hash key will be retrieved. If a *FilterExpression* or *QueryFilter* is present, it will be applied after the items are retrieved.

For a query on an index, you can have conditions only on the index key attributes. You must provide the index hash attribute name and value as an EQ condition. You can optionally provide a second condition, referring to the index key range attribute.

Each *KeyConditions* element consists of an attribute name to compare, along with the following:

- *AttributeValueList* - One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *ComparisonOperator* being used. For type Number, value comparisons are numeric. String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, `a` is greater than `A`, and `a` is greater than `B`. For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters. For Bi-

nary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.

- *ComparisonOperator* - A comparator for evaluating attributes, for example, equals, greater than, less than, and so on. For *KeyConditions*, only the following comparison operators are supported: EQ | LE | LT | GE | GT | BEGINS_WITH | BETWEEN. The following are descriptions of these comparison operators. * EQ : Equal. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one specified in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" }. Also, { "N" : "6" } does not equal { "NS" : ["6", "2", "1"] }.
- LE : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" }. Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }.
- LT : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" }. Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }.
- GE : Greater than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" }. Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }.
- GT : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" }. Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }.
- BEGINS_WITH : Checks for a prefix. *AttributeValueList* can contain only one *AttributeValue* of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).
- BETWEEN : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not compare to { "N" : "6" }. Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }

For usage examples of *AttributeValueList* and *ComparisonOperator*, see [Legacy Conditional Parameters](#) in the *Amazon DynamoDB Developer Guide*.

- (string) –
- * (dict) –

Represents the selection criteria for a *Query* or *Scan* operation:

- For a *Query* operation, *Condition* is used for specifying the *KeyConditions* to use when querying a table or an index. For *KeyConditions*, only the following comparison operators are supported: EQ | LE | LT | GE | GT | BEGINS_WITH | BETWEEN. *Condition* is also used in a *QueryFilter*, which evaluates the query results and returns only the desired values.
- For a *Scan* operation, *Condition* is used in a *ScanFilter*, which evaluates the scan results and returns only the desired values.

- **AttributeValueList** (*list*) –

One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *ComparisonOperator* being used.

For type Number, value comparisons are numeric.

String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A, and a is greater than B. For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters

·

For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –

- **M** (*dict*) –

A Map of attribute values.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **ComparisonOperator** (*string*) – **[REQUIRED]**

A comparator for evaluating attributes. For example, equals, greater than, less than, etc.

The following comparison operators are available:

```
EQ | NE | LE | LT | GE | GT | NOT_NULL  
| NULL | CONTAINS | NOT_CONTAINS |  
BEGINS_WITH | IN | BETWEEN
```

The following are descriptions of each comparison operator.

- **EQ** : Equal. EQ is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] } .
- **NE** : Not equal. NE is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does

not equal `{"N": "6"}` . Also, `{"N": "6"}` does not equal `{"NS": ["6", "2", "1"]}` .

- **LE** : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}` . Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}` .
- **LT** : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}` . Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}` .
- **GE** : Greater than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}` . Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}` .
- **GT** : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}` . Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}` .
- **NOT_NULL** : The attribute exists. **NOT_NULL** is supported for all datatypes, including lists and maps. .. note::This operator tests for the existence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using **NOT_NULL** , the result is a Boolean *true* . This result is because the attribute “a ” exists; its data type is not relevant to the **NOT_NULL** comparison operator.
- **NULL** : The attribute does not exist. **NULL** is supported for all datatypes, including lists and maps. .. note::This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using **NULL** , the result is a Boolean *false* . This is because the attribute “a ” exists; its data type is not relevant to the **NULL** comparison operator.
- **CONTAINS** : Checks for a subsequence, or value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS ”, “NS ”, or “BS ”), then the operator evaluates to true if it finds an exact match with any member of the set. **CONTAINS** is supported

for lists: When evaluating “a CONTAINS b”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.

- **NOT_CONTAINS** : Checks for absence of a subsequence, or absence of a value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS ”, “NS ”, or “BS ”), then the operator evaluates to true if it *does not* find an exact match with any member of the set. NOT_CONTAINS is supported for lists: When evaluating “a NOT CONTAINS b”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
- **BEGINS_WITH** : Checks for a prefix. *AttributeValueList* can contain only one *AttributeValue* of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).
- **IN** : Checks for matching elements within two sets. *AttributeValueList* can contain one or more *AttributeValue* elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
- **BETWEEN** : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not compare to { "N" : "6" } . Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }

For usage examples of *AttributeValueList* and *ComparisonOperator* , see [Legacy Conditional Parameters](#) in the *Amazon DynamoDB Developer Guide* .

- **QueryFilter** (*dict*) –

Warning: This is a legacy parameter, for backward compatibility. New applications should use *FilterExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception.

A condition that evaluates the query results after the items are read and returns only the desired values.

This parameter does not support attributes of type List or Map.

Note: A *QueryFilter* is applied after the items have already been read; the process of filtering does not consume any additional read capacity units.

If you provide more than one condition in the *QueryFilter* map, then by default all of the conditions must evaluate to true. In other words, the conditions are ANDed together. (You can use the *ConditionalOperator* parameter to OR the conditions instead. If you do this, then at least one of the conditions must evaluate to true, rather than all of them.)

Note that *QueryFilter* does not allow key attributes. You cannot define a filter condition on a hash key or range key.

Each *QueryFilter* element consists of an attribute name to compare, along with the following:

- *AttributeValueList* - One or more values to evaluate against the supplied attribute. The number of values in the list depends on the operator specified in *ComparisonOperator* . For type Number, value comparisons are numeric. String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A , and a is greater than B . For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters . For type Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values. For information on specifying data types in JSON, see [JSON Data Format](#) in the *Amazon DynamoDB Developer Guide* .
- *ComparisonOperator* - A comparator for evaluating attributes. For example, equals, greater than, less than, etc. The following comparison operators are available: EQ | NE | LE | LT | GE | GT | NOT_NULL | NULL | CONTAINS | NOT_CONTAINS | BEGINS_WITH | IN | BETWEEN For complete descriptions of all comparison operators, see the [Condition](#) data type.
- (string) –
- * (dict) –

Represents the selection criteria for a *Query* or *Scan* operation:

- For a *Query* operation, *Condition* is used for specifying the *KeyConditions* to use when querying a table or an index. For *KeyConditions* , only the following comparison operators are supported: EQ | LE | LT | GE | GT | BEGINS_WITH | BETWEEN *Condition* is also used in a *QueryFilter* , which evaluates the query results and returns only the desired values.
- For a *Scan* operation, *Condition* is used in a *ScanFilter* , which evaluates the scan results and returns only the desired values.
- **AttributeValueList** (*list*) –

One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *ComparisonOperator* being used.

For type Number, value comparisons are numeric.

String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A , and a is greater than B . For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters .

For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.

- **(dict)** –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S (string)** –

A String data type.

- **N (string)** –

A Number data type.

- **B (bytes)** –

A Binary data type.

- **SS (list)** –

A String Set data type.

- **(string)** –

- **NS (list)** –

A Number Set data type.

- **(string)** –

- **BS (list)** –

A Binary Set data type.

- **(bytes)** –

- **M (dict)** –

A Map of attribute values.

- **(string)** –

- **(dict)** –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L (list)** –

A List of attribute values.

- **(dict)** –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **ComparisonOperator** (*string*) – **[REQUIRED]**

A comparator for evaluating attributes. For example, equals, greater than, less than, etc.

The following comparison operators are available:

```
EQ | NE | LE | LT | GE | GT | NOT_NULL
| NULL | CONTAINS | NOT_CONTAINS |
BEGINS_WITH | IN | BETWEEN
```

The following are descriptions of each comparison operator.

- **EQ** : Equal. EQ is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] } .
- **NE** : Not equal. NE is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] } .
- **LE** : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- **LT** : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- **GE** : Greater than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- **GT** : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not

a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .

- NOT_NULL : The attribute exists. NOT_NULL is supported for all datatypes, including lists and maps. .. note::This operator tests for the existence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using NOT_NULL , the result is a Boolean *true* . This result is because the attribute “a ” exists; its data type is not relevant to the NOT_NULL comparison operator.
- NULL : The attribute does not exist. NULL is supported for all datatypes, including lists and maps. .. note::This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using NULL , the result is a Boolean *false* . This is because the attribute “a ” exists; its data type is not relevant to the NULL comparison operator.
- CONTAINS : Checks for a subsequence, or value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS ”, “NS ”, or “BS ”), then the operator evaluates to true if it finds an exact match with any member of the set. CONTAINS is supported for lists: When evaluating “a CONTAINS b ”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
- NOT_CONTAINS : Checks for absence of a subsequence, or absence of a value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS ”, “NS ”, or “BS ”), then the operator evaluates to true if it *does not* find an exact match with any member of the set. NOT_CONTAINS is supported for lists: When evaluating “a NOT CONTAINS b ”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
- BEGINS_WITH : Checks for a prefix. *AttributeValueList* can contain only one *AttributeValue* of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).
- IN : Checks for matching elements within two sets. *AttributeValueList* can contain one or more *AttributeValue* elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the

item attribute, the expression evaluates to true.

- **BETWEEN** : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not compare to { "N" : "6" } . Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }

For usage examples of *AttributeValueList* and *ComparisonOperator* , see [Legacy Conditional Parameters](#) in the *Amazon DynamoDB Developer Guide* .

Warning: This is a legacy parameter, for backward compatibility applications should use *FilterExpression* instead. Do not combine parameters and expression parameters in a single API call; otherwise DynamoDB will return a *ValidationException* exception.

- **ConditionalOperator** (*string*) –

A logical operator to apply to the conditions in a *QueryFilter* map:

- AND - If all of the conditions evaluate to true, then the entire map evaluates to true.
- OR - If at least one of the conditions evaluate to true, then the entire map evaluates to true.

If you omit *ConditionalOperator* , then AND is the default.

The operation will succeed only if the entire map evaluates to true.

Note: This parameter does not support attributes of type List or Map.

- **ScanIndexForward** (*boolean*) – Specifies the order in which to return the query results - either ascending (*true*) or descending (*false*).

Items with the same hash key are stored in sorted order by range key .If the range key data type is Number, the results are stored in numeric order. For type String, the results are returned in order of ASCII character code values. For type Binary, DynamoDB treats each byte of the binary data as unsigned.

If *ScanIndexForward* is *true* , DynamoDB returns the results in order, by range key. This is the default behavior.

If *ScanIndexForward* is *false* , DynamoDB sorts the results in descending order by range key, and then returns the results to the client.

- **ExclusiveStartKey** (*dict*) – The primary key of the first item that this operation will evaluate. Use the value that was returned for *LastEvaluatedKey* in the previous operation.

The data type for *ExclusiveStartKey* must be String, Number or Binary. No set data types are allowed.

- (*string*) –
- * (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –

- **M** (*dict*) –

A Map of attribute values.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **ReturnConsumedCapacity** (*string*) – Determines the level of detail about provisioned throughput consumption that is returned in the response:

- *INDEXES* - The response includes the aggregate *ConsumedCapacity* for the operation, together with *ConsumedCapacity* for each table and secondary index that was accessed. Note that some operations, such as *GetItem* and *BatchGetItem*, do not access any indexes at all. In these cases, specifying *INDEXES* will only return *ConsumedCapacity* information for table(s).
- *TOTAL* - The response includes only the aggregate *ConsumedCapacity* for the operation.
- *NONE* - No *ConsumedCapacity* details are included in the response.
- **ProjectionExpression** (*string*) – A string that identifies one or more attributes to retrieve from the table. These attributes can include scalars, sets, or elements of a JSON document. The attributes in the expression must be separated by commas.

If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

For more information, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

Note: *ProjectionExpression* replaces the legacy *AttributesToGet* parameter.

- **FilterExpression** (*string*) – A string that contains conditions that DynamoDB applies after the *Query* operation, but before the data is returned to you. Items that do not satisfy the *FilterExpression* criteria are not returned.

Note: A *FilterExpression* is applied after the items have already been read; the process of filtering does not consume any additional read capacity units.

For more information, see [Filter Expressions](#) in the *Amazon DynamoDB Developer Guide*.

Note: *FilterExpression* replaces the legacy *QueryFilter* and *ConditionalOperator* parameters.

- **KeyConditionExpression** (*string*) – The condition that specifies the key value(s) for items to be retrieved by the *Query* action.

The condition must perform an equality test on a single hash key value. The condition can also perform one of several comparison tests on a single range key value. *Query* can use *KeyConditionExpression* to retrieve one item with a given hash and range key value, or several items that have the same hash key value but different range key values.

The hash key equality test is required, and must be specified in the following format:

```
hashAttributeName = :hashval
```

If you also want to provide a range key condition, it must be combined using *AND* with the hash key condition. Following is an example, using the *=* comparison operator for the range key:

```
hashAttributeName      =      :hashval      AND
rangeAttributeName = :rangeval
```

Valid comparisons for the range key condition are as follows:

- `rangeAttributeName = :rangeval` - true if the range key is equal to `:rangeval`.
- `rangeAttributeName ** :rangeval` - true if the range key is less than `:rangeval`.

- `rangeAttributeName = :rangeval` - true if the range key is less than or equal to `:rangeval`.
- `rangeAttributeName ** :rangeval` - true if the range key is greater than `:rangeval`.
- `rangeAttributeName = :rangeval` - true if the range key is greater than or equal to `:rangeval`.
- `rangeAttributeName BETWEEN :rangeval1 AND :rangeval2` - true if the range key is greater than or equal to `:rangeval1`, and less than or equal to `:rangeval2`.
- `begins_with ( rangeAttributeName , :rangeval )` - true if the range key begins with a particular operand. (You cannot use this function with a range key that is of type Number.) Note that the function name `begins_with` is case-sensitive.

Use the *ExpressionAttributeValues* parameter to replace tokens such as `:hashval` and `:rangeval` with actual values at runtime.

You can optionally use the *ExpressionAttributeNames* parameter to replace the names of the hash and range attributes with placeholder tokens. This option might be necessary if an attribute name conflicts with a DynamoDB reserved word. For example, the following *KeyConditionExpression* parameter causes an error because *Size* is a reserved word:

- `Size = :myval`

To work around this, define a placeholder (such as `#S`) to represent the attribute name *Size*. *KeyConditionExpression* then is as follows:

- `#S = :myval`

For a list of reserved words, see [Reserved Words](#) in the *Amazon DynamoDB Developer Guide*.

For more information on *ExpressionAttributeNames* and *ExpressionAttributeValues*, see [Using Placeholders for Attribute Names and Values](#) in the *Amazon DynamoDB Developer Guide*.

Note: *KeyConditionExpression* replaces the legacy *KeyConditions* parameter.

- **ExpressionAttributeNames** (*dict*) – One or more substitution tokens for attribute names in an expression. The following are some use cases for using *ExpressionAttributeNames*:
 - To access an attribute whose name conflicts with a DynamoDB reserved word.
 - To create a placeholder for repeating occurrences of an attribute name in an expression.
 - To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the `#` character in an expression to dereference an attribute name. For example, consider the following attribute name:

- `Percentile`

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see [Reserved Words](#) in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for *ExpressionAttributeNames*:

- `{"#P": "Percentile"}`

You could then use this substitution in an expression, as in this example:

- `#P = :val`

Note: Tokens that begin with the `:` character are *expression attribute values*, which are placeholders for the actual value at runtime.

For more information on expression attribute names, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

- *(string)* –
- * *(string)* –

- **ExpressionAttributeValues** (*dict*) – One or more values that can be substituted in an expression.

Use the **:** (colon) character in an expression to dereference an attribute value. For example, suppose that you wanted to check whether the value of the *ProductStatus* attribute was one of the following:

```
Available | Backordered | Discontinued
```

You would first need to specify *ExpressionAttributeValues* as follows:

```
{ ":avail":{"S":"Available"}, ":back":{"S":"Backordered"},
  ":disc":{"S":"Discontinued"} }
```

You could then use these values in an expression, such as this:

```
ProductStatus IN (:avail, :back, :disc)
```

For more information on expression attribute values, see [Specifying Conditions](#) in the *Amazon DynamoDB Developer Guide*.

- *(string)* –
- * *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –
A String data type.
- **N** (*string*) –
A Number data type.
- **B** (*bytes*) –
A Binary data type.
- **SS** (*list*) –
A String Set data type.
- (*string*) –
- **NS** (*list*) –
A Number Set data type.
- (*string*) –
- **BS** (*list*) –
A Binary Set data type.
- (*bytes*) –
- **M** (*dict*) –
A Map of attribute values.
- (*string*) –

- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

Return type dict

Returns

Response Syntax

```
{
  'Items': [
    {
      'string': {
        'S': 'string',
        'N': 'string',
        'B': b'bytes',
        'SS': [
          'string',
        ],
        'NS': [
          'string',
        ],
        'BS': [
          b'bytes',
        ],
        'M': {
          'string': {'... recursive ...'}
        },
        'L': [
          {'... recursive ...'},
        ],
        'NULL': True|False,
        'BOOL': True|False
      }
    },
  ],
}
```



```

],
'Count': 123,
'ScannedCount': 123,
'LastEvaluatedKey': {
    'string': {
        'S': 'string',
        'N': 'string',
        'B': b'bytes',
        'SS': [
            'string',
        ],
        'NS': [
            'string',
        ],
        'BS': [
            b'bytes',
        ],
        'M': {
            'string': {'... recursive ...'}
        },
        'L': [
            {'... recursive ...'},
        ],
        'NULL': True|False,
        'BOOL': True|False
    }
},
'ConsumedCapacity': {
    'TableName': 'string',
    'CapacityUnits': 123.0,
    'Table': {
        'CapacityUnits': 123.0
    },
    'LocalSecondaryIndexes': {
        'string': {
            'CapacityUnits': 123.0
        }
    },
    'GlobalSecondaryIndexes': {
        'string': {
            'CapacityUnits': 123.0
        }
    }
}
}
}

```

Response Structure

- (*dict*) –

Represents the output of a *Query* operation.

- **Items** (*list*) –

An array of item attributes that match the query criteria. Each element in this array consists of an attribute name and the value for that attribute.

- * (*dict*) –

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only

one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –

- **M** (*dict*) –

A Map of attribute values.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

– **Count** (*integer*) –

The number of items in the response.

If you used a *QueryFilter* in the request, then *Count* is the number of items returned after the filter was applied, and *ScannedCount* is the number of matching items before the filter was applied.

If you did not use a filter in the request, then *Count* and *ScannedCount* are the same.

– **ScannedCount** (*integer*) –

The number of items evaluated, before any *QueryFilter* is applied. A high *ScannedCount* value with few, or no, *Count* results indicates an inefficient *Query* operation. For more information, see [Count and ScannedCount](#) in the *Amazon DynamoDB Developer Guide*.

If you did not use a filter in the request, then *ScannedCount* is the same as *Count*.

– **LastEvaluatedKey** (*dict*) –

The primary key of the item where the operation stopped, inclusive of the previous result set. Use this value to start a new operation, excluding this value in the new request.

If *LastEvaluatedKey* is empty, then the “last page” of results has been processed and there is no more data to be retrieved.

If *LastEvaluatedKey* is not empty, it does not necessarily mean that there is more data in the result set. The only way to know when you have reached the end of the result set is when *LastEvaluatedKey* is empty.

* (*string*) –

· (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

· **S** (*string*) –

A String data type.

· **N** (*string*) –

A Number data type.

· **B** (*bytes*) –

A Binary data type.

· **SS** (*list*) –

A String Set data type.

· (*string*) –

· **NS** (*list*) –

A Number Set data type.

· (*string*) –

· **BS** (*list*) –

A Binary Set data type.

- *(bytes)* –
- **M** (*dict*) –

A Map of attribute values.

- *(string)* –
- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

– **ConsumedCapacity** (*dict*) –

The capacity units consumed by an operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. *ConsumedCapacity* is only returned if the request asked for it. For more information, see [Provisioned Throughput](#) in the *Amazon DynamoDB Developer Guide* .

- * **TableName** (*string*) –

The name of the table that was affected by the operation.

- * **CapacityUnits** (*float*) –

The total number of capacity units consumed by the operation.

- * **Table** (*dict*) –

The amount of throughput consumed on the table affected by the operation.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

- * **LocalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each local index affected by the operation.

- *(string)* –

- *(dict)* –
Represents the amount of provisioned throughput capacity consumed on a table or an index.
 - **CapacityUnits** (*float*) –
The total number of capacity units consumed on a table or an index.
- * **GlobalSecondaryIndexes** (*dict*) –
- The amount of throughput consumed on each global index affected by the operation.
- *(string)* –
 - *(dict)* –
Represents the amount of provisioned throughput capacity consumed on a table or an index.
 - **CapacityUnits** (*float*) –
The total number of capacity units consumed on a table or an index.

scan (***kwargs*)

The *Scan* operation returns one or more items and item attributes by accessing every item in a table or a secondary index. To have DynamoDB return fewer items, you can provide a *ScanFilter* operation.

If the total number of scanned items exceeds the maximum data set size limit of 1 MB, the scan stops and results are returned to the user as a *LastEvaluatedKey* value to continue the scan in a subsequent operation. The results also include the number of items exceeding the limit. A scan can result in no table data meeting the filter criteria.

By default, *Scan* operations proceed sequentially; however, for faster performance on a large table or secondary index, applications can request a parallel *Scan* operation by providing the *Segment* and *TotalSegments* parameters. For more information, see [Parallel Scan](#) in the *Amazon DynamoDB Developer Guide*.

By default, *Scan* uses eventually consistent reads when accessing the data in the table or local secondary index. However, you can use strongly consistent reads instead by setting the *ConsistentRead* parameter to *true*.

Request Syntax

```
response = client.scan(
    TableName='string',
    IndexName='string',
    AttributesToGet=[
        'string',
    ],
    Limit=123,
    Select='ALL_ATTRIBUTES' | 'ALL_PROJECTED_ATTRIBUTES' | 'SPECIFIC_ATTRIBUTES' | 'COUNT',
    ScanFilter={
        'string': {
            'AttributeValueList': [
                {
                    'S': 'string',
                    'N': 'string',
                    'B': b'bytes',
                    'SS': [
                        'string',
                    ],
                }
            ]
        }
    }
)
```

```
        'NS': [
            'string',
        ],
        'BS': [
            b'bytes',
        ],
        'M': {
            'string': {'... recursive ...'}
        },
        'L': [
            {'... recursive ...'},
        ],
        'NULL': True|False,
        'BOOL': True|False
    },
],
'ComparisonOperator': 'EQ'|'NE'|'IN'|'LE'|'LT'|'GE'|'GT'|'BETWEEN'|'NOT_NULL'|'...'
},
ConditionalOperator='AND'|'OR',
ExclusiveStartKey={
    'string': {
        'S': 'string',
        'N': 'string',
        'B': b'bytes',
        'SS': [
            'string',
        ],
        'NS': [
            'string',
        ],
        'BS': [
            b'bytes',
        ],
        'M': {
            'string': {'... recursive ...'}
        },
        'L': [
            {'... recursive ...'},
        ],
        'NULL': True|False,
        'BOOL': True|False
    }
},
ReturnConsumedCapacity='INDEXES'|'TOTAL'|'NONE',
TotalSegments=123,
Segment=123,
ProjectionExpression='string',
FilterExpression='string',
ExpressionAttributeNames={
    'string': 'string'
},
ExpressionAttributeValues={
    'string': {
        'S': 'string',
        'N': 'string',
        'B': b'bytes',
        'SS': [
```

```

        'string',
    ],
    'NS': [
        'string',
    ],
    'BS': [
        b'bytes',
    ],
    'M': {
        'string': {'... recursive ...'}
    },
    'L': [
        {'... recursive ...'},
    ],
    'NULL': True|False,
    'BOOL': True|False
    }
},
ConsistentRead=True|False
)

```

Parameters

- **TableName** (*string*) – [REQUIRED]

The name of the table containing the requested items; or, if you provide `IndexName`, the name of the table to which that index belongs.

- **IndexName** (*string*) – The name of a secondary index to scan. This index can be any local secondary index or global secondary index. Note that if you use the `IndexName` parameter, you must also provide `TableName`.

Warning: This is a legacy parameter, for backward compatibility. New applications should use *ProjectionExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception.

- **AttributesToGet** (*list*) –

This parameter allows you to retrieve attributes of type List or Map; however, it cannot retrieve individual elements within a List or a Map.

The names of one or more attributes to retrieve. If no attribute names are provided, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

Note that *AttributesToGet* has no effect on provisioned throughput consumption. DynamoDB determines capacity units consumed based on item size, not on the amount of data that is returned to an application.

– (*string*) –

- **Limit** (*integer*) – The maximum number of items to evaluate (not necessarily the number of matching items). If DynamoDB processes the number of items up to the limit while processing the results, it stops the operation and returns the matching values up to that point, and a key in *LastEvaluatedKey* to apply in a subsequent operation, so that you can pick up where you left off. Also, if the processed data set size exceeds 1 MB before DynamoDB reaches this limit, it stops the operation and returns the matching values up to the limit, and a key in *LastEvaluatedKey* to apply in a subsequent operation to continue the operation. For more information, see [Query and Scan](#) in the *Amazon DynamoDB Developer Guide*.
- **Select** (*string*) – The attributes to be returned in the result. You can retrieve all item attributes, specific item attributes, or the count of matching items.

- ALL_ATTRIBUTES - Returns all of the item attributes.
- COUNT - Returns the number of matching items, rather than the matching items themselves.
- SPECIFIC_ATTRIBUTES - Returns only the attributes listed in *AttributesToGet* . This return value is equivalent to specifying *AttributesToGet* without specifying any value for *Select* .

If neither *Select* nor *AttributesToGet* are specified, DynamoDB defaults to ALL_ATTRIBUTES . You cannot use both *AttributesToGet* and *Select* together in a single request, unless the value for *Select* is SPECIFIC_ATTRIBUTES . (This usage is equivalent to specifying *AttributesToGet* without any value for *Select* .)

• **ScanFilter** (*dict*) –

Warning: This is a legacy parameter, for backward compatibility. New applications should use *FilterExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception.

A condition that evaluates the scan results and returns only the desired values.

Note: This parameter does not support attributes of type List or Map.

If you specify more than one condition in the *ScanFilter* map, then by default all of the conditions must evaluate to true. In other words, the conditions are ANDed together. (You can use the *ConditionalOperator* parameter to OR the conditions instead. If you do this, then at least one of the conditions must evaluate to true, rather than all of them.)

Each *ScanFilter* element consists of an attribute name to compare, along with the following:

- *AttributeValueList* - One or more values to evaluate against the supplied attribute. The number of values in the list depends on the operator specified in *ComparisonOperator* . For type Number, value comparisons are numeric. String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A , and a is greater than B . For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters . For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values. For information on specifying data types in JSON, see [JSON Data Format](#) in the *Amazon DynamoDB Developer Guide* .
- *ComparisonOperator* - A comparator for evaluating attributes. For example, equals, greater than, less than, etc. The following comparison operators are available: EQ | NE | LE | LT | GE | GT | NOT_NULL | NULL | CONTAINS | NOT_CONTAINS | BEGINS_WITH | IN | BETWEEN For complete descriptions of all comparison operators, see [Condition](#) .
- (*string*) –
 - * (*dict*) –

Represents the selection criteria for a *Query* or *Scan* operation:

- For a *Query* operation, *Condition* is used for specifying the *KeyConditions* to use when querying a table or an index. For *KeyConditions* , only the following comparison operators are supported: EQ | LE | LT | GE | GT | BEGINS_WITH | BETWEEN *Condition* is also used in a

QueryFilter , which evaluates the query results and returns only the desired values.

- For a *Scan* operation, *Condition* is used in a *ScanFilter* , which evaluates the scan results and returns only the desired values.

- **AttributeValueList** (*list*) –

One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *Comparison-Operator* being used.

For type Number, value comparisons are numeric.

String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A , and a is greater than B . For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters

·

For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –

- **M** (*dict*) –

A Map of attribute values.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **ComparisonOperator** (*string*) – **[REQUIRED]**

A comparator for evaluating attributes. For example, equals, greater than, less than, etc.

The following comparison operators are available:

```
EQ | NE | LE | LT | GE | GT | NOT_NULL
| NULL | CONTAINS | NOT_CONTAINS |
BEGINS_WITH | IN | BETWEEN
```

The following are descriptions of each comparison operator.

- **EQ** : Equal. EQ is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] } .
- **NE** : Not equal. NE is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] } .
- **LE** : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the

value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.

- **LT** : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
- **GE** : Greater than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
- **GT** : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
- **NOT_NULL** : The attribute exists. **NOT_NULL** is supported for all datatypes, including lists and maps. .. note::This operator tests for the existence of an attribute, not its data type. If the data type of attribute “a” is null, and you evaluate it using **NOT_NULL**, the result is a Boolean *true*. This result is because the attribute “a” exists; its data type is not relevant to the **NOT_NULL** comparison operator.
- **NULL** : The attribute does not exist. **NULL** is supported for all datatypes, including lists and maps. .. note::This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute “a” is null, and you evaluate it using **NULL**, the result is a Boolean *false*. This is because the attribute “a” exists; its data type is not relevant to the **NULL** comparison operator.
- **CONTAINS** : Checks for a subsequence, or value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS”, “NS”, or “BS”), then the operator evaluates to true if it finds an exact match with any member of the set. **CONTAINS** is supported for lists: When evaluating “a **CONTAINS** b”, “a” can be a list; however, “b” cannot be a set, a map, or a list.
- **NOT_CONTAINS** : Checks for absence of a subsequence, or absence of a value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the compar-

ison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set ("SS ", "NS ", or "BS "), then the operator evaluates to true if it *does not* find an exact match with any member of the set. NOT_CONTAINS is supported for lists: When evaluating "a NOT CONTAINS b", "a " can be a list; however, "b " cannot be a set, a map, or a list.

- **BEGINS_WITH** : Checks for a prefix. *AttributeValueList* can contain only one *AttributeValue* of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).
- **IN** : Checks for matching elements within two sets. *AttributeValueList* can contain one or more *AttributeValue* elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
- **BETWEEN** : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not compare to { "N" : "6" } . Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }

For usage examples of *AttributeValueList* and *ComparisonOperator* , see [Legacy Conditional Parameters](#) in the *Amazon DynamoDB Developer Guide* .

- **ConditionalOperator** (*string*) –

Warning: This is a legacy parameter, for backward compatibility applications should use *FilterExpression* instead. Do not combine parameters and expression parameters in a single API call; otherwise DynamoDB will return a *ValidationException* exception.

A logical operator to apply to the conditions in a *ScanFilter* map:

- **AND** - If all of the conditions evaluate to true, then the entire map evaluates to true.
- **OR** - If at least one of the conditions evaluate to true, then the entire map evaluates to true.

If you omit *ConditionalOperator* , then AND is the default.

The operation will succeed only if the entire map evaluates to true.

Note: This parameter does not support attributes of type List or Map.

- **ExclusiveStartKey** (*dict*) – The primary key of the first item that this operation will evaluate. Use the value that was returned for *LastEvaluatedKey* in the previous operation.

The data type for *ExclusiveStartKey* must be String, Number or Binary. No set data types are allowed.

In a parallel scan, a *Scan* request that includes *ExclusiveStartKey* must specify the same segment whose previous *Scan* returned the corresponding value of *LastEvaluatedKey*.

- (*string*) –
- * (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –

- **M** (*dict*) –

A Map of attribute values.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **ReturnConsumedCapacity** (*string*) – Determines the level of detail about provisioned throughput consumption that is returned in the response:
 - **INDEXES** - The response includes the aggregate *ConsumedCapacity* for the operation, together with *ConsumedCapacity* for each table and secondary index that was accessed. Note that some operations, such as *GetItem* and *BatchGetItem*, do not access any indexes at all. In these cases, specifying **INDEXES** will only return *ConsumedCapacity* information for table(s).
 - **TOTAL** - The response includes only the aggregate *ConsumedCapacity* for the operation.
 - **NONE** - No *ConsumedCapacity* details are included in the response.
- **TotalSegments** (*integer*) – For a parallel *Scan* request, *TotalSegments* represents the total number of segments into which the *Scan* operation will be divided. The value of *TotalSegments* corresponds to the number of application workers that will perform the parallel scan. For example, if you want to use four application threads to scan a table or an index, specify a *TotalSegments* value of 4.

The value for *TotalSegments* must be greater than or equal to 1, and less than or equal to 1000000. If you specify a *TotalSegments* value of 1, the *Scan* operation will be sequential rather than parallel.

If you specify *TotalSegments*, you must also specify *Segment*.

- **Segment** (*integer*) – For a parallel *Scan* request, *Segment* identifies an individual segment to be scanned by an application worker.

Segment IDs are zero-based, so the first segment is always 0. For example, if you want to use four application threads to scan a table or an index, then the first thread specifies a *Segment* value of 0, the second thread specifies 1, and so on.

The value of *LastEvaluatedKey* returned from a parallel *Scan* request must be used as *ExclusiveStartKey* with the same segment ID in a subsequent *Scan* operation.

The value for *Segment* must be greater than or equal to 0, and less than the value provided for *TotalSegments*.

If you provide *Segment*, you must also provide *TotalSegments*.

- **ProjectionExpression** (*string*) – A string that identifies one or more attributes to retrieve from the specified table or index. These attributes can include scalars, sets, or elements of a JSON document. The attributes in the expression must be separated by commas.

If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

For more information, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

Note: *ProjectionExpression* replaces the legacy *AttributesToGet* parameter.

- **FilterExpression** (*string*) – A string that contains conditions that DynamoDB applies after the *Scan* operation, but before the data is returned to you. Items that do not satisfy the *FilterExpression* criteria are not returned.
-

Note: A *FilterExpression* is applied after the items have already been read; the process of filtering does not consume any additional read capacity units.

For more information, see [Filter Expressions](#) in the *Amazon DynamoDB Developer Guide*.

Note: *FilterExpression* replaces the legacy *ScanFilter* and *ConditionalOperator* parameters.

- **ExpressionAttributeNames** (*dict*) – One or more substitution tokens for attribute names in an expression. The following are some use cases for using *ExpressionAttributeNames*:
 - To access an attribute whose name conflicts with a DynamoDB reserved word.
 - To create a placeholder for repeating occurrences of an attribute name in an expression.
 - To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the `#` character in an expression to dereference an attribute name. For example, consider the following attribute name:

- Percentile

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see [Reserved Words](#) in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for *ExpressionAttributeNames*:

- {`"#P": "Percentile"`}

You could then use this substitution in an expression, as in this example:

- `#P = :val`

Note: Tokens that begin with the `:` character are *expression attribute values*, which are placeholders for the actual value at runtime.

For more information on expression attribute names, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

- (*string*) –
- * (*string*) –

- **ExpressionAttributeValues** (*dict*) – One or more values that can be substituted in an expression.

Use the `:` (colon) character in an expression to dereference an attribute value. For example, suppose that you wanted to check whether the value of the *ProductStatus* attribute was one of the following:

```
Available | Backordered | Discontinued
```

You would first need to specify *ExpressionAttributeValues* as follows:

```
{ ":avail":{"S":"Available"}, ":back":{"S":"Backordered"},
  ":disc":{"S":"Discontinued"} }
```

You could then use these values in an expression, such as this:

```
ProductStatus IN (:avail, :back, :disc)
```

For more information on expression attribute values, see [Specifying Conditions in the Amazon DynamoDB Developer Guide](#).

- *(string)* –
- * *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** *(string)* –

A String data type.

- **N** *(string)* –

A Number data type.

- **B** *(bytes)* –

A Binary data type.

- **SS** *(list)* –

A String Set data type.

- *(string)* –

- **NS** *(list)* –

A Number Set data type.

- *(string)* –

- **BS** *(list)* –

A Binary Set data type.

- *(bytes)* –

- **M** *(dict)* –

A Map of attribute values.

- *(string)* –

- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** *(list)* –

A List of attribute values.

- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one

title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **ConsistentRead** (*boolean*) – A Boolean value that determines the read consistency model during the scan:

- If *ConsistentRead* is `false`, then *Scan* will use eventually consistent reads. The data returned from *Scan* might not contain the results of other recently completed write operations (*PutItem*, *UpdateItem* or *DeleteItem*). The *Scan* response might include some stale data.
- If *ConsistentRead* is `true`, then *Scan* will use strongly consistent reads. All of the write operations that completed before the *Scan* began are guaranteed to be contained in the *Scan* response.

The default setting for *ConsistentRead* is `false`, meaning that eventually consistent reads will be used.

Strongly consistent reads are not supported on global secondary indexes. If you scan a global secondary index with *ConsistentRead* set to `true`, you will receive a *ValidationException*.

Return type dict

Returns

Response Syntax

```
{
  'Items': [
    {
      'string': {
        'S': 'string',
        'N': 'string',
        'B': b'bytes',
        'SS': [
          'string',
        ],
        'NS': [
          'string',
        ],
        'BS': [
          b'bytes',
        ],
        'M': {
          'string': {'... recursive ...'}
        },
        'L': [
          {'... recursive ...'},
        ],
        'NULL': True|False,
        'BOOL': True|False
      }
    ],
  'Count': 123,
  'ScannedCount': 123,
  'LastEvaluatedKey': {
```

```
    'string': {
        'S': 'string',
        'N': 'string',
        'B': b'bytes',
        'SS': [
            'string',
        ],
        'NS': [
            'string',
        ],
        'BS': [
            b'bytes',
        ],
        'M': {
            'string': {'... recursive ...'}
        },
        'L': [
            {'... recursive ...'},
        ],
        'NULL': True|False,
        'BOOL': True|False
    },
    'ConsumedCapacity': {
        'TableName': 'string',
        'CapacityUnits': 123.0,
        'Table': {
            'CapacityUnits': 123.0
        },
        'LocalSecondaryIndexes': {
            'string': {
                'CapacityUnits': 123.0
            }
        },
        'GlobalSecondaryIndexes': {
            'string': {
                'CapacityUnits': 123.0
            }
        }
    }
}
```

Response Structure

- (*dict*) –

Represents the output of a *Scan* operation.

- **Items** (*list*) –

An array of item attributes that match the scan criteria. Each element in this array consists of an attribute name and the value for that attribute.

- * (*dict*) –

- (*string*) –
- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book

item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –

- **M** (*dict*) –

A Map of attribute values.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **Count** (*integer*) –

The number of items in the response.

If you set *ScanFilter* in the request, then *Count* is the number of items returned after the filter was applied, and *ScannedCount* is the number of matching items before the filter was applied.

If you did not use a filter in the request, then *Count* is the same as *ScannedCount*.

– **ScannedCount** (*integer*) –

The number of items evaluated, before any *ScanFilter* is applied. A high *ScannedCount* value with few, or no, *Count* results indicates an inefficient *Scan* operation. For more information, see [Count and ScannedCount](#) in the *Amazon DynamoDB Developer Guide*.

If you did not use a filter in the request, then *ScannedCount* is the same as *Count*.

– **LastEvaluatedKey** (*dict*) –

The primary key of the item where the operation stopped, inclusive of the previous result set. Use this value to start a new operation, excluding this value in the new request.

If *LastEvaluatedKey* is empty, then the “last page” of results has been processed and there is no more data to be retrieved.

If *LastEvaluatedKey* is not empty, it does not necessarily mean that there is more data in the result set. The only way to know when you have reached the end of the result set is when *LastEvaluatedKey* is empty.

· * (*string*) –

· (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

· **S** (*string*) –

A String data type.

· **N** (*string*) –

A Number data type.

· **B** (*bytes*) –

A Binary data type.

· **SS** (*list*) –

A String Set data type.

· (*string*) –

· **NS** (*list*) –

A Number Set data type.

· (*string*) –

· **BS** (*list*) –

A Binary Set data type.

· (*bytes*) –

· **M** (*dict*) –

A Map of attribute values.

- *(string)* –
- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

– **ConsumedCapacity** (*dict*) –

The capacity units consumed by an operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. *ConsumedCapacity* is only returned if the request asked for it. For more information, see [Provisioned Throughput](#) in the *Amazon DynamoDB Developer Guide*.

- * **TableName** (*string*) –

The name of the table that was affected by the operation.

- * **CapacityUnits** (*float*) –

The total number of capacity units consumed by the operation.

- * **Table** (*dict*) –

The amount of throughput consumed on the table affected by the operation.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

- * **LocalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each local index affected by the operation.

- *(string)* –
- *(dict)* –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

- * **GlobalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each global index affected by the operation.

- (*string*) –

- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

update_item (***kwargs*)

Edits an existing item's attributes, or adds a new item to the table if it does not already exist. You can put, delete, or add attribute values. You can also perform a conditional update on an existing item (insert a new attribute name-value pair if it doesn't exist, or replace an existing name-value pair if it has certain expected attribute values). If conditions are specified and the item does not exist, then the operation fails and a new item is not created.

You can also return the item's attribute values in the same *UpdateItem* operation using the *ReturnValues* parameter.

Request Syntax

```
response = client.update_item(
    TableName='string',
    Key={
        'string': {
            'S': 'string',
            'N': 'string',
            'B': b'bytes',
            'SS': [
                'string',
            ],
            'NS': [
                'string',
            ],
            'BS': [
                b'bytes',
            ],
            'M': {
                'string': {'... recursive ...'}
            },
            'L': [
                {'... recursive ...'},
            ],
            'NULL': True|False,
            'BOOL': True|False
        }
    },
    AttributeUpdates={
        'string': {
            'Value': {
                'S': 'string',
```

```

        'N': 'string',
        'B': b'bytes',
        'SS': [
            'string',
        ],
        'NS': [
            'string',
        ],
        'BS': [
            b'bytes',
        ],
        'M': {
            'string': {'... recursive ...'}
        },
        'L': [
            {'... recursive ...'},
        ],
        'NULL': True|False,
        'BOOL': True|False
    },
    'Action': 'ADD'|'PUT'|'DELETE'
}
},
Expected={
    'string': {
        'Value': {
            'S': 'string',
            'N': 'string',
            'B': b'bytes',
            'SS': [
                'string',
            ],
            'NS': [
                'string',
            ],
            'BS': [
                b'bytes',
            ],
            'M': {
                'string': {'... recursive ...'}
            },
            'L': [
                {'... recursive ...'},
            ],
            'NULL': True|False,
            'BOOL': True|False
        },
        'Exists': True|False,
        'ComparisonOperator': 'EQ'|'NE'|'IN'|'LE'|'LT'|'GE'|'GT'|'BETWEEN'|'NOT_NULL'|'...',
        'AttributeValueList': [
            {
                'S': 'string',
                'N': 'string',
                'B': b'bytes',
                'SS': [
                    'string',
                ],
                'NS': [

```

```
        'string',
    ],
    'BS': [
        b'bytes',
    ],
    'M': {
        'string': {'... recursive ...'}
    },
    'L': [
        {'... recursive ...'},
    ],
    'NULL': True|False,
    'BOOL': True|False
},
]
}
},
ConditionalOperator='AND'|'OR',
ReturnValues='NONE'|'ALL_OLD'|'UPDATED_OLD'|'ALL_NEW'|'UPDATED_NEW',
ReturnConsumedCapacity='INDEXES'|'TOTAL'|'NONE',
ReturnItemCollectionMetrics='SIZE'|'NONE',
UpdateExpression='string',
ConditionExpression='string',
ExpressionAttributeNames={
    'string': 'string'
},
ExpressionAttributeValues={
    'string': {
        'S': 'string',
        'N': 'string',
        'B': b'bytes',
        'SS': [
            'string',
        ],
        'NS': [
            'string',
        ],
        'BS': [
            b'bytes',
        ],
        'M': {
            'string': {'... recursive ...'}
        },
        'L': [
            {'... recursive ...'},
        ],
        'NULL': True|False,
        'BOOL': True|False
    }
}
)
```

Parameters

- **TableName** (*string*) – [REQUIRED]

The name of the table containing the item to update.

- **Key** (*dict*) – [REQUIRED]

The primary key of the item to be updated. Each element consists of an attribute

name and a value for that attribute.

For the primary key, you must provide all of the attributes. For example, with a hash type primary key, you only need to provide the hash attribute. For a hash-and-range type primary key, you must provide both the hash attribute and the range attribute.

- *(string)* –
- * *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** *(string)* –

A String data type.

- **N** *(string)* –

A Number data type.

- **B** *(bytes)* –

A Binary data type.

- **SS** *(list)* –

A String Set data type.

- *(string)* –

- **NS** *(list)* –

A Number Set data type.

- *(string)* –

- **BS** *(list)* –

A Binary Set data type.

- *(bytes)* –

- **M** *(dict)* –

A Map of attribute values.

- *(string)* –

- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** *(list)* –

A List of attribute values.

- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

Warning: This is a legacy parameter, for backward compatibility. Applications should use *UpdateExpression* instead. Do not combine list parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception. This parameter can be used for modifying top-level attributes; however, it does not support individual list or map elements.

- **AttributeUpdates** (*dict*) –

The names of attributes to be modified, the action to perform on each, and the new value for each. If you are updating an attribute that is an index key attribute for any indexes on that table, the attribute type must match the index key type defined in the *AttributesDefinition* of the table description. You can use *UpdateItem* to update any nonkey attributes.

Attribute values cannot be null. String and Binary type attributes must have lengths greater than zero. Set type attributes must not be empty. Requests with empty values will be rejected with a *ValidationException* exception.

Each *AttributeUpdates* element consists of an attribute name to modify, along with the following:

- *Value* - The new value, if applicable, for this attribute.
- *Action* - A value that specifies how to perform the update. This action is only valid for an existing attribute whose data type is Number or is a set; do not use ADD for other data types. If an item with the specified primary key is found in the table, the following values perform the following actions:
 - * PUT - Adds the specified attribute to the item. If the attribute already exists, it is replaced by the new value.
- DELETE - Removes the attribute and its value, if no value is specified for DELETE. The data type of the specified value must match the existing value's data type. If a set of values is specified, then those values are subtracted from the old set. For example, if the attribute value was the set [a, b, c] and the DELETE action specifies [a, c], then the final attribute value is [b]. Specifying an empty set is an error.
- ADD - Adds the specified value to the item, if the attribute does not already exist. If the attribute does exist, then the behavior of ADD depends on the data type of the attribute:
 - * If the existing attribute is a number, and if *Value* is also a number, then *Value* is mathematically added to the existing attribute. If *Value* is a negative number, then it is subtracted from the existing attribute. .. note:: If you use ADD to increment or decrement a number value for an item that doesn't exist before the update, DynamoDB uses 0 as the initial value. Similarly, if you use ADD for an existing item to increment or decrement an attribute value that doesn't exist before the update, DynamoDB uses 0 as the initial value. For example, suppose that the item you want to update doesn't have an attribute named *itemcount*, but you decide to ADD the number 3 to this attribute anyway. DynamoDB

will create the *itemcount* attribute, set its initial value to 0 , and finally add 3 to it. The result will be a new *itemcount* attribute, with a value of 3 .

- If the existing data type is a set, and if *Value* is also a set, then *Value* is appended to the existing set. For example, if the attribute value is the set [1 , 2] , and the ADD action specified [3] , then the final attribute value is [1 , 2 , 3] . An error occurs if an ADD action is specified for a set attribute and the attribute type specified does not match the existing set type. Both sets must have the same primitive data type. For example, if the existing data type is a set of strings, *Value* must also be a set of strings.

If no item with the specified key is found in the table, the following values perform the following actions:

- PUT - Causes DynamoDB to create a new item with the specified primary key, and then adds the attribute.
- DELETE - Nothing happens, because attributes cannot be deleted from a nonexistent item. The operation succeeds, but DynamoDB does not create a new item.
- ADD - Causes DynamoDB to create an item with the supplied primary key and number (or set of numbers) for the attribute value. The only data types allowed are Number and Number Set.

If you provide any attributes that are part of an index key, then the data types for those attributes must match those of the schema in the table's attribute definition.

- (*string*) –
- * (*dict*) –

For the *UpdateItem* operation, represents the attributes to be modified, the action to perform on each, and the new value for each.

Note: You cannot use *UpdateItem* to update any primary key attributes. Instead, you will need to delete the item, and then use *PutItem* to create a new item with new attributes.

Attribute values cannot be null; string and binary type attributes must have lengths greater than zero; and set type attributes must not be empty. Requests with empty values will be rejected with a *ValidationException* exception.

- **Value** (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- *(string)* –
- **NS** (*list*) –

A Number Set data type.

- *(string)* –
- **BS** (*list*) –

A Binary Set data type.

- *(bytes)* –
- **M** (*dict*) –

A Map of attribute values.

- *(string)* –
- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **Action** (*string*) –

Specifies how to perform the update. Valid values are PUT (default), DELETE , and ADD . The behavior depends on whether the specified primary key already exists in the table.

If an item with the specified *Key* is found in the table:

- PUT - Adds the specified attribute to the item. If the attribute already exists, it is replaced by the new value.
- DELETE - If no value is specified, the attribute and its value are removed from the item. The data type of the specified value must match the existing value's data type. If a *set* of values is specified, then those values are subtracted from the old set. For example, if the attribute value was the set [a, b, c] and the *DELETE* action specified [a, c] , then the final attribute value would be [b] . Specifying an empty set is an error.

- **ADD** - If the attribute does not already exist, then the attribute and its values are added to the item. If the attribute does exist, then the behavior of **ADD** depends on the data type of the attribute:
 - * If the existing attribute is a number, and if *Value* is also a number, then the *Value* is mathematically added to the existing attribute. If *Value* is a negative number, then it is subtracted from the existing attribute. .. note:: If you use **ADD** to increment or decrement a number value for an item that doesn't exist before the update, DynamoDB uses 0 as the initial value. In addition, if you use **ADD** to update an existing item, and intend to increment or decrement an attribute value which does not yet exist, DynamoDB uses 0 as the initial value. For example, suppose that the item you want to update does not yet have an attribute named *itemcount* , but you decide to **ADD** the number 3 to this attribute anyway, even though it currently does not exist. DynamoDB will create the *itemcount* attribute, set its initial value to 0 , and finally add 3 to it. The result will be a new *itemcount* attribute in the item, with a value of 3 .
 - * If the existing data type is a set, and if the *Value* is also a set, then the *Value* is added to the existing set. (This is a *set* operation, not mathematical addition.) For example, if the attribute value was the set [1 , 2] , and the **ADD** action specified [3] , then the final attribute value would be [1 , 2 , 3] . An error occurs if an Add action is specified for a set attribute and the attribute type specified does not match the existing set type. Both sets must have the same primitive data type. For example, if the existing data type is a set of strings, the *Value* must also be a set of strings. The same holds true for number sets and binary sets.

This action is only valid for an existing attribute whose data type is number or is a set. Do not use **ADD** for any other data types.

If no item with the specified *Key* is found:

- **PUT** - DynamoDB creates a new item with the specified primary key, and then adds the attribute.
- **DELETE** - Nothing happens; there is no attribute to delete.
- **ADD** - DynamoDB creates an item with the supplied primary key and number (or set of numbers) for the attribute value. The only data types allowed are number and number set; no other data types can be specified.

Warning: This is a legacy parameter, for backward compatibility. New applications should use *ConditionExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception.

- **Expected** (*dict*) –

A map of attribute/condition pairs. *Expected* provides a conditional block for the *UpdateItem* operation.

Each element of *Expected* consists of an attribute name, a comparison operator, and one or more values. DynamoDB compares the attribute with the value(s) you supplied, using the comparison operator. For each *Expected* element, the result of the evaluation is either true or false.

If you specify more than one element in the *Expected* map, then by default all of the conditions must evaluate to true. In other words, the conditions are ANDed together. (You can use the *ConditionalOperator* parameter to OR the conditions instead. If you do this, then at least one of the conditions must evaluate to true, rather than all of them.)

If the *Expected* map evaluates to true, then the conditional operation succeeds; otherwise, it fails.

Expected contains the following:

- *AttributeValueList* - One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *ComparisonOperator* being used. For type Number, value comparisons are numeric. String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A, and a is greater than B. For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters. For type Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.
- *ComparisonOperator* - A comparator for evaluating attributes in the *AttributeValueList*. When performing the comparison, DynamoDB uses strongly consistent reads. The following comparison operators are available: EQ | NE | LE | LT | GE | GT | NOT_NULL | NULL | CONTAINS | NOT_CONTAINS | BEGINS_WITH | IN | BETWEEN. The following are descriptions of each comparison operator. * EQ : Equal. EQ is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" }. Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] }.
- NE : Not equal. NE is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" }. Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] }.
- LE : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" }. Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] }.
- LT : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" }. Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] }.
- GE : Greater than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example,

- `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
- **GT** : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
 - **NOT_NULL** : The attribute exists. **NOT_NULL** is supported for all datatypes, including lists and maps. .. note::This operator tests for the existence of an attribute, not its data type. If the data type of attribute “a” is null, and you evaluate it using **NOT_NULL**, the result is a Boolean *true*. This result is because the attribute “a” exists; its data type is not relevant to the **NOT_NULL** comparison operator.
 - **NULL** : The attribute does not exist. **NULL** is supported for all datatypes, including lists and maps. .. note::This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute “a” is null, and you evaluate it using **NULL**, the result is a Boolean *false*. This is because the attribute “a” exists; its data type is not relevant to the **NULL** comparison operator.
 - **CONTAINS** : Checks for a subsequence, or value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS”, “NS”, or “BS”), then the operator evaluates to true if it finds an exact match with any member of the set. **CONTAINS** is supported for lists: When evaluating “a **CONTAINS** b”, “a” can be a list; however, “b” cannot be a set, a map, or a list.
 - **NOT_CONTAINS** : Checks for absence of a subsequence, or absence of a value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS”, “NS”, or “BS”), then the operator evaluates to true if it *does not* find an exact match with any member of the set. **NOT_CONTAINS** is supported for lists: When evaluating “a **NOT CONTAINS** b”, “a” can be a list; however, “b” cannot be a set, a map, or a list.
 - **BEGINS_WITH** : Checks for a prefix. *AttributeValueList* can contain only one *AttributeValue* of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).
 - **IN** : Checks for matching elements within two sets. *AttributeValueList* can contain one or more *AttributeValue* elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
 - **BETWEEN** : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the

first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not compare to `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`

For usage examples of *AttributeValueList* and *ComparisonOperator*, see [Legacy Conditional Parameters](#) in the *Amazon DynamoDB Developer Guide*.

For backward compatibility with previous DynamoDB releases, the following parameters can be used instead of *AttributeValueList* and *ComparisonOperator*:

- *Value* - A value for DynamoDB to compare with an attribute.
- *Exists* - A Boolean value that causes DynamoDB to evaluate the value before attempting the conditional operation: * If *Exists* is `true`, DynamoDB will check to see if that attribute value already exists in the table. If it is found, then the condition evaluates to true; otherwise the condition evaluates to false.
- If *Exists* is `false`, DynamoDB assumes that the attribute value does *not* exist in the table. If in fact the value does not exist, then the assumption is valid and the condition evaluates to true. If the value is found, despite the assumption that it does not exist, the condition evaluates to false.

Note that the default value for *Exists* is `true`.

The *Value* and *Exists* parameters are incompatible with *AttributeValueList* and *ComparisonOperator*. Note that if you use both sets of parameters at once, DynamoDB will return a *ValidationException* exception.

Note: This parameter does not support attributes of type List or Map.

- (*string*) –
- * (*dict*) –

Represents a condition to be compared with an attribute value. This condition can be used with *DeleteItem*, *PutItem* or *UpdateItem* operations; if the comparison evaluates to true, the operation succeeds; if not, the operation fails. You can use *ExpectedAttributeValue* in one of two different ways:

- Use *AttributeValueList* to specify one or more values to compare against an attribute. Use *ComparisonOperator* to specify how you want to perform the comparison. If the comparison evaluates to true, then the conditional operation succeeds.
- Use *Value* to specify a value that DynamoDB will compare against an attribute. If the values match, then *ExpectedAttributeValue* evaluates to true and the conditional operation succeeds. Optionally, you can also set *Exists* to false, indicating that you *do not* expect to find the attribute value in the table. In this case, the conditional operation succeeds only if the comparison evaluates to false.

Value and *Exists* are incompatible with *AttributeValueList* and *ComparisonOperator*. Note that if you use both sets of parameters at once, DynamoDB will return a *ValidationException* exception.

- **Value** (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute

can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –

- **M** (*dict*) –

A Map of attribute values.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **Exists** (*boolean*) –

Causes DynamoDB to evaluate the value before attempting a conditional operation:

- If *Exists* is `true`, DynamoDB will check to see if that attribute value already exists in the table. If it is found, then the operation succeeds. If it is not found, the operation fails with a *ConditionalCheckFailedException*.
- If *Exists* is `false`, DynamoDB assumes that the attribute value does not exist in the table. If in fact the value does not exist, then the assumption is valid and the operation succeeds. If the value is found, despite the assumption that it does not exist, the operation fails with a *ConditionalCheckFailedException*.

The default setting for *Exists* is `true`. If you supply a *Value* all by itself, DynamoDB assumes the attribute exists: You don't have to set *Exists* to `true`, because it is implied.

DynamoDB returns a *ValidationException* if:

- *Exists* is `true` but there is no *Value* to check. (You expect a value to exist, but don't specify what that value is.)
- *Exists* is `false` but you also provide a *Value*. (You cannot expect an attribute to have a value, while also expecting it not to exist.)
- **ComparisonOperator** (*string*) –

A comparator for evaluating attributes in the *AttributeValueList*. For example, equals, greater than, less than, etc.

The following comparison operators are available:

```
EQ | NE | LE | LT | GE | GT | NOT_NULL
| NULL | CONTAINS | NOT_CONTAINS |
BEGINS_WITH | IN | BETWEEN
```

The following are descriptions of each comparison operator.

- **EQ** : Equal. **EQ** is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not equal `{"NS": ["6", "2", "1"]}`.
- **NE** : Not equal. **NE** is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not equal `{"NS": ["6", "2", "1"]}`.
- **LE** : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare

- to `{"NS": ["6", "2", "1"]}`.
- **LT** : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
 - **GE** : Greater than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
 - **GT** : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
 - **NOT_NULL** : The attribute exists. **NOT_NULL** is supported for all datatypes, including lists and maps. .. note::This operator tests for the existence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using **NOT_NULL** , the result is a Boolean *true* . This result is because the attribute “a ” exists; its data type is not relevant to the **NOT_NULL** comparison operator.
 - **NULL** : The attribute does not exist. **NULL** is supported for all datatypes, including lists and maps. .. note::This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using **NULL** , the result is a Boolean *false* . This is because the attribute “a ” exists; its data type is not relevant to the **NULL** comparison operator.
 - **CONTAINS** : Checks for a subsequence, or value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS”, “NS”, or “BS”), then the operator evaluates to true if it finds an exact match with any member of the set. **CONTAINS** is supported for lists: When evaluating “a **CONTAINS** b”, “a” can be a list; however, “b” cannot be a set, a map, or a list.
 - **NOT_CONTAINS** : Checks for absence of a subsequence, or absence of a value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison

is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set ("SS ", "NS ", or "BS "), then the operator evaluates to true if it *does not* find an exact match with any member of the set. NOT_CONTAINS is supported for lists: When evaluating "a NOT CONTAINS b", "a " can be a list; however, "b " cannot be a set, a map, or a list.

- **BEGINS_WITH** : Checks for a prefix. *AttributeValueList* can contain only one *AttributeValue* of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).
- **IN** : Checks for matching elements within two sets. *AttributeValueList* can contain one or more *AttributeValue* elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
- **BETWEEN** : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not compare to { "N" : "6" } . Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }
- **AttributeValueList** (*list*) –

One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *Comparison-Operator* being used.

For type Number, value comparisons are numeric.

String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A , and a is greater than B . For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters .

For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.

For information on specifying data types in JSON, see [JSON Data Format](#) in the *Amazon DynamoDB Developer Guide* .

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one

title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –

- **M** (*dict*) –

A Map of attribute values.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **ConditionalOperator** (*string*) –

Warning: This is a legacy parameter, for backward compatibility. Applications should use *ConditionExpression* instead. Do not combine parameters and expression parameters in a single API call; otherwise, boto3 will return a *ValidationException* exception.

A logical operator to apply to the conditions in the *Expected* map:

- AND - If all of the conditions evaluate to true, then the entire map evaluates to true.
- OR - If at least one of the conditions evaluate to true, then the entire map evaluates to true.

If you omit *ConditionalOperator* , then AND is the default.

The operation will succeed only if the entire map evaluates to true.

Note: This parameter does not support attributes of type List or Map.

- **ReturnValues** (*string*) – Use *ReturnValues* if you want to get the item attributes as they appeared either before or after they were updated. For *UpdateItem* , the valid values are:
 - NONE - If *ReturnValues* is not specified, or if its value is NONE , then nothing is returned. (This setting is the default for *ReturnValues* .)
 - ALL_OLD - If *UpdateItem* overwrote an attribute name-value pair, then the content of the old item is returned.
 - UPDATED_OLD - The old versions of only the updated attributes are returned.
 - ALL_NEW - All of the attributes of the new version of the item are returned.
 - UPDATED_NEW - The new versions of only the updated attributes are returned.
- **ReturnConsumedCapacity** (*string*) – Determines the level of detail about provisioned throughput consumption that is returned in the response:
 - INDEXES - The response includes the aggregate *ConsumedCapacity* for the operation, together with *ConsumedCapacity* for each table and secondary index that was accessed. Note that some operations, such as *GetItem* and *BatchGetItem* , do not access any indexes at all. In these cases, specifying *INDEXES* will only return *ConsumedCapacity* information for table(s).
 - TOTAL - The response includes only the aggregate *ConsumedCapacity* for the operation.
 - NONE - No *ConsumedCapacity* details are included in the response.
- **ReturnItemCollectionMetrics** (*string*) – Determines whether item collection metrics are returned. If set to *SIZE* , the response includes statistics about item collections, if any, that were modified during the operation are returned in the response. If set to *NONE* (the default), no statistics are returned.
- **UpdateExpression** (*string*) – An expression that defines one or more attributes to be updated, the action to be performed on them, and new value(s) for them.

The following action values are available for *UpdateExpression* .

- SET - Adds one or more attributes and values to an item. If any of these attribute already exist, they are replaced by the new values. You can also use SET to add or subtract from an attribute that is of type Number. For example: SET myNum = myNum + :val SET supports the following functions: * if_not_exists (path, operand) - if the item does not contain an attribute at the specified path, then if_not_exists evaluates to operand; otherwise, it evaluates to path. You can use this function to avoid overwriting an attribute that may already be present in the item.
- list_append (operand, operand) - evaluates to a list with a new element added to it. You can append the new element to the start or the end of the list by reversing the order of the operands.

These function names are case-sensitive.

- REMOVE - Removes one or more attributes from an item.
- ADD - Adds the specified value to the item, if the attribute does not already exist. If the attribute does exist, then the behavior of ADD depends on the data type of the attribute:
 - * If the existing attribute is a number, and if *Value* is also a number, then *Value* is mathematically added to the existing attribute. If *Value* is a negative number, then it is subtracted from the existing attribute. .. note:: If you use ADD to increment or decrement a number value for an item that doesn't exist before the update, DynamoDB uses 0 as the initial value. Similarly, if you use ADD for an existing item to increment or decrement an attribute value that doesn't exist before the update, DynamoDB uses 0 as the initial value. For example, suppose that the item you want to update doesn't have an attribute named *itemcount* , but you decide to ADD the number 3 to this attribute anyway. DynamoDB will create the *itemcount* attribute, set its initial value to 0 , and finally add 3 to it. The result will be a new *itemcount* attribute in the item, with a value of 3 .
 - If the existing data type is a set and if *Value* is also a set, then *Value* is added to the existing set. For example, if the attribute value is the set [1, 2] , and the ADD action specified [3] , then the final attribute value is [1, 2, 3] . An error occurs if an ADD action is specified for a set attribute and the attribute type specified does not match the existing set type. Both sets must have the same primitive data type. For example, if the existing data type is a set of strings, the *Value* must also be a set of strings.

Warning: The ADD action only supports Number and set data types. In addition, ADD can only be used on top-level attributes, not nested attributes.

- DELETE - Deletes an element from a set. If a set of values is specified, then those values are subtracted from the old set. For example, if the attribute value was the set [a, b, c] and the DELETE action specifies [a, c] , then the final attribute value is [b] . Specifying an empty set is an error. .. warning::The DELETE action only supports set data types. In addition, DELETE can only be used on top-level attributes, not nested attributes.

You can have many actions in a single expression, such as the following: SET a=:value1, b=:value2 DELETE :value3, :value4, :value5

For more information on update expressions, see [Modifying Items and Attributes](#) in the *Amazon DynamoDB Developer Guide* .

Note: *UpdateExpression* replaces the legacy *AttributeUpdates* parameter.

- **ConditionExpression** (*string*) – A condition that must be satisfied in order for a conditional update to succeed.

An expression can contain any of the following:

- Functions: attribute_exists | attribute_not_exists | attribute_type | contains | begins_with | size
These function names are case-sensitive.
- Comparison operators: = | > | < | >= | <= | BETWEEN | IN
- Logical operators: AND | OR | NOT

For more information on condition expressions, see [Specifying Conditions](#) in the *Amazon DynamoDB Developer Guide* .

Note: *ConditionExpression* replaces the legacy *ConditionalOperator* and *Expected* parameters.

- **ExpressionAttributeNames** (*dict*) – One or more substitution tokens for attribute names in an expression. The following are some use cases for using *ExpressionAttributeNames* :

- To access an attribute whose name conflicts with a DynamoDB reserved word.
- To create a placeholder for repeating occurrences of an attribute name in an expression.
- To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the # character in an expression to dereference an attribute name. For example, consider the following attribute name:

- Percentile

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see [Reserved Words](#) in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for *ExpressionAttributeNames* :

- { "#P": "Percentile" }

You could then use this substitution in an expression, as in this example:

- #P = :val

Note: Tokens that begin with the : character are *expression attribute values* , which are placeholders for the actual value at runtime.

For more information on expression attribute names, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide* .

- (*string*) –
- * (*string*) –

- **ExpressionAttributeValues** (*dict*) – One or more values that can be substituted in an expression.

Use the : (colon) character in an expression to dereference an attribute value. For example, suppose that you wanted to check whether the value of the *ProductStatus* attribute was one of the following:

```
Available | Backordered | Discontinued
```

You would first need to specify *ExpressionAttributeValues* as follows:

```
{ ":avail":{"S":"Available"}, ":back":{"S":"Backordered"},  
  ":disc":{"S":"Discontinued"} }
```

You could then use these values in an expression, such as this:

```
ProductStatus IN (:avail, :back, :disc)
```

For more information on expression attribute values, see [Specifying Conditions](#) in the *Amazon DynamoDB Developer Guide* .

- (*string*) –
- * (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –
A String data type.
- **N** (*string*) –
A Number data type.
- **B** (*bytes*) –
A Binary data type.
- **SS** (*list*) –
A String Set data type.
- (*string*) –
- **NS** (*list*) –
A Number Set data type.
- (*string*) –
- **BS** (*list*) –
A Binary Set data type.
- (*bytes*) –
- **M** (*dict*) –
A Map of attribute values.
- (*string*) –
- (*dict*) –
Represents the data for an attribute. You can set one, and only one, of the elements.
Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.
- **L** (*list*) –
A List of attribute values.
- (*dict*) –
Represents the data for an attribute. You can set one, and only one, of the elements.
Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.
- **NULL** (*boolean*) –
A Null data type.
- **BOOL** (*boolean*) –
A Boolean data type.

Return type dict

Returns

Response Syntax

```
{
    'Attributes': {
```

```
    'string': {
        'S': 'string',
        'N': 'string',
        'B': b'bytes',
        'SS': [
            'string',
        ],
        'NS': [
            'string',
        ],
        'BS': [
            b'bytes',
        ],
        'M': {
            'string': {'... recursive ...'}
        },
        'L': [
            {'... recursive ...'},
        ],
        'NULL': True|False,
        'BOOL': True|False
    }
},
'ConsumedCapacity': {
    'TableName': 'string',
    'CapacityUnits': 123.0,
    'Table': {
        'CapacityUnits': 123.0
    },
    'LocalSecondaryIndexes': {
        'string': {
            'CapacityUnits': 123.0
        }
    },
    'GlobalSecondaryIndexes': {
        'string': {
            'CapacityUnits': 123.0
        }
    }
},
'ItemCollectionMetrics': {
    'ItemCollectionKey': {
        'string': {
            'S': 'string',
            'N': 'string',
            'B': b'bytes',
            'SS': [
                'string',
            ],
            'NS': [
                'string',
            ],
            'BS': [
                b'bytes',
            ],
            'M': {
                'string': {'... recursive ...'}
            },
        },
    },
}
```

```

        'L': [
            {'... recursive ...'},
        ],
        'NULL': True|False,
        'BOOL': True|False
    }
},
'SizeEstimateRangeGB': [
    123.0,
]
}
}

```

Response Structure

- *(dict)* –

Represents the output of an *UpdateItem* operation.

- **Attributes** (*dict*) –

A map of attribute values as they appeared before the *UpdateItem* operation. This map only appears if *ReturnValues* was specified as something other than *NONE* in the request. Each element represents one attribute.

- * *(string)* –

- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –

- **M** (*dict*) –

A Map of attribute values.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

– **ConsumedCapacity** (*dict*) –

The capacity units consumed by an operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. *ConsumedCapacity* is only returned if the request asked for it. For more information, see [Provisioned Throughput](#) in the *Amazon DynamoDB Developer Guide*.

- * **TableName** (*string*) –

The name of the table that was affected by the operation.

- * **CapacityUnits** (*float*) –

The total number of capacity units consumed by the operation.

- * **Table** (*dict*) –

The amount of throughput consumed on the table affected by the operation.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

- * **LocalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each local index affected by the operation.

- (*string*) –
- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

* **GlobalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each global index affected by the operation.

- (*string*) –
- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

– **ItemCollectionMetrics** (*dict*) –

Information about item collections, if any, that were affected by the operation. *ItemCollectionMetrics* is only returned if the request asked for it. If the table does not have any local secondary indexes, this information is not returned in the response.

* **ItemCollectionKey** (*dict*) –

The hash key value of the item collection. This value is the same as the hash key of the item.

- (*string*) –
- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –

- **M** (*dict*) –

A Map of attribute values.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- * **SizeEstimateRangeGB** (*list*) –

An estimate of item collection size, in gigabytes. This value is a two-element array containing a lower bound and an upper bound for the estimate. The estimate includes the size of all the items in the table, plus the size of all attributes projected into all of the local secondary indexes on that table. Use this estimate to measure whether a local secondary index is approaching its size limit.

The estimate is subject to change over time; therefore, do not rely on the precision or accuracy of the estimate.

- (*float*) –

update_table (**kwargs)

Modifies the provisioned throughput settings, global secondary indexes, or DynamoDB Streams settings for a given table.

You can only perform one of the following operations at once:

- Modify the provisioned throughput settings of the table.
- Enable or disable Streams on the table.
- Remove a global secondary index from the table.
- Create a new global secondary index on the table. Once the index begins backfilling, you can use *UpdateTable* to perform other operations.

UpdateTable is an asynchronous operation; while it is executing, the table status changes from **ACTIVE** to **UPDATING**. While it is **UPDATING**, you cannot issue another *UpdateTable* request. When the table returns to the **ACTIVE** state, the *UpdateTable* operation is complete.

Request Syntax

```
response = client.update_table(  
    AttributeDefinitions=[  
        {
```

```

        'AttributeName': 'string',
        'AttributeType': 'S'|'N'|'B'
    },
],
TableName='string',
ProvisionedThroughput={
    'ReadCapacityUnits': 123,
    'WriteCapacityUnits': 123
},
GlobalSecondaryIndexUpdates=[
    {
        'Update': {
            'IndexName': 'string',
            'ProvisionedThroughput': {
                'ReadCapacityUnits': 123,
                'WriteCapacityUnits': 123
            }
        },
        'Create': {
            'IndexName': 'string',
            'KeySchema': [
                {
                    'AttributeName': 'string',
                    'KeyType': 'HASH'|'RANGE'
                },
            ],
            'Projection': {
                'ProjectionType': 'ALL'|'KEYS_ONLY'|'INCLUDE',
                'NonKeyAttributes': [
                    'string',
                ]
            },
            'ProvisionedThroughput': {
                'ReadCapacityUnits': 123,
                'WriteCapacityUnits': 123
            }
        },
        'Delete': {
            'IndexName': 'string'
        }
    },
],
StreamSpecification={
    'StreamEnabled': True|False,
    'StreamViewType': 'NEW_IMAGE'|'OLD_IMAGE'|'NEW_AND_OLD_IMAGES'|'KEYS_ONLY'
}
)

```

Parameters

- **AttributeDefinitions** (*list*) – An array of attributes that describe the key schema for the table and indexes. If you are adding a new global secondary index to the table, *AttributeDefinitions* must include the key element(s) of the new index.

– (*dict*) –

Represents an attribute for describing the key schema for the table and indexes.

* **AttributeName** (*string*) – **[REQUIRED]**

A name for the attribute.

* **AttributeType** (*string*) – [REQUIRED]

The data type for the attribute.

• **TableName** (*string*) – [REQUIRED]

The name of the table to be updated.

• **ProvisionedThroughput** (*dict*) – Represents the provisioned throughput settings for a specified table or index. The settings can be modified using the *UpdateTable* operation.

For current minimum and maximum provisioned throughput values, see [Limits](#) in the *Amazon DynamoDB Developer Guide*.

– **ReadCapacityUnits** (*integer*) – [REQUIRED]

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a *ThrottlingException*. For more information, see [Specifying Read and Write Requirements](#) in the *Amazon DynamoDB Developer Guide*.

– **WriteCapacityUnits** (*integer*) – [REQUIRED]

The maximum number of writes consumed per second before DynamoDB returns a *ThrottlingException*. For more information, see [Specifying Read and Write Requirements](#) in the *Amazon DynamoDB Developer Guide*.

• **GlobalSecondaryIndexUpdates** (*list*) – An array of one or more global secondary indexes for the table. For each index in the array, you can request one action:

- *Create* - add a new global secondary index to the table.
- *Update* - modify the provisioned throughput settings of an existing global secondary index.
- *Delete* - remove a global secondary index from the table.

For more information, see [Managing Global Secondary Indexes](#) in the *Amazon DynamoDB Developer Guide*.

– (*dict*) –

Represents one of the following:

- * A new global secondary index to be added to an existing table.
- * New provisioned throughput parameters for an existing global secondary index.
- * An existing global secondary index to be removed from an existing table.

* **Update** (*dict*) –

The name of an existing global secondary index, along with new provisioned throughput settings to be applied to that index.

· **IndexName** (*string*) – [REQUIRED]

The name of the global secondary index to be updated.

· **ProvisionedThroughput** (*dict*) – [REQUIRED]

Represents the provisioned throughput settings for a specified table or index. The settings can be modified using the *UpdateTable* operation.

For current minimum and maximum provisioned throughput values, see [Limits](#) in the *Amazon DynamoDB Developer Guide*.

· **ReadCapacityUnits** (*integer*) – [REQUIRED]

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a *ThrottlingException* . For more information, see [Specifying Read and Write Requirements](#) in the *Amazon DynamoDB Developer Guide* .

- **WriteCapacityUnits** (*integer*) – [REQUIRED]

The maximum number of writes consumed per second before DynamoDB returns a *ThrottlingException* . For more information, see [Specifying Read and Write Requirements](#) in the *Amazon DynamoDB Developer Guide* .

* **Create** (*dict*) –

The parameters required for creating a global secondary index on an existing table:

- IndexName
- KeySchema
- AttributeDefinitions
- Projection
- ProvisionedThroughput
- **IndexName** (*string*) – [REQUIRED]

The name of the global secondary index to be created.

- **KeySchema** (*list*) – [REQUIRED]

The key schema for the global secondary index.

- (*dict*) –

Represents *a single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement* . A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.

- **AttributeName** (*string*) – [REQUIRED]

The name of a key attribute.

- **KeyType** (*string*) – [REQUIRED]

The attribute data, consisting of the data type and the attribute value itself.

- **Projection** (*dict*) – [REQUIRED]

Represents attributes that are copied (projected) from the table into an index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

- **ProjectionType** (*string*) –

The set of attributes that are projected into the index:

- KEYS_ONLY - Only the index and primary keys are projected into the index.
- INCLUDE - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes* .

- **ALL** - All of the table attributes are projected into the index.
- **NonKeyAttributes** (*list*) –

Represents the non-key attribute names which will be projected into the index.

For local secondary indexes, the total count of *NonKeyAttributes* summed across all of the local secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.

- (*string*) –
- **ProvisionedThroughput** (*dict*) – **[REQUIRED]**

Represents the provisioned throughput settings for a specified table or index. The settings can be modified using the *UpdateTable* operation.

For current minimum and maximum provisioned throughput values, see [Limits](#) in the *Amazon DynamoDB Developer Guide*.

- **ReadCapacityUnits** (*integer*) – **[REQUIRED]**

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a *ThrottlingException*. For more information, see [Specifying Read and Write Requirements](#) in the *Amazon DynamoDB Developer Guide*.

- **WriteCapacityUnits** (*integer*) – **[REQUIRED]**

The maximum number of writes consumed per second before DynamoDB returns a *ThrottlingException*. For more information, see [Specifying Read and Write Requirements](#) in the *Amazon DynamoDB Developer Guide*.

- * **Delete** (*dict*) –

The name of an existing global secondary index to be removed.

- **IndexName** (*string*) – **[REQUIRED]**

The name of the global secondary index to be deleted.

- **StreamSpecification** (*dict*) – Represents the DynamoDB Streams configuration for the table.

Note: You will receive a *ResourceInUseException* if you attempt to enable a stream on a table that already has a stream, or if you attempt to disable a stream on a table which does not have a stream.

- **StreamEnabled** (*boolean*) –

Indicates whether DynamoDB Streams is enabled (true) or disabled (false) on the table.

- **StreamViewType** (*string*) –

The DynamoDB Streams settings for the table. These settings consist of:

- * *StreamEnabled* - Indicates whether DynamoDB Streams is enabled (true) or disabled (false) on the table.
- * *StreamViewType* - When an item in the table is modified, *StreamViewType* determines what information is written to the stream for this table. Valid values for *StreamViewType* are: *

KEYS_ONLY - Only the key attributes of the modified item are written to the stream.

* *NEW_IMAGE* - The entire item, as it appears after it was modified, is written to the stream.

* *OLD_IMAGE* - The entire item, as it appeared before it was modified, is written to the stream.

* *NEW_AND_OLD_IMAGES* - Both the new and the old item images of the item are written to the stream.

Return type dict

Returns

Response Syntax

```
{
  'TableDescription': {
    'AttributeDefinitions': [
      {
        'AttributeName': 'string',
        'AttributeType': 'S'|'N'|'B'
      },
    ],
    'TableName': 'string',
    'KeySchema': [
      {
        'AttributeName': 'string',
        'KeyType': 'HASH'|'RANGE'
      },
    ],
    'TableStatus': 'CREATING'|'UPDATING'|'DELETING'|'ACTIVE',
    'CreationDateTime': datetime(2015, 1, 1),
    'ProvisionedThroughput': {
      'LastIncreaseDateTime': datetime(2015, 1, 1),
      'LastDecreaseDateTime': datetime(2015, 1, 1),
      'NumberOfDecreasesToday': 123,
      'ReadCapacityUnits': 123,
      'WriteCapacityUnits': 123
    },
    'TableSizeBytes': 123,
    'ItemCount': 123,
    'TableArn': 'string',
    'LocalSecondaryIndexes': [
      {
        'IndexName': 'string',
        'KeySchema': [
          {
            'AttributeName': 'string',
            'KeyType': 'HASH'|'RANGE'
          },
        ],
        'Projection': {
          'ProjectionType': 'ALL'|'KEYS_ONLY'|'INCLUDE',
          'NonKeyAttributes': [
            'string',
          ]
        },
        'IndexSizeBytes': 123,
        'ItemCount': 123,
        'IndexArn': 'string'
      },
    ],
  },
}
```

```
    },
  ],
  'GlobalSecondaryIndexes': [
    {
      'IndexName': 'string',
      'KeySchema': [
        {
          'AttributeName': 'string',
          'KeyType': 'HASH' | 'RANGE'
        },
      ],
      'Projection': {
        'ProjectionType': 'ALL' | 'KEYS_ONLY' | 'INCLUDE',
        'NonKeyAttributes': [
          'string',
        ]
      },
      'IndexStatus': 'CREATING' | 'UPDATING' | 'DELETING' | 'ACTIVE',
      'Backfilling': True | False,
      'ProvisionedThroughput': {
        'LastIncreaseDateTime': datetime(2015, 1, 1),
        'LastDecreaseDateTime': datetime(2015, 1, 1),
        'NumberOfDecreasesToday': 123,
        'ReadCapacityUnits': 123,
        'WriteCapacityUnits': 123
      },
      'IndexSizeBytes': 123,
      'ItemCount': 123,
      'IndexArn': 'string'
    },
  ],
  'StreamSpecification': {
    'StreamEnabled': True | False,
    'StreamViewType': 'NEW_IMAGE' | 'OLD_IMAGE' | 'NEW_AND_OLD_IMAGES' | 'KEYS_ONLY',
  },
  'LatestStreamLabel': 'string',
  'LatestStreamArn': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of an *UpdateTable* operation.

- **TableDescription** (*dict*) –

Represents the properties of a table.

- * **AttributeDefinitions** (*list*) –

An array of *AttributeDefinition* objects. Each of these objects describes one attribute in the table and index key schema.

Each *AttributeDefinition* object in this array is composed of:

- *AttributeName* - The name of the attribute.
- *AttributeType* - The data type for the attribute.

- (*dict*) –

Represents an attribute for describing the key schema for the table and indexes.

- **AttributeName** (*string*) –
A name for the attribute.
- **AttributeType** (*string*) –
The data type for the attribute.
- * **TableName** (*string*) –
The name of the table.
- * **KeySchema** (*list*) –
The primary key structure for the table. Each *KeySchemaElement* consists of:
 - *AttributeName* - The name of the attribute.
 - *KeyType* - The key type for the attribute. Can be either `HASH` or `RANGE`.
 For more information about primary keys, see [Primary Key](#) in the *Amazon DynamoDB Developer Guide*.
 - (*dict*) –
Represents a *single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement*. A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.
 - **AttributeName** (*string*) –
The name of a key attribute.
 - **KeyType** (*string*) –
The attribute data, consisting of the data type and the attribute value itself.
- * **TableStatus** (*string*) –
The current state of the table:
 - *CREATING* - The table is being created.
 - *UPDATING* - The table is being updated.
 - *DELETING* - The table is being deleted.
 - *ACTIVE* - The table is ready for use.
- * **CreationDateTime** (*datetime*) –
The date and time when the table was created, in [UNIX epoch time](#) format.
- * **ProvisionedThroughput** (*dict*) –
The provisioned throughput settings for the table, consisting of read and write capacity units, along with data about increases and decreases.
 - **LastIncreaseDateTime** (*datetime*) –
The date and time of the last provisioned throughput increase for this table.
 - **LastDecreaseDateTime** (*datetime*) –

The date and time of the last provisioned throughput decrease for this table.

- **NumberOfDecreasesToday** (*integer*) –

The number of provisioned throughput decreases for this table during this UTC calendar day. For current maximums on provisioned throughput decreases, see [Limits](#) in the *Amazon DynamoDB Developer Guide*.

- **ReadCapacityUnits** (*integer*) –

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a *ThrottlingException*. Eventually consistent reads require less effort than strongly consistent reads, so a setting of 50 *ReadCapacityUnits* per second provides 100 eventually consistent *ReadCapacityUnits* per second.

- **WriteCapacityUnits** (*integer*) –

The maximum number of writes consumed per second before DynamoDB returns a *ThrottlingException*.

- * **TableSizeBytes** (*integer*) –

The total size of the specified table, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- * **ItemCount** (*integer*) –

The number of items in the specified table. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- * **TableArn** (*string*) –

The Amazon Resource Name (ARN) that uniquely identifies the table.

- * **LocalSecondaryIndexes** (*list*) –

Represents one or more local secondary indexes on the table. Each index is scoped to a given hash key value. Tables with one or more local secondary indexes are subject to an item collection size limit, where the amount of data within a given item collection cannot exceed 10 GB. Each element is composed of:

- *IndexName* - The name of the local secondary index.
- *KeySchema* - Specifies the complete index key schema. The attribute names in the key schema must be between 1 and 255 characters (inclusive). The key schema must begin with the same hash key attribute as the table.
- *Projection* - Specifies attributes that are copied (projected) from the table into the index. These are in addition to the primary key attributes and index key attributes, which are automatically projected. Each attribute specification is composed of: * *ProjectionType* - One of the following: * *KEYS_ONLY* - Only the index and primary keys are projected into the index.
- *INCLUDE* - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes*.
- *ALL* - All of the table attributes are projected into the index.
- *NonKeyAttributes* - A list of one or more non-key attribute

names that are projected into the secondary index. The total count of attributes provided in *NonKeyAttributes* , summed across all of the secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.

- *IndexSizeBytes* - Represents the total size of the index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.
- *ItemCount* - Represents the number of items in the index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

If the table is in the `DELETING` state, no information about indexes will be returned.

- *(dict)* –

Represents the properties of a local secondary index.

- **IndexName** (*string*) –

Represents the name of the local secondary index.

- **KeySchema** (*list*) –

The complete index key schema, which consists of one or more pairs of attribute names and key types (`HASH` or `RANGE`).

- *(dict)* –

Represents *a single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement* . A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.

- **AttributeName** (*string*) –

The name of a key attribute.

- **KeyType** (*string*) –

The attribute data, consisting of the data type and the attribute value itself.

- **Projection** (*dict*) –

Represents attributes that are copied (projected) from the table into an index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

- **ProjectionType** (*string*) –

The set of attributes that are projected into the index:

- `KEYS_ONLY` - Only the index and primary keys are projected into the index.
- `INCLUDE` - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes* .
- `ALL` - All of the table attributes are projected into the index.

- **NonKeyAttributes** (*list*) –

Represents the non-key attribute names which will be projected into the index.

For local secondary indexes, the total count of *NonKeyAttributes* summed across all of the local secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.

- (*string*) –

- **IndexSizeBytes** (*integer*) –

The total size of the specified index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- **ItemCount** (*integer*) –

The number of items in the specified index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- **IndexArn** (*string*) –

The Amazon Resource Name (ARN) that uniquely identifies the index.

- * **GlobalSecondaryIndexes** (*list*) –

The global secondary indexes, if any, on the table. Each index is scoped to a given hash key value. Each element is composed of:

- *Backfilling* - If true, then the index is currently in the backfilling phase. Backfilling occurs only when a new global secondary index is added to the table; it is the process by which DynamoDB populates the new index with data from the table. (This attribute does not appear for indexes that were created during a *CreateTable* operation.)
- *IndexName* - The name of the global secondary index.
- *IndexSizeBytes* - The total size of the global secondary index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.
- *IndexStatus* - The current status of the global secondary index:
 - * *CREATING* - The index is being created.
 - * *UPDATING* - The index is being updated.
 - * *DELETING* - The index is being deleted.
 - * *ACTIVE* - The index is ready for use.
- *ItemCount* - The number of items in the global secondary index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.
- *KeySchema* - Specifies the complete index key schema. The attribute names in the key schema must be between 1 and 255 characters (inclusive). The key schema must begin with the same hash key attribute as the table.
- *Projection* - Specifies attributes that are copied (projected) from the table into the index. These are in addition to the primary key attributes and index key attributes, which are automatically projected. Each attribute specification is composed of:
 - * *ProjectionType* - One of the following:
 - * *KEYS_ONLY* - Only the index and primary keys are projected into the index.

- **INCLUDE** - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes*.
- **ALL** - All of the table attributes are projected into the index.
- **NonKeyAttributes** - A list of one or more non-key attribute names that are projected into the secondary index. The total count of attributes provided in *NonKeyAttributes*, summed across all of the secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.
- **ProvisionedThroughput** - The provisioned throughput settings for the global secondary index, consisting of read and write capacity units, along with data about increases and decreases.

If the table is in the `DELETING` state, no information about indexes will be returned.

- *(dict)* –

Represents the properties of a global secondary index.

- **IndexName** (*string*) –

The name of the global secondary index.

- **KeySchema** (*list*) –

The complete key schema for the global secondary index, consisting of one or more pairs of attribute names and key types (`HASH` or `RANGE`).

- *(dict)* –

Represents *a single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement*. A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.

- **AttributeName** (*string*) –

The name of a key attribute.

- **KeyType** (*string*) –

The attribute data, consisting of the data type and the attribute value itself.

- **Projection** (*dict*) –

Represents attributes that are copied (projected) from the table into an index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

- **ProjectionType** (*string*) –

The set of attributes that are projected into the index:

- **KEYS_ONLY** - Only the index and primary keys are projected into the index.
- **INCLUDE** - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKey-*

Attributes .

- **ALL** - All of the table attributes are projected into the index.
- **NonKeyAttributes** (*list*) –

Represents the non-key attribute names which will be projected into the index.

For local secondary indexes, the total count of *NonKeyAttributes* summed across all of the local secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.

- (*string*) –
- **IndexStatus** (*string*) –

The current state of the global secondary index:

- **CREATING** - The index is being created.
- **UPDATING** - The index is being updated.
- **DELETING** - The index is being deleted.
- **ACTIVE** - The index is ready for use.
- **Backfilling** (*boolean*) –

Indicates whether the index is currently backfilling. *Backfilling* is the process of reading items from the table and determining whether they can be added to the index. (Not all items will qualify: For example, a hash key attribute cannot have any duplicates.) If an item can be added to the index, DynamoDB will do so. After all items have been processed, the backfilling operation is complete and *Backfilling* is false.

Note: For indexes that were created during a *CreateTable* operation, the *Backfilling* attribute does not appear in the *DescribeTable* output.

- **ProvisionedThroughput** (*dict*) –

Represents the provisioned throughput settings for the table, consisting of read and write capacity units, along with data about increases and decreases.

- **LastIncreaseDateTime** (*datetime*) –

The date and time of the last provisioned throughput increase for this table.

- **LastDecreaseDateTime** (*datetime*) –

The date and time of the last provisioned throughput decrease for this table.

- **NumberOfDecreasesToday** (*integer*) –

The number of provisioned throughput decreases for this table during this UTC calendar day. For current maximums on provisioned throughput decreases, see [Limits](#) in the *Amazon DynamoDB Developer Guide* .

- **ReadCapacityUnits** (*integer*) –

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a *ThrottlingException* . Eventually consistent reads require less effort than strongly consistent reads, so a setting of 50 *ReadCapacityUnits* per

second provides 100 eventually consistent *ReadCapacityUnits* per second.

- **WriteCapacityUnits** (*integer*) –

The maximum number of writes consumed per second before DynamoDB returns a *ThrottlingException*.

- **IndexSizeBytes** (*integer*) –

The total size of the specified index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- **ItemCount** (*integer*) –

The number of items in the specified index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- **IndexArn** (*string*) –

The Amazon Resource Name (ARN) that uniquely identifies the index.

- * **StreamSpecification** (*dict*) –

The current DynamoDB Streams configuration for the table.

- **StreamEnabled** (*boolean*) –

Indicates whether DynamoDB Streams is enabled (true) or disabled (false) on the table.

- **StreamViewType** (*string*) –

The DynamoDB Streams settings for the table. These settings consist of:

- *StreamEnabled* - Indicates whether DynamoDB Streams is enabled (true) or disabled (false) on the table.
- *StreamViewType* - When an item in the table is modified, *StreamViewType* determines what information is written to the stream for this table. Valid values for *StreamViewType* are: * *KEYS_ONLY* - Only the key attributes of the modified item are written to the stream.
- *NEW_IMAGE* - The entire item, as it appears after it was modified, is written to the stream.
- *OLD_IMAGE* - The entire item, as it appeared before it was modified, is written to the stream.
- *NEW_AND_OLD_IMAGES* - Both the new and the old item images of the item are written to the stream.

- * **LatestStreamLabel** (*string*) –

A timestamp, in ISO 8601 format, for this stream.

Note that *LatestStreamLabel* is not a unique identifier for the stream, because it is possible that a stream from another table might have the same timestamp. However, the combination of the following three elements is guaranteed to be unique:

- the AWS customer ID.
- the table name.
- the *StreamLabel*.

- * **LatestStreamArn** (*string*) –

The Amazon Resource Name (ARN) that uniquely identifies the latest stream for this table.

Paginators

The available paginators are:

- `DynamoDB.Paginator.ListTables`
- `DynamoDB.Paginator.Query`
- `DynamoDB.Paginator.Scan`

class `DynamoDB.Paginator.ListTables`

```
paginator = client.get_paginator('list_tables')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `DynamoDB.Client.list_tables()`.

Request Syntax

```
response_iterator = paginator.paginate(
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
    'TableNames': [
        'string',
    ],
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a *ListTables* operation.

– **TableNames** (*list*) –

The names of the tables associated with the current account at the current endpoint. The maximum size of this array is 100.

If *LastEvaluatedTableName* also appears in the output, you can use this value as the *ExclusiveStartTableName* parameter in a subsequent *ListTables* request and obtain the next page of results.

* (*string*) –

– **NextToken** (*string*) –

A token to resume pagination.

class `DynamoDB.Paginator.Query`

```
paginator = client.get_paginator('query')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `DynamoDB.Client.query()`.

Request Syntax

```
response_iterator = paginator.paginate(
    TableName='string',
    IndexName='string',
    Select='ALL_ATTRIBUTES' | 'ALL_PROJECTED_ATTRIBUTES' | 'SPECIFIC_ATTRIBUTES' | 'COUNT',
    AttributesToGet=[
        'string',
    ],
    ConsistentRead=True|False,
    KeyConditions={
        'string': {
            'AttributeValueList': [
                {
                    'S': 'string',
                    'N': 'string',
                    'B': b'bytes',
                    'SS': [
                        'string',
                    ],
                    'NS': [
                        'string',
                    ],
                    'BS': [
                        b'bytes',
                    ],
                    'M': {
                        'string': {'... recursive ...'}
                    },
                    'L': [
                        {'... recursive ...'},
                    ],
                    'NULL': True|False,
                    'BOOL': True|False
                },
            ],
            'ComparisonOperator': 'EQ' | 'NE' | 'IN' | 'LE' | 'LT' | 'GE' | 'GT' | 'BETWEEN' | 'NOT_NULL' | '...
```

```
},
QueryFilter={
    'string': {
        'AttributeValueList': [
            {
                'S': 'string',
                'N': 'string',
                'B': b'bytes',
                'SS': [
                    'string',
                ],
                'NS': [
                    'string',
                ],
                'BS': [
                    b'bytes',
                ],
                'M': {
                    'string': {'... recursive ...'}
                },
                'L': [
                    {'... recursive ...'},
                ],
                'NULL': True|False,
                'BOOL': True|False
            },
        ],
        'ComparisonOperator': 'EQ'|'NE'|'IN'|'LE'|'LT'|'GE'|'GT'|'BETWEEN'|'NOT_NULL'|'...'
    },
},
ConditionalOperator='AND'|'OR',
ScanIndexForward=True|False,
ReturnConsumedCapacity='INDEXES'|'TOTAL'|'NONE',
ProjectionExpression='string',
FilterExpression='string',
KeyConditionExpression='string',
ExpressionAttributeNames={
    'string': 'string'
},
ExpressionAttributeValues={
    'string': {
        'S': 'string',
        'N': 'string',
        'B': b'bytes',
        'SS': [
            'string',
        ],
        'NS': [
            'string',
        ],
        'BS': [
            b'bytes',
        ],
        'M': {
            'string': {'... recursive ...'}
        },
        'L': [
            {'... recursive ...'},
        ],
    },
}
```

```

    ],
    'NULL': True|False,
    'BOOL': True|False
  }
},
PaginationConfig={
    'MaxItems': 123,
    'PageSize': 123,
    'StartingToken': 'string'
}
)

```

Parameters

- **TableName** (*string*) – [REQUIRED]

The name of the table containing the requested items.

- **IndexName** (*string*) – The name of an index to query. This index can be any local secondary index or global secondary index on the table. Note that if you use the *IndexName* parameter, you must also provide *TableName*.
- **Select** (*string*) – The attributes to be returned in the result. You can retrieve all item attributes, specific item attributes, the count of matching items, or in the case of an index, some or all of the attributes projected into the index.
 - ALL_ATTRIBUTES - Returns all of the item attributes from the specified table or index. If you query a local secondary index, then for each matching item in the index DynamoDB will fetch the entire item from the parent table. If the index is configured to project all item attributes, then all of the data can be obtained from the local secondary index, and no fetching is required.
 - ALL_PROJECTED_ATTRIBUTES - Allowed only when querying an index. Retrieves all attributes that have been projected into the index. If the index is configured to project all attributes, this return value is equivalent to specifying ALL_ATTRIBUTES .
 - COUNT - Returns the number of matching items, rather than the matching items themselves.
 - SPECIFIC_ATTRIBUTES - Returns only the attributes listed in *AttributesToGet* . This return value is equivalent to specifying *AttributesToGet* without specifying any value for *Select* . If you query a local secondary index and request only attributes that are projected into that index, the operation will read only the index and not the table. If any of the requested attributes are not projected into the local secondary index, DynamoDB will fetch each of these attributes from the parent table. This extra fetching incurs additional throughput cost and latency. If you query a global secondary index, you can only request attributes that are projected into the index. Global secondary index queries cannot fetch attributes from the parent table.

If neither *Select* nor *AttributesToGet* are specified, DynamoDB defaults to ALL_ATTRIBUTES when accessing a table, and ALL_PROJECTED_ATTRIBUTES when accessing an index. You cannot use both *Select* and *AttributesToGet* together in a single request, unless the value for *Select* is SPECIFIC_ATTRIBUTES . (This usage is equivalent to specifying *AttributesToGet* without any value for *Select* .)

Note: If you use the *ProjectionExpression* parameter, then the value for *Select* can only be SPECIFIC_ATTRIBUTES . Any other value for *Select* will return an error.

- **AttributesToGet** (*list*) –

Warning: This is a legacy parameter, for backward compatibility. New applications should use *ProjectionExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception. This parameter allows you to retrieve attributes of type List or Map; however, it cannot retrieve individual elements within a List or a Map.

The names of one or more attributes to retrieve. If no attribute names are provided, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

Note that *AttributesToGet* has no effect on provisioned throughput consumption. DynamoDB determines capacity units consumed based on item size, not on the amount of data that is returned to an application.

You cannot use both *AttributesToGet* and *Select* together in a *Query* request, unless the value for *Select* is `SPECIFIC_ATTRIBUTES`. (This usage is equivalent to specifying *AttributesToGet* without any value for *Select*.)

If you query a local secondary index and request only attributes that are projected into that index, the operation will read only the index and not the table. If any of the requested attributes are not projected into the local secondary index, DynamoDB will fetch each of these attributes from the parent table. This extra fetching incurs additional throughput cost and latency.

If you query a global secondary index, you can only request attributes that are projected into the index. Global secondary index queries cannot fetch attributes from the parent table.

– (*string*) –

- **ConsistentRead** (*boolean*) – Determines the read consistency model: If set to `true`, then the operation uses strongly consistent reads; otherwise, the operation uses eventually consistent reads.

Strongly consistent reads are not supported on global secondary indexes. If you query a global secondary index with *ConsistentRead* set to `true`, you will receive a *ValidationException*.

Warning: This is a legacy parameter, for backward compatibility. New applications should use *KeyConditionExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception.

- **KeyConditions** (*dict*) –

The selection criteria for the query. For a query on a table, you can have conditions only on the table primary key attributes. You must provide the hash key attribute name and value as an EQ condition. You can optionally provide a second condition, referring to the range key attribute.

Note: If you don't provide a range key condition, all of the items that match the hash key will be retrieved. If a *FilterExpression* or *QueryFilter* is present, it will be applied after the items are retrieved.

For a query on an index, you can have conditions only on the index key attributes. You must provide the index hash attribute name and value as an EQ condition. You can optionally provide a second condition, referring to the index key range attribute.

Each *KeyConditions* element consists of an attribute name to compare, along with the following:

- *AttributeValueList* - One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *ComparisonOperator* being used. For type Number, value comparisons are numeric. String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A, and a is greater than B. For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters. For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.
- *ComparisonOperator* - A comparator for evaluating attributes, for example, equals, greater than, less than, and so on. For *KeyConditions*, only the following comparison operators are supported: EQ | LE | LT | GE | GT | BEGINS_WITH | BETWEEN. The following are descriptions of these comparison operators.
 - * EQ : Equal. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one specified in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" }. Also, { "N" : "6" } does not equal { "NS" : ["6", "2", "1"] }.
 - LE : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" }. Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }.
 - LT : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" }. Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }.
 - GE : Greater than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" }. Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }.
 - GT : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" }. Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }.
 - BEGINS_WITH : Checks for a prefix. *AttributeValueList* can contain only one *AttributeValue* of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).
 - BETWEEN : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item con-

tains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not compare to { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] }

For usage examples of *AttributeValueList* and *ComparisonOperator* , see [Legacy Conditional Parameters](#) in the *Amazon DynamoDB Developer Guide* .

- (string) -
- * (dict) -

Represents the selection criteria for a *Query* or *Scan* operation:

- For a *Query* operation, *Condition* is used for specifying the *KeyConditions* to use when querying a table or an index. For *KeyConditions* , only the following comparison operators are supported: EQ | LE | LT | GE | GT | BEGINS_WITH | BETWEEN *Condition* is also used in a *QueryFilter* , which evaluates the query results and returns only the desired values.
- For a *Scan* operation, *Condition* is used in a *ScanFilter* , which evaluates the scan results and returns only the desired values.

- **AttributeValueList** (list) -

One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *ComparisonOperator* being used.

For type Number, value comparisons are numeric.

String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A , and a is greater than B . For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters .

For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.

- (dict) -

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (string) -

A String data type.

- **N** (string) -

A Number data type.

- **B** (bytes) -

A Binary data type.

- **SS** (list) -

A String Set data type.

- *(string)* –
- **NS** (*list*) –

A Number Set data type.

- *(string)* –
- **BS** (*list*) –

A Binary Set data type.

- *(bytes)* –
- **M** (*dict*) –

A Map of attribute values.

- *(string)* –
- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **ComparisonOperator** (*string*) – **[REQUIRED]**

A comparator for evaluating attributes. For example, equals, greater than, less than, etc.

The following comparison operators are available:

```
EQ | NE | LE | LT | GE | GT | NOT_NULL
| NULL | CONTAINS | NOT_CONTAINS |
BEGINS_WITH | IN | BETWEEN
```

The following are descriptions of each comparison operator.

- **EQ** : Equal. EQ is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *Attribute-Value* element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *Attribute-Value* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" }

does not equal `{"N": "6"}` . Also, `{"N": "6"}` does not equal `{"NS": ["6", "2", "1"]}` .

- **NE** : Not equal. **NE** is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}` . Also, `{"N": "6"}` does not equal `{"NS": ["6", "2", "1"]}` .
- **LE** : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}` . Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}` .
- **LT** : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}` . Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}` .
- **GE** : Greater than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}` . Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}` .
- **GT** : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}` . Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}` .
- **NOT_NULL** : The attribute exists. **NOT_NULL** is supported for all datatypes, including lists and maps. .. note::This operator tests for the existence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using **NOT_NULL** , the result is a Boolean *true* . This result is because the attribute “a ” exists; its data type is not relevant to the **NOT_NULL** comparison operator.
- **NULL** : The attribute does not exist. **NULL** is supported for all datatypes, including lists and maps. .. note::This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using **NULL** , the result is a Boolean *false* . This is because the attribute “a ” exists; its data type is not relevant to the **NULL** comparison operator.
- **CONTAINS** : Checks for a subsequence, or value in a set. *AttributeValueList* can contain only one *AttributeValue* element

of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set ("SS ", "NS ", or "BS "), then the operator evaluates to true if it finds an exact match with any member of the set. CONTAINS is supported for lists: When evaluating "a CONTAINS b", "a" can be a list; however, "b" cannot be a set, a map, or a list.

- NOT_CONTAINS : Checks for absence of a subsequence, or absence of a value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set ("SS ", "NS ", or "BS "), then the operator evaluates to true if it *does not* find an exact match with any member of the set. NOT_CONTAINS is supported for lists: When evaluating "a NOT CONTAINS b", "a" can be a list; however, "b" cannot be a set, a map, or a list.
- BEGINS_WITH : Checks for a prefix. *AttributeValueList* can contain only one *AttributeValue* of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).
- IN : Checks for matching elements within two sets. *AttributeValueList* can contain one or more *AttributeValue* elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
- BETWEEN : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not compare to { "N" : "6" } . Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }

For usage examples of *AttributeValueList* and *ComparisonOperator*, see [Legacy Conditional Parameters](#) in the *Amazon DynamoDB Developer Guide*.

- **QueryFilter** (*dict*) –

Warning: This is a legacy parameter, for backward compatibility. New applications should use *FilterExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception.

A condition that evaluates the query results after the items are read and returns only the desired values.

This parameter does not support attributes of type List or Map.

Note: A *QueryFilter* is applied after the items have already been read; the process of filtering does not consume any additional read capacity units.

If you provide more than one condition in the *QueryFilter* map, then by default all of the conditions must evaluate to true. In other words, the conditions are ANDed together. (You can use the *ConditionalOperator* parameter to OR the conditions instead. If you do this, then at least one of the conditions must evaluate to true, rather than all of them.)

Note that *QueryFilter* does not allow key attributes. You cannot define a filter condition on a hash key or range key.

Each *QueryFilter* element consists of an attribute name to compare, along with the following:

- *AttributeValueList* - One or more values to evaluate against the supplied attribute. The number of values in the list depends on the operator specified in *ComparisonOperator* . For type Number, value comparisons are numeric. String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A , and a is greater than B . For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters . For type Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values. For information on specifying data types in JSON, see [JSON Data Format](#) in the *Amazon DynamoDB Developer Guide* .
- *ComparisonOperator* - A comparator for evaluating attributes. For example, equals, greater than, less than, etc. The following comparison operators are available: EQ | NE | LE | LT | GE | GT | NOT_NULL | NULL | CONTAINS | NOT_CONTAINS | BEGINS_WITH | IN | BETWEEN For complete descriptions of all comparison operators, see the [Condition](#) data type.
- (string) –
- * (dict) –

Represents the selection criteria for a *Query* or *Scan* operation:

- For a *Query* operation, *Condition* is used for specifying the *KeyConditions* to use when querying a table or an index. For *KeyConditions* , only the following comparison operators are supported: EQ | LE | LT | GE | GT | BEGINS_WITH | BETWEEN *Condition* is also used in a *QueryFilter* , which evaluates the query results and returns only the desired values.
- For a *Scan* operation, *Condition* is used in a *ScanFilter* , which evaluates the scan results and returns only the desired values.
- **AttributeValueList** (*list*) –
One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *ComparisonOperator* being used.

For type Number, value comparisons are numeric.

String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, `a` is greater than `A`, and `a` is greater than `B`. For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters.

For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.

- **(dict)** –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S (string)** –

A String data type.

- **N (string)** –

A Number data type.

- **B (bytes)** –

A Binary data type.

- **SS (list)** –

A String Set data type.

- **(string)** –

- **NS (list)** –

A Number Set data type.

- **(string)** –

- **BS (list)** –

A Binary Set data type.

- **(bytes)** –

- **M (dict)** –

A Map of attribute values.

- **(string)** –

- **(dict)** –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L (list)** –

A List of attribute values.

- **(dict)** –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **ComparisonOperator** (*string*) – **[REQUIRED]**

A comparator for evaluating attributes. For example, equals, greater than, less than, etc.

The following comparison operators are available:

```
EQ | NE | LE | LT | GE | GT | NOT_NULL
| NULL | CONTAINS | NOT_CONTAINS |
BEGINS_WITH | IN | BETWEEN
```

The following are descriptions of each comparison operator.

- **EQ** : Equal. **EQ** is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not equal `{"NS": ["6", "2", "1"]}`.
- **NE** : Not equal. **NE** is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not equal `{"NS": ["6", "2", "1"]}`.
- **LE** : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
- **LT** : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
- **GE** : Greater than or equal. *AttributeValueList* can contain

only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .

- GT : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- NOT_NULL : The attribute exists. NOT_NULL is supported for all datatypes, including lists and maps. .. note::This operator tests for the existence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using NOT_NULL , the result is a Boolean *true* . This result is because the attribute “a ” exists; its data type is not relevant to the NOT_NULL comparison operator.
- NULL : The attribute does not exist. NULL is supported for all datatypes, including lists and maps. .. note::This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using NULL , the result is a Boolean *false* . This is because the attribute “a ” exists; its data type is not relevant to the NULL comparison operator.
- CONTAINS : Checks for a subsequence, or value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS ”, “NS ”, or “BS ”), then the operator evaluates to true if it finds an exact match with any member of the set. CONTAINS is supported for lists: When evaluating “a CONTAINS b ”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
- NOT_CONTAINS : Checks for absence of a subsequence, or absence of a value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS ”, “NS ”, or “BS ”), then the operator evaluates to true if it *does not* find an exact match with any member of the set. NOT_CONTAINS is supported for lists: When evaluating “a NOT CONTAINS b ”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
- BEGINS_WITH : Checks for a prefix. *AttributeValueList* can

contain only one *AttributeValue* of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).

- **IN** : Checks for matching elements within two sets. *AttributeValueList* can contain one or more *AttributeValue* elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
- **BETWEEN** : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not compare to { "N" : "6" } . Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }

For usage examples of *AttributeValueList* and *ComparisonOperator* , see [Legacy Conditional Parameters](#) in the *Amazon DynamoDB Developer Guide* .

- **ConditionalOperator** (*string*) –

Warning: This is a legacy parameter, for backward compatibility applications should use *FilterExpression* instead. Do not combine parameters and expression parameters in a single API call; otherwise DynamoDB will return a *ValidationException* exception.

A logical operator to apply to the conditions in a *QueryFilter* map:

- **AND** - If all of the conditions evaluate to true, then the entire map evaluates to true.
- **OR** - If at least one of the conditions evaluate to true, then the entire map evaluates to true.

If you omit *ConditionalOperator* , then AND is the default.

The operation will succeed only if the entire map evaluates to true.

Note: This parameter does not support attributes of type List or Map.

- **ScanIndexForward** (*boolean*) – Specifies the order in which to return the query results - either ascending (*true*) or descending (*false*).

Items with the same hash key are stored in sorted order by range key .If the range key data type is Number, the results are stored in numeric order. For type String, the results are returned in order of ASCII character code values. For type Binary, DynamoDB treats each byte of the binary data as unsigned.

If *ScanIndexForward* is *true* , DynamoDB returns the results in order, by range key. This is the default behavior.

If *ScanIndexForward* is *false* , DynamoDB sorts the results in descending order by range key, and then returns the results to the client.

- **ReturnConsumedCapacity** (*string*) – Determines the level of detail about provisioned throughput consumption that is returned in the response:
 - **INDEXES** - The response includes the aggregate *ConsumedCapacity* for

the operation, together with *ConsumedCapacity* for each table and secondary index that was accessed. Note that some operations, such as *GetItem* and *BatchGetItem*, do not access any indexes at all. In these cases, specifying *INDEXES* will only return *ConsumedCapacity* information for table(s).

- *TOTAL* - The response includes only the aggregate *ConsumedCapacity* for the operation.
- *NONE* - No *ConsumedCapacity* details are included in the response.
- **ProjectionExpression** (*string*) – A string that identifies one or more attributes to retrieve from the table. These attributes can include scalars, sets, or elements of a JSON document. The attributes in the expression must be separated by commas.

If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

For more information, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

Note: *ProjectionExpression* replaces the legacy *AttributesToGet* parameter.

- **FilterExpression** (*string*) – A string that contains conditions that DynamoDB applies after the *Query* operation, but before the data is returned to you. Items that do not satisfy the *FilterExpression* criteria are not returned.

Note: A *FilterExpression* is applied after the items have already been read; the process of filtering does not consume any additional read capacity units.

For more information, see [Filter Expressions](#) in the *Amazon DynamoDB Developer Guide*.

Note: *FilterExpression* replaces the legacy *QueryFilter* and *ConditionalOperator* parameters.

- **KeyConditionExpression** (*string*) – The condition that specifies the key value(s) for items to be retrieved by the *Query* action.

The condition must perform an equality test on a single hash key value. The condition can also perform one of several comparison tests on a single range key value. *Query* can use *KeyConditionExpression* to retrieve one item with a given hash and range key value, or several items that have the same hash key value but different range key values.

The hash key equality test is required, and must be specified in the following format:

```
hashAttributeName = :hashval
```

If you also want to provide a range key condition, it must be combined using *AND* with the hash key condition. Following is an example, using the *=* comparison operator for the range key:

```
hashAttributeName      =      :hashval      AND
rangeAttributeName = :rangeval
```

Valid comparisons for the range key condition are as follows:

- `rangeAttributeName = :rangeval` - true if the range key is equal to `:rangeval`.
- `rangeAttributeName ** :rangeval` - true if the range key is less than `:rangeval`.
- `rangeAttributeName = :rangeval` - true if the range key is less

- than or equal to :rangeval.
- rangeAttributeName ** :rangeval - true if the range key is greater than :rangeval.
- rangeAttributeName = :rangeval - true if the range key is greater than or equal to :rangeval.
- rangeAttributeName BETWEEN :rangeval1 AND :rangeval2 - true if the range key is greater than or equal to :rangeval1, and less than or equal to :rangeval2.
- begins_with (rangeAttributeName , :rangeval) - true if the range key begins with a particular operand. (You cannot use this function with a range key that is of type Number.) Note that the function name begins_with is case-sensitive.

Use the *ExpressionAttributeValues* parameter to replace tokens such as :hashval and :rangeval with actual values at runtime.

You can optionally use the *ExpressionAttributeNames* parameter to replace the names of the hash and range attributes with placeholder tokens. This option might be necessary if an attribute name conflicts with a DynamoDB reserved word. For example, the following *KeyConditionExpression* parameter causes an error because *Size* is a reserved word:

- Size = :myval

To work around this, define a placeholder (such as #S) to represent the attribute name *Size* . *KeyConditionExpression* then is as follows:

- #S = :myval

For a list of reserved words, see [Reserved Words](#) in the *Amazon DynamoDB Developer Guide* .

For more information on *ExpressionAttributeNames* and *ExpressionAttributeValues* , see [Using Placeholders for Attribute Names and Values](#) in the *Amazon DynamoDB Developer Guide* .

Note: *KeyConditionExpression* replaces the legacy *KeyConditions* parameter.

- **ExpressionAttributeNames** (*dict*) – One or more substitution tokens for attribute names in an expression. The following are some use cases for using *ExpressionAttributeNames* :
 - To access an attribute whose name conflicts with a DynamoDB reserved word.
 - To create a placeholder for repeating occurrences of an attribute name in an expression.
 - To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the # character in an expression to dereference an attribute name. For example, consider the following attribute name:

- Percentile

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see [Reserved Words](#) in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for *ExpressionAttributeNames* :

- {"#P": "Percentile"}

You could then use this substitution in an expression, as in this example:

- #P = :val

Note: Tokens that begin with the : character are *expression attribute values* , which are placeholders for the actual value at runtime.

For more information on expression attribute names, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

- *(string)* –
- * *(string)* –

- **ExpressionAttributeValues** (*dict*) – One or more values that can be substituted in an expression.

Use the **:** (colon) character in an expression to dereference an attribute value. For example, suppose that you wanted to check whether the value of the *ProductStatus* attribute was one of the following:

```
Available | Backordered | Discontinued
```

You would first need to specify *ExpressionAttributeValues* as follows:

```
{ ":avail":{"S":"Available"}, ":back":{"S":"Backordered"},
  ":disc":{"S":"Discontinued"} }
```

You could then use these values in an expression, such as this:

```
ProductStatus IN (:avail, :back, :disc)
```

For more information on expression attribute values, see [Specifying Conditions](#) in the *Amazon DynamoDB Developer Guide*.

- *(string)* –
- * *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –
A String data type.
- **N** (*string*) –
A Number data type.
- **B** (*bytes*) –
A Binary data type.
- **SS** (*list*) –
A String Set data type.
- (*string*) –
- **NS** (*list*) –
A Number Set data type.
- (*string*) –
- **BS** (*list*) –
A Binary Set data type.
- (*bytes*) –
- **M** (*dict*) –
A Map of attribute values.
- (*string*) –
- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Items': [
    {
      'string': {
        'S': 'string',
        'N': 'string',
        'B': b'bytes',
        'SS': [
          'string',
        ],
        'NS': [
          'string',
        ],
        'BS': [
```

```

        b'bytes',
    ],
    'M': {
        'string': {'... recursive ...'}
    },
    'L': [
        {'... recursive ...'},
    ],
    'NULL': True|False,
    'BOOL': True|False
    }
},
],
'Count': 123,
'ScannedCount': 123,
'ConsumedCapacity': {
    'TableName': 'string',
    'CapacityUnits': 123.0,
    'Table': {
        'CapacityUnits': 123.0
    },
    'LocalSecondaryIndexes': {
        'string': {
            'CapacityUnits': 123.0
        }
    },
    'GlobalSecondaryIndexes': {
        'string': {
            'CapacityUnits': 123.0
        }
    }
},
'NextToken': 'string'
}

```

Response Structure

- (*dict*) –

Represents the output of a *Query* operation.

- **Items** (*list*) –

An array of item attributes that match the query criteria. Each element in this array consists of an attribute name and the value for that attribute.

- * (*dict*) –

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –
- **NS** (*list*) –

A Number Set data type.

- (*string*) –
- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –
- **M** (*dict*) –

A Map of attribute values.

- (*string*) –
- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **Count** (*integer*) –

The number of items in the response.

If you used a *QueryFilter* in the request, then *Count* is the number of items returned after the filter was applied, and *ScannedCount* is the number of matching items before the filter was applied.

If you did not use a filter in the request, then *Count* and *ScannedCount* are the same.

- **ScannedCount** (*integer*) –

The number of items evaluated, before any *QueryFilter* is applied. A high *ScannedCount* value with few, or no, *Count* results indicates an inefficient *Query* operation. For more information, see [Count and ScannedCount](#) in the *Amazon DynamoDB Developer Guide*.

If you did not use a filter in the request, then *ScannedCount* is the same as *Count*.

– **ConsumedCapacity** (*dict*) –

The capacity units consumed by an operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. *ConsumedCapacity* is only returned if the request asked for it. For more information, see [Provisioned Throughput](#) in the *Amazon DynamoDB Developer Guide*.

* **TableName** (*string*) –

The name of the table that was affected by the operation.

* **CapacityUnits** (*float*) –

The total number of capacity units consumed by the operation.

* **Table** (*dict*) –

The amount of throughput consumed on the table affected by the operation.

· **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

* **LocalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each local index affected by the operation.

· (*string*) –

· (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

· **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

* **GlobalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each global index affected by the operation.

· (*string*) –

· (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

· **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

– **NextToken** (*string*) –

A token to resume pagination.

```
class DynamoDB.Paginator.Scan
```

```
paginator = client.get_paginator('scan')
```

paginate (**kwargs)

Creates an iterator that will paginate through responses from `DynamoDB.Client.scan()`.

Request Syntax

```
response_iterator = paginator.paginate(
    TableName='string',
    IndexName='string',
    AttributesToGet=[
        'string',
    ],
    Select='ALL_ATTRIBUTES'|'ALL_PROJECTED_ATTRIBUTES'|'SPECIFIC_ATTRIBUTES'|'COUNT',
    ScanFilter={
        'string': {
            'AttributeValueList': [
                {
                    'S': 'string',
                    'N': 'string',
                    'B': b'bytes',
                    'SS': [
                        'string',
                    ],
                    'NS': [
                        'string',
                    ],
                    'BS': [
                        b'bytes',
                    ],
                    'M': {
                        'string': {'... recursive ...'}
                    },
                    'L': [
                        {'... recursive ...'},
                    ],
                    'NULL': True|False,
                    'BOOL': True|False
                },
            ],
            'ComparisonOperator': 'EQ'|'NE'|'IN'|'LE'|'LT'|'GE'|'GT'|'BETWEEN'|'NOT_NULL'|'...'
        },
    ],
    ConditionalOperator='AND'|'OR',
    ReturnConsumedCapacity='INDEXES'|'TOTAL'|'NONE',
    TotalSegments=123,
    Segment=123,
    ProjectionExpression='string',
    FilterExpression='string',
    ExpressionAttributeNames={
        'string': 'string'
    },
    ExpressionAttributeValues={
        'string': {
            'S': 'string',
            'N': 'string',
            'B': b'bytes',
            'SS': [
```

```

        'string',
    ],
    'NS': [
        'string',
    ],
    'BS': [
        b'bytes',
    ],
    'M': {
        'string': {'... recursive ...'}
    },
    'L': [
        {'... recursive ...'},
    ],
    'NULL': True|False,
    'BOOL': True|False
}
},
ConsistentRead=True|False,
PaginationConfig={
    'MaxItems': 123,
    'PageSize': 123,
    'StartingToken': 'string'
}
)

```

Parameters

- **TableName** (*string*) – [REQUIRED]

The name of the table containing the requested items; or, if you provide `IndexName`, the name of the table to which that index belongs.

- **IndexName** (*string*) – The name of a secondary index to scan. This index can be any local secondary index or global secondary index. Note that if you use the `IndexName` parameter, you must also provide `TableName`.

Warning: This is a legacy parameter, for backward compatibility. New applications should use *ProjectionExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception. This parameter allows you to retrieve attributes of type List or Map; however, it cannot retrieve individual elements within a List or a Map.

- **AttributesToGet** (*list*) –

The names of one or more attributes to retrieve. If no attribute names are provided, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

Note that *AttributesToGet* has no effect on provisioned throughput consumption. DynamoDB determines capacity units consumed based on item size, not on the amount of data that is returned to an application.

- (*string*) –
- **Select** (*string*) – The attributes to be returned in the result. You can retrieve all item attributes, specific item attributes, or the count of matching items.
 - ALL_ATTRIBUTES - Returns all of the item attributes.
 - COUNT - Returns the number of matching items, rather than the matching items themselves.
 - SPECIFIC_ATTRIBUTES - Returns only the attributes listed in *AttributesToGet*. This return value is equivalent to specifying *AttributesToGet*.

Get without specifying any value for *Select* .

If neither *Select* nor *AttributesToGet* are specified, DynamoDB defaults to `ALL_ATTRIBUTES` . You cannot use both *AttributesToGet* and *Select* together in a single request, unless the value for *Select* is `SPECIFIC_ATTRIBUTES` . (This usage is equivalent to specifying *AttributesToGet* without any value for *Select* .)

- **ScanFilter** (*dict*) –

Warning: This is a legacy parameter, for backward compatibility. New applications should use *FilterExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception.

A condition that evaluates the scan results and returns only the desired values.

Note: This parameter does not support attributes of type List or Map.

If you specify more than one condition in the *ScanFilter* map, then by default all of the conditions must evaluate to true. In other words, the conditions are ANDed together. (You can use the *ConditionalOperator* parameter to OR the conditions instead. If you do this, then at least one of the conditions must evaluate to true, rather than all of them.)

Each *ScanFilter* element consists of an attribute name to compare, along with the following:

- *AttributeValueList* - One or more values to evaluate against the supplied attribute. The number of values in the list depends on the operator specified in *ComparisonOperator* . For type Number, value comparisons are numeric. String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, `a` is greater than `A` , and `a` is greater than `B` . For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters . For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values. For information on specifying data types in JSON, see [JSON Data Format](#) in the *Amazon DynamoDB Developer Guide* .
- *ComparisonOperator* - A comparator for evaluating attributes. For example, equals, greater than, less than, etc. The following comparison operators are available: `EQ` | `NE` | `LE` | `LT` | `GE` | `GT` | `NOT_NULL` | `NULL` | `CONTAINS` | `NOT_CONTAINS` | `BEGINS_WITH` | `IN` | `BETWEEN` For complete descriptions of all comparison operators, see [Condition](#) .
- (*string*) –
- * (*dict*) –

Represents the selection criteria for a *Query* or *Scan* operation:

- For a *Query* operation, *Condition* is used for specifying the *KeyConditions* to use when querying a table or an index. For *KeyConditions* , only the following comparison operators are supported: `EQ` | `LE` | `LT` | `GE` | `GT` | `BEGINS_WITH` | `BETWEEN` *Condition* is also used in a *QueryFilter* , which evaluates the query results and returns only the desired values.
- For a *Scan* operation, *Condition* is used in a *ScanFilter* , which evaluates the scan results and returns only the desired values.

- **AttributeValueList** (*list*) –

One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *Comparison-Operator* being used.

For type Number, value comparisons are numeric.

String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A , and a is greater than B . For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters

·

For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –

- **M** (*dict*) –

A Map of attribute values.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one

title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **ComparisonOperator** (*string*) – **[REQUIRED]**

A comparator for evaluating attributes. For example, equals, greater than, less than, etc.

The following comparison operators are available:

```
EQ | NE | LE | LT | GE | GT | NOT_NULL
| NULL | CONTAINS | NOT_CONTAINS |
BEGINS_WITH | IN | BETWEEN
```

The following are descriptions of each comparison operator.

- **EQ** : Equal. **EQ** is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] } .
- **NE** : Not equal. **NE** is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] } .
- **LE** : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- **LT** : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set

type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.

- GE : Greater than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
- GT : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
- NOT_NULL : The attribute exists. NOT_NULL is supported for all datatypes, including lists and maps. .. note::This operator tests for the existence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using NOT_NULL , the result is a Boolean *true* . This result is because the attribute “a ” exists; its data type is not relevant to the NOT_NULL comparison operator.
- NULL : The attribute does not exist. NULL is supported for all datatypes, including lists and maps. .. note::This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using NULL , the result is a Boolean *false* . This is because the attribute “a ” exists; its data type is not relevant to the NULL comparison operator.
- CONTAINS : Checks for a subsequence, or value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS”, “NS”, or “BS”), then the operator evaluates to true if it finds an exact match with any member of the set. CONTAINS is supported for lists: When evaluating “a CONTAINS b”, “a” can be a list; however, “b” cannot be a set, a map, or a list.
- NOT_CONTAINS : Checks for absence of a subsequence, or absence of a value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS”, “NS”, or “BS”),

then the operator evaluates to true if it *does not* find an exact match with any member of the set. `NOT_CONTAINS` is supported for lists: When evaluating “a NOT CONTAINS b”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.

- `BEGINS_WITH` : Checks for a prefix. *AttributeValueList* can contain only one *AttributeValue* of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).
- `IN` : Checks for matching elements within two sets. *AttributeValueList* can contain one or more *AttributeValue* elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
- `BETWEEN` : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not compare to { "N" : "6" } . Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }

For usage examples of *AttributeValueList* and *ComparisonOperator* , see [Legacy Conditional Parameters](#) in the *Amazon DynamoDB Developer Guide* .

- **ConditionalOperator** (*string*) –

Warning: This is a legacy parameter, for backward compatibility applications should use *FilterExpression* instead. Do not combine parameters and expression parameters in a single API call; otherwise DynamoDB will return a *ValidationException* exception.

A logical operator to apply to the conditions in a *ScanFilter* map:

- `AND` - If all of the conditions evaluate to true, then the entire map evaluates to true.
- `OR` - If at least one of the conditions evaluate to true, then the entire map evaluates to true.

If you omit *ConditionalOperator* , then `AND` is the default.

The operation will succeed only if the entire map evaluates to true.

Note: This parameter does not support attributes of type List or Map.

- **ReturnConsumedCapacity** (*string*) – Determines the level of detail about provisioned throughput consumption that is returned in the response:
 - `INDEXES` - The response includes the aggregate *ConsumedCapacity* for the operation, together with *ConsumedCapacity* for each table and secondary index that was accessed. Note that some operations, such as *GetItem* and *BatchGetItem* , do not access any indexes at all. In these cases, specifying `INDEXES` will only return *ConsumedCapacity* information for table(s).

- *TOTAL* - The response includes only the aggregate *ConsumedCapacity* for the operation.
- *NONE* - No *ConsumedCapacity* details are included in the response.
- **TotalSegments** (*integer*) – For a parallel *Scan* request, *TotalSegments* represents the total number of segments into which the *Scan* operation will be divided. The value of *TotalSegments* corresponds to the number of application workers that will perform the parallel scan. For example, if you want to use four application threads to scan a table or an index, specify a *TotalSegments* value of 4.

The value for *TotalSegments* must be greater than or equal to 1, and less than or equal to 1000000. If you specify a *TotalSegments* value of 1, the *Scan* operation will be sequential rather than parallel.

If you specify *TotalSegments*, you must also specify *Segment*.

- **Segment** (*integer*) – For a parallel *Scan* request, *Segment* identifies an individual segment to be scanned by an application worker.

Segment IDs are zero-based, so the first segment is always 0. For example, if you want to use four application threads to scan a table or an index, then the first thread specifies a *Segment* value of 0, the second thread specifies 1, and so on.

The value of *LastEvaluatedKey* returned from a parallel *Scan* request must be used as *ExclusiveStartKey* with the same segment ID in a subsequent *Scan* operation.

The value for *Segment* must be greater than or equal to 0, and less than the value provided for *TotalSegments*.

If you provide *Segment*, you must also provide *TotalSegments*.

- **ProjectionExpression** (*string*) – A string that identifies one or more attributes to retrieve from the specified table or index. These attributes can include scalars, sets, or elements of a JSON document. The attributes in the expression must be separated by commas.

If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

For more information, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

Note: *ProjectionExpression* replaces the legacy *AttributesToGet* parameter.

- **FilterExpression** (*string*) – A string that contains conditions that DynamoDB applies after the *Scan* operation, but before the data is returned to you. Items that do not satisfy the *FilterExpression* criteria are not returned.

Note: A *FilterExpression* is applied after the items have already been read; the process of filtering does not consume any additional read capacity units.

For more information, see [Filter Expressions](#) in the *Amazon DynamoDB Developer Guide*.

Note: *FilterExpression* replaces the legacy *ScanFilter* and *ConditionalOperator* parameters.

- **ExpressionAttributeNames** (*dict*) – One or more substitution tokens for attribute names in an expression. The following are some use cases for using *ExpressionAttributeNames*:

- To access an attribute whose name conflicts with a DynamoDB reserved word.
- To create a placeholder for repeating occurrences of an attribute name in an expression.
- To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the `#` character in an expression to dereference an attribute name. For example, consider the following attribute name:

- Percentile

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see [Reserved Words](#) in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for *ExpressionAttributeNames* :

- { "#P": "Percentile" }

You could then use this substitution in an expression, as in this example:

- #P = :val

Note: Tokens that begin with the `:` character are *expression attribute values* , which are placeholders for the actual value at runtime.

For more information on expression attribute names, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide* .

- (string) –
- * (string) –

- **ExpressionAttributeValues** (*dict*) – One or more values that can be substituted in an expression.

Use the `:` (colon) character in an expression to dereference an attribute value. For example, suppose that you wanted to check whether the value of the *ProductStatus* attribute was one of the following:

```
Available | Backordered | Discontinued
```

You would first need to specify *ExpressionAttributeValues* as follows:

```
{ ":avail":{"S":"Available"}, ":back":{"S":"Backordered"},  
  ":disc":{"S":"Discontinued"} }
```

You could then use these values in an expression, such as this:

```
ProductStatus IN (:avail, :back, :disc)
```

For more information on expression attribute values, see [Specifying Conditions](#) in the *Amazon DynamoDB Developer Guide* .

- (string) –
- * (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –
A String data type.
- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –
- **NS** (*list*) –

A Number Set data type.

- (*string*) –
- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –
- **M** (*dict*) –

A Map of attribute values.

- (*string*) –
- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **ConsistentRead** (*boolean*) – A Boolean value that determines the read consistency model during the scan:

- If *ConsistentRead* is `false`, then *Scan* will use eventually consistent reads. The data returned from *Scan* might not contain the results of other recently completed write operations (*PutItem*, *UpdateItem* or *DeleteItem*). The *Scan* response might include some stale data.
- If *ConsistentRead* is `true`, then *Scan* will use strongly consistent reads. All of the write operations that completed before the *Scan* began are guaranteed to be contained in the *Scan* response.

The default setting for *ConsistentRead* is `false`, meaning that eventually consistent reads will be used.

Strongly consistent reads are not supported on global secondary indexes. If you scan a global secondary index with *ConsistentRead* set to true, you will receive a *ValidationException*.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Items': [
    {
      'string': {
        'S': 'string',
        'N': 'string',
        'B': b'bytes',
        'SS': [
          'string',
        ],
        'NS': [
          'string',
        ],
        'BS': [
          b'bytes',
        ],
        'M': {
          'string': {'... recursive ...'}
        },
        'L': [
          {'... recursive ...'},
        ],
        'NULL': True|False,
        'BOOL': True|False
      }
    },
  ],
  'Count': 123,
  'ScannedCount': 123,
  'ConsumedCapacity': {
    'TableName': 'string',
    'CapacityUnits': 123.0,
    'Table': {
      'CapacityUnits': 123.0
    },
  },
  'LocalSecondaryIndexes': {
    'string': {
```

```

        'CapacityUnits': 123.0
    },
    'GlobalSecondaryIndexes': {
        'string': {
            'CapacityUnits': 123.0
        }
    }
},
'NextToken': 'string'
}

```

Response Structure

- *(dict)* –

Represents the output of a *Scan* operation.

- **Items** (*list*) –

An array of item attributes that match the scan criteria. Each element in this array consists of an attribute name and the value for that attribute.

- * *(dict)* –

- *(string)* –

- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- *(string)* –

- **NS** (*list*) –

A Number Set data type.

- *(string)* –

- **BS** (*list*) –

A Binary Set data type.

- *(bytes)* –

- **M** (*dict*) –

A Map of attribute values.

- *(string)* –

- *(dict)* –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List of attribute values.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **Count** (*integer*) –

The number of items in the response.

If you set *ScanFilter* in the request, then *Count* is the number of items returned after the filter was applied, and *ScannedCount* is the number of matching items before the filter was applied.

If you did not use a filter in the request, then *Count* is the same as *ScannedCount*.

- **ScannedCount** (*integer*) –

The number of items evaluated, before any *ScanFilter* is applied. A high *ScannedCount* value with few, or no, *Count* results indicates an inefficient *Scan* operation. For more information, see [Count and ScannedCount](#) in the *Amazon DynamoDB Developer Guide*.

If you did not use a filter in the request, then *ScannedCount* is the same as *Count*.

- **ConsumedCapacity** (*dict*) –

The capacity units consumed by an operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. *ConsumedCapacity* is only returned if the request asked for it. For more information, see [Provisioned Throughput](#) in the *Amazon DynamoDB Developer Guide*.

- * **TableName** (*string*) –

The name of the table that was affected by the operation.

- * **CapacityUnits** (*float*) –

The total number of capacity units consumed by the operation.

- * **Table** (*dict*) –

The amount of throughput consumed on the table affected by the operation.

- **CapacityUnits** (*float*) –
The total number of capacity units consumed on a table or an index.
- * **LocalSecondaryIndexes** (*dict*) –
The amount of throughput consumed on each local index affected by the operation.
 - (*string*) –
 - (*dict*) –
Represents the amount of provisioned throughput capacity consumed on a table or an index.
 - **CapacityUnits** (*float*) –
The total number of capacity units consumed on a table or an index.
- * **GlobalSecondaryIndexes** (*dict*) –
The amount of throughput consumed on each global index affected by the operation.
 - (*string*) –
 - (*dict*) –
Represents the amount of provisioned throughput capacity consumed on a table or an index.
 - **CapacityUnits** (*float*) –
The total number of capacity units consumed on a table or an index.
- **NextToken** (*string*) –
A token to resume pagination.

Waiters

The available waiters are:

- `DynamoDB.Waiter.TableExists`
- `DynamoDB.Waiter.TableNotExists`

class `DynamoDB.Waiter.TableExists`

```
waiter = client.get_waiter('table_exists')
```

wait (***kwargs*)

Polls `DynamoDB.Client.describe_table()` every 20 seconds until a successful state is reached. An error is returned after 25 failed checks.

Request Syntax

```
waiter.wait(
    TableName='string'
)
```

Parameters **TableName** (*string*) – [REQUIRED]

The name of the table to describe.

Returns None

class `DynamoDB.Waiter.TableNotExists`

```
waiter = client.get_waiter('table_not_exists')
```

wait (***kwargs*)

Polls `DynamoDB.Client.describe_table()` every 20 seconds until a successful state is reached. An error is returned after 25 failed checks.

Request Syntax

```
waiter.wait(  
    TableName='string'  
)
```

Parameters `TableName` (*string*) – [REQUIRED]

The name of the table to describe.

Returns None

Service Resource

class `DynamoDB.ServiceResource`

A resource representing Amazon DynamoDB:

```
import boto3
```

```
dynamodb = boto3.resource('dynamodb')
```

These are the resource’s available actions:

- `batch_get_item()`
- `batch_write_item()`
- `create_table()`

These are the resource’s available sub-resources:

- `Table()`

These are the resource’s available collections:

- `tables`

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

batch_get_item (***kwargs*)

The *BatchGetItem* operation returns the attributes of one or more items from one or more tables. You identify requested items by primary key.

A single operation can retrieve up to 16 MB of data, which can contain as many as 100 items. *BatchGetItem* will return a partial result if the response size limit is exceeded, the table’s provisioned throughput is exceeded, or an internal processing failure occurs. If a partial result is returned, the operation returns a value for *UnprocessedKeys*. You can use this value to retry the operation starting with the next item to get.

Warning: If you request more than 100 items *BatchGetItem* will return a *ValidationException* with the message “Too many items requested for the BatchGetItem call”.

For example, if you ask to retrieve 100 items, but each individual item is 300 KB in size, the system returns 52 items (so as not to exceed the 16 MB limit). It also returns an appropriate *UnprocessedKeys* value so you can get the next page of results. If desired, your application can include its own logic to assemble the pages of results into one data set.

If *none* of the items can be processed due to insufficient provisioned throughput on all of the tables in the request, then *BatchGetItem* will return a *ProvisionedThroughputExceededException* . If *at least one* of the items is successfully processed, then *BatchGetItem* completes successfully, while returning the keys of the unread items in *UnprocessedKeys* .

Warning: If DynamoDB returns any unprocessed items, you should retry the batch operation on those items. However, *we strongly recommend that you use an exponential backoff algorithm* . If you retry the batch operation immediately, the underlying read or write requests can still fail due to throttling on the individual tables. If you delay the batch operation using exponential backoff, the individual requests in the batch are much more likely to succeed.

For more information, see [Batch Operations and Error Handling](#) in the *Amazon DynamoDB Developer Guide* .

By default, *BatchGetItem* performs eventually consistent reads on every table in the request. If you want strongly consistent reads instead, you can set *ConsistentRead* to `true` for any or all tables.

In order to minimize response latency, *BatchGetItem* retrieves items in parallel.

When designing your application, keep in mind that DynamoDB does not return attributes in any particular order. To help parse the response by item, include the primary key values for the items in your request in the *AttributesToGet* parameter.

If a requested item does not exist, it is not returned in the result. Requests for nonexistent items consume the minimum read capacity units according to the type of read. For more information, see [Capacity Units Calculations](#) in the *Amazon DynamoDB Developer Guide* .

Request Syntax

```
response = dynamodb.batch_get_item(
    RequestItems={
        'string': {
            'Keys': [
                {
                    'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])|set([
                ],
            ],
            'AttributesToGet': [
                'string',
            ],
            'ConsistentRead': True|False,
            'ProjectionExpression': 'string',
            'ExpressionAttributeNames': {
                'string': 'string'
            }
        },
    },
    ReturnConsumedCapacity='INDEXES'|'TOTAL'|'NONE'
)
```

Parameters

- **RequestItems** (*dict*) – [REQUIRED]

A map of one or more table names and, for each table, a map that describes one or more items to retrieve from that table. Each table name can be used only once

per *BatchGetItem* request.

Each element in the map of items to retrieve consists of the following:

- *ConsistentRead* - If `true`, a strongly consistent read is used; if `false` (the default), an eventually consistent read is used.
- *ExpressionAttributeNames* - One or more substitution tokens for attribute names in the *ProjectionExpression* parameter. The following are some use cases for using *ExpressionAttributeNames*:
 - * To access an attribute whose name conflicts with a DynamoDB reserved word.
 - To create a placeholder for repeating occurrences of an attribute name in an expression.
 - To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the `#` character in an expression to dereference an attribute name. For example, consider the following attribute name:

- `Percentile`

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see [Reserved Words](#) in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for *ExpressionAttributeNames*:

- `{"#P": "Percentile"}`

You could then use this substitution in an expression, as in this example:

- `#P = :val`

Note: Tokens that begin with the `:` character are *expression attribute values*, which are placeholders for the actual value at runtime.

For more information on expression attribute names, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

- *Keys* - An array of primary key attribute values that define specific items in the table. For each primary key, you must provide *all* of the key attributes. For example, with a hash type primary key, you only need to provide the hash attribute. For a hash-and-range type primary key, you must provide *both* the hash attribute and the range attribute.
- *ProjectionExpression* - A string that identifies one or more attributes to retrieve from the table. These attributes can include scalars, sets, or elements of a JSON document. The attributes in the expression must be separated by commas. If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result. For more information, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.
- *AttributesToGet* - .. warning:: This is a legacy parameter, for backward compatibility. New applications should use *ProjectionExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception. This parameter allows you to retrieve attributes of type List or Map; however, it cannot retrieve individual elements within a List or a Map. The names of one or more attributes to retrieve. If no attribute names are provided, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result. Note that *AttributesToGet* has no effect on provisioned throughput consumption. DynamoDB determines capacity units consumed based on item size, not on the amount of data that is returned to an application.
- *(string)* –

* (*dict*) –

Represents a set of primary keys and, for each key, the attributes to retrieve from the table.

For each primary key, you must provide *all* of the key attributes. For example, with a hash type primary key, you only need to provide the hash attribute. For a hash-and-range type primary key, you must provide *both* the hash attribute and the range attribute.

- **Keys** (*list*) – **[REQUIRED]**

The primary key attribute values that define the items and the attributes associated with the items.

- (*dict*) –

- (*string*) –

- (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).

- **AttributesToGet** (*list*) –

One or more attributes to retrieve from the table or index. If no attribute names are specified then all attributes will be returned. If any of the specified attributes are not found, they will not appear in the result.

- (*string*) –

- **ConsistentRead** (*boolean*) –

The consistency of a read operation. If set to `true`, then a strongly consistent read is used; otherwise, an eventually consistent read is used.

- **ProjectionExpression** (*string*) –

A string that identifies one or more attributes to retrieve from the table. These attributes can include scalars, sets, or elements of a JSON document. The attributes in the *ProjectionExpression* must be separated by commas.

If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

For more information, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

Note: *ProjectionExpression* replaces the legacy *AttributesToGet* parameter.

- **ExpressionAttributeNames** (*dict*) –

One or more substitution tokens for attribute names in an expression. The following are some use cases for using *ExpressionAttributeNames*:

- To access an attribute whose name conflicts with a DynamoDB reserved word.
- To create a placeholder for repeating occurrences of an attribute name in an expression.
- To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the # character in an expression to dereference an attribute name. For example, consider the following attribute name:

- Percentile

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see [Reserved Words](#) in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for *ExpressionAttributeNames* :

- { "#P": "Percentile" }

You could then use this substitution in an expression, as in this example:

- #P = :val

Note: Tokens that begin with the : character are *expression attribute values* , which are placeholders for the actual value at runtime.

For more information on expression attribute names, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide* .

- (string) –

- (string) –

- **ReturnConsumedCapacity** (string) – Determines the level of detail about provisioned throughput consumption that is returned in the response:
 - *INDEXES* - The response includes the aggregate *ConsumedCapacity* for the operation, together with *ConsumedCapacity* for each table and secondary index that was accessed. Note that some operations, such as *GetItem* and *BatchGetItem* , do not access any indexes at all. In these cases, specifying *INDEXES* will only return *ConsumedCapacity* information for table(s).
 - *TOTAL* - The response includes only the aggregate *ConsumedCapacity* for the operation.
 - *NONE* - No *ConsumedCapacity* details are included in the response.

Return type dict

Returns

Response Syntax

```
{
  'Responses': {
    'string': [
      {
        'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])
      },
    ]
  },
  'UnprocessedKeys': {
    'string': {
      'Keys': [
        {
          'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])
        },
      ],
      'AttributesToGet': [
        'string',
      ],
    },
  },
}
```

```

    ],
    'ConsistentRead': True|False,
    'ProjectionExpression': 'string',
    'ExpressionAttributeNames': {
        'string': 'string'
    }
}
},
'ConsumedCapacity': [
    {
        'TableName': 'string',
        'CapacityUnits': 123.0,
        'Table': {
            'CapacityUnits': 123.0
        },
        'LocalSecondaryIndexes': {
            'string': {
                'CapacityUnits': 123.0
            }
        },
        'GlobalSecondaryIndexes': {
            'string': {
                'CapacityUnits': 123.0
            }
        }
    },
]
}

```

Response Structure

- (*dict*) –

Represents the output of a *BatchGetItem* operation.

- **Responses** (*dict*) –

A map of table name to a list of items. Each object in *Responses* consists of a table name, along with a map of attribute data consisting of the data type and attribute value.

- * (*string*) –

- (*list*) –
- (*dict*) –
- (*string*) –
- (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).

- **UnprocessedKeys** (*dict*) –

A map of tables and their respective keys that were not processed with the current response. The *UnprocessedKeys* value is in the same form as *RequestItems*, so the value can be provided directly to a subsequent *BatchGetItem* operation. For more information, see *RequestItems* in the Request Parameters section.

Each element consists of:

- * *Keys* - An array of primary key attribute values that define specific items in the table.
- * *AttributesToGet* - One or more attributes to be retrieved from the table or index. By default, all attributes are returned. If a requested

attribute is not found, it does not appear in the result.

- * *ConsistentRead* - The consistency of a read operation. If set to `true`, then a strongly consistent read is used; otherwise, an eventually consistent read is used.

If there are no unprocessed keys remaining, the response contains an empty *UnprocessedKeys* map.

- * (*string*) –

- (*dict*) –

Represents a set of primary keys and, for each key, the attributes to retrieve from the table.

For each primary key, you must provide *all* of the key attributes. For example, with a hash type primary key, you only need to provide the hash attribute. For a hash-and-range type primary key, you must provide *both* the hash attribute and the range attribute.

- **Keys** (*list*) –

The primary key attribute values that define the items and the attributes associated with the items.

- (*dict*) –

- (*string*) –

- (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).

- **AttributesToGet** (*list*) –

One or more attributes to retrieve from the table or index. If no attribute names are specified then all attributes will be returned. If any of the specified attributes are not found, they will not appear in the result.

- (*string*) –

- **ConsistentRead** (*boolean*) –

The consistency of a read operation. If set to `true`, then a strongly consistent read is used; otherwise, an eventually consistent read is used.

- **ProjectionExpression** (*string*) –

A string that identifies one or more attributes to retrieve from the table. These attributes can include scalars, sets, or elements of a JSON document. The attributes in the *ProjectionExpression* must be separated by commas.

If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

For more information, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

Note: *ProjectionExpression* replaces the legacy *AttributesToGet* parameter.

- **ExpressionAttributeNames** (*dict*) –

One or more substitution tokens for attribute names in an expression. The following are some use cases for using *Expres-*

sionAttributeNames :

- To access an attribute whose name conflicts with a DynamoDB reserved word.
- To create a placeholder for repeating occurrences of an attribute name in an expression.
- To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the # character in an expression to dereference an attribute name. For example, consider the following attribute name:

- Percentile

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see [Reserved Words](#) in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for *ExpressionAttributeNames* :

- { "#P": "Percentile" }

You could then use this substitution in an expression, as in this example:

- #P = :val

Note: Tokens that begin with the : character are *expression attribute values* , which are placeholders for the actual value at runtime.

For more information on expression attribute names, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide* .

- (string) –
- (string) –

– ConsumedCapacity (list) –

The read capacity units consumed by the operation.

Each element consists of:

- * *TableName* - The table that consumed the provisioned throughput.
- * *CapacityUnits* - The total number of capacity units consumed.
- * (dict) –

The capacity units consumed by an operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. *ConsumedCapacity* is only returned if the request asked for it. For more information, see [Provisioned Throughput](#) in the *Amazon DynamoDB Developer Guide* .

- **TableName** (string) –

The name of the table that was affected by the operation.

- **CapacityUnits** (float) –

The total number of capacity units consumed by the operation.

- **Table** (dict) –

The amount of throughput consumed on the table affected by the operation.

- **CapacityUnits** (float) –

The total number of capacity units consumed on a table or an index.

- **LocalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each local index affected by the operation.

- (*string*) –
- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

- **GlobalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each global index affected by the operation.

- (*string*) –
- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

batch_write_item (***kwargs*)

The *BatchWriteItem* operation puts or deletes multiple items in one or more tables. A single call to *BatchWriteItem* can write up to 16 MB of data, which can comprise as many as 25 put or delete requests. Individual items to be written can be as large as 400 KB.

Note: *BatchWriteItem* cannot update items. To update items, use the *UpdateItem* API.

The individual *PutItem* and *DeleteItem* operations specified in *BatchWriteItem* are atomic; however *BatchWriteItem* as a whole is not. If any requested operations fail because the table's provisioned throughput is exceeded or an internal processing failure occurs, the failed operations are returned in the *UnprocessedItems* response parameter. You can investigate and optionally resend the requests. Typically, you would call *BatchWriteItem* in a loop. Each iteration would check for unprocessed items and submit a new *BatchWriteItem* request with those unprocessed items until all items have been processed.

Note that if *none* of the items can be processed due to insufficient provisioned throughput on all of the tables in the request, then *BatchWriteItem* will return a *ProvisionedThroughputExceededException*.

Warning: If DynamoDB returns any unprocessed items, you should retry the batch operation on those items. However, *we strongly recommend that you use an exponential backoff algorithm*. If you retry the batch operation immediately, the underlying read or write requests can still fail due to throttling on the individual tables. If you delay the batch operation using exponential backoff, the individual requests in the batch are much more likely to succeed.

For more information, see [Batch Operations and Error Handling](#) in the *Amazon DynamoDB Developer Guide*.

With *BatchWriteItem*, you can efficiently write or delete large amounts of data, such as from Amazon Elastic MapReduce (EMR), or copy data from another database into DynamoDB. In order to improve performance with these large-scale operations, *BatchWriteItem* does not behave in the same way as indi-

vidual *PutItem* and *DeleteItem* calls would. For example, you cannot specify conditions on individual put and delete requests, and *BatchWriteItem* does not return deleted items in the response.

If you use a programming language that supports concurrency, you can use threads to write items in parallel. Your application must include the necessary logic to manage the threads. With languages that don't support threading, you must update or delete the specified items one at a time. In both situations, *BatchWriteItem* provides an alternative where the API performs the specified put and delete operations in parallel, giving you the power of the thread pool approach without having to introduce complexity into your application.

Parallel processing reduces latency, but each specified put and delete request consumes the same number of write capacity units whether it is processed in parallel or not. Delete operations on nonexistent items consume one write capacity unit.

If one or more of the following is true, DynamoDB rejects the entire batch write operation:

- One or more tables specified in the *BatchWriteItem* request does not exist.
- Primary key attributes specified on an item in the request do not match those in the corresponding table's primary key schema.
- You try to perform multiple operations on the same item in the same *BatchWriteItem* request. For example, you cannot put and delete the same item in the same *BatchWriteItem* request.
- There are more than 25 requests in the batch.
- Any individual item in a batch exceeds 400 KB.
- The total request size exceeds 16 MB.

Request Syntax

```
response = dynamodb.batch_write_item(
    RequestItems={
        'string': [
            {
                'PutRequest': {
                    'Item': {
                        'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])|s
                    },
                },
                'DeleteRequest': {
                    'Key': {
                        'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])|s
                    },
                },
            ],
        ],
    },
    ReturnConsumedCapacity='INDEXES'|'TOTAL'|'NONE',
    ReturnItemCollectionMetrics='SIZE'|'NONE'
)
```

Parameters

- **RequestItems** (*dict*) – [REQUIRED]

A map of one or more table names and, for each table, a list of operations to be performed (*DeleteRequest* or *PutRequest*). Each element in the map consists of the following:

- *DeleteRequest* - Perform a *DeleteItem* operation on the specified item. The item to be deleted is identified by a *Key* subelement:
 - * *Key* - A map of primary key attribute values that uniquely identify the ! item. Each entry in this map consists of an attribute name and an attribute value. For each primary key, you must provide *all* of the key attributes. For example, with a hash type primary key, you only need to provide the hash attribute. For a

hash-and-range type primary key, you must provide *both* the hash attribute and the range attribute.

- *PutRequest* - Perform a *PutItem* operation on the specified item. The item to be put is identified by an *Item* subelement: * *Item* - A map of attributes and their values. Each entry in this map consists of an attribute name and an attribute value. Attribute values must not be null; string and binary type attributes must have lengths greater than zero; and set type attributes must not be empty. Requests that contain empty values will be rejected with a *ValidationException* exception. If you specify any attributes that are part of an index key, then the data types for those attributes must match those of the schema in the table's attribute definition.
- (*string*) –
 - * (*list*) –
 - (*dict*) –

Represents an operation to perform - either *DeleteItem* or *PutItem* . You can only request one of these operations, not both, in a single *WriteRequest* . If you do need to perform both of these operations, you will need to provide two separate *WriteRequest* objects.
 - **PutRequest** (*dict*) –

A request to perform a *PutItem* operation.
 - **Item** (*dict*) – [REQUIRED]

A map of attribute name to attribute values, representing the primary key of an item to be processed by *PutItem* . All of the table's primary key attributes must be specified, and their data types must match those of the table's key schema. If any attributes are present in the item which are part of an index key schema for the table, their types must match the index key schema.
 - (*string*) –
 - (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).
 - **DeleteRequest** (*dict*) –

A request to perform a *DeleteItem* operation.
 - **Key** (*dict*) – [REQUIRED]

A map of attribute name to attribute values, representing the primary key of the item to delete. All of the table's primary key attributes must be specified, and their data types must match those of the table's key schema.
 - (*string*) –
 - (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).
- **ReturnConsumedCapacity** (*string*) – Determines the level of detail about provisioned throughput consumption that is returned in the response:
 - *INDEXES* - The response includes the aggregate *ConsumedCapacity* for the operation, together with *ConsumedCapacity* for each table and secondary index that was accessed. Note that some operations, such as *GetItem* and *BatchGetItem* , do not access any indexes at all. In these cases, specifying *INDEXES* will only return *ConsumedCapacity* informa-

- *TOTAL* - The response includes only the aggregate *ConsumedCapacity* for the operation.

- **ReturnItemCollectionMetrics** (*string*) – Determines whether item collection metrics are returned. If set to `SIZE`, the response includes statistics about item collections, if any, that were modified during the operation are returned in the response. If set to `NONE` (the default), no statistics are returned.

Returns

```
{
  'UnprocessedItems': {
    'string': [
      {
        'PutRequest': {
          'Item': {
            'string': 'string'|123|Binary(b'bytes')|True|None|set(['st
          }
        },
        'DeleteRequest': {
          'Key': {
            'string': 'string'|123|Binary(b'bytes')|True|None|set(['st
          }
        }
      ],
    ]
  },
  'ItemCollectionMetrics': {
    'string': [
      {
        'ItemCollectionKey': {
          'string': 'string'|123|Binary(b'bytes')|True|None|set(['string
        },
        'SizeEstimateRangeGB': [
          123.0,
        ]
      },
    ]
  },
  'ConsumedCapacity': [
    {
      'TableName': 'string',
      'CapacityUnits': 123.0,
      'Table': {
        'CapacityUnits': 123.0
      },
      'LocalSecondaryIndexes': {
        'string': {
          'CapacityUnits': 123.0
        }
      },
      'GlobalSecondaryIndexes': {
        'string': {
          'CapacityUnits': 123.0
        }
      }
    }
  ]
}
```

```
    }  
  },  
]  
}
```

Response Structure

- *(dict)* –

Represents the output of a *BatchWriteItem* operation.

- **UnprocessedItems** *(dict)* –

A map of tables and requests against those tables that were not processed. The *UnprocessedItems* value is in the same form as *RequestItems*, so you can provide this value directly to a subsequent *BatchGetItem* operation. For more information, see *RequestItems* in the Request Parameters section.

Each *UnprocessedItems* entry consists of a table name and, for that table, a list of operations to perform (*DeleteRequest* or *PutRequest*).

- * *DeleteRequest* - Perform a *DeleteItem* operation on the specified item. The item to be deleted is identified by a *Key* subelement: * *Key* - A map of primary key attribute values that uniquely identify the item. Each entry in this map consists of an attribute name and an attribute value.
- * *PutRequest* - Perform a *PutItem* operation on the specified item. The item to be put is identified by an *Item* subelement: * *Item* - A map of attributes and their values. Each entry in this map consists of an attribute name and an attribute value. Attribute values must not be null; string and binary type attributes must have lengths greater than zero; and set type attributes must not be empty. Requests that contain empty values will be rejected with a *ValidationException* exception. If you specify any attributes that are part of an index key, then the data types for those attributes must match those of the schema in the table's attribute definition.

If there are no unprocessed items remaining, the response contains an empty *UnprocessedItems* map.

- * *(string)* –

- *(list)* –

- *(dict)* –

Represents an operation to perform - either *DeleteItem* or *PutItem*. You can only request one of these operations, not both, in a single *WriteRequest*. If you do need to perform both of these operations, you will need to provide two separate *WriteRequest* objects.

- **PutRequest** *(dict)* –

A request to perform a *PutItem* operation.

- **Item** *(dict)* –

A map of attribute name to attribute values, representing the primary key of an item to be processed by *PutItem*. All of the table's primary key attributes must be specified, and their data types must match those of the table's key schema. If any attributes are present in the item which are part of an index key schema for the table, their types must match the index key schema.

- *(string)* –

- (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).
- **DeleteRequest** (*dict*) –
A request to perform a *DeleteItem* operation.
- **Key** (*dict*) –
A map of attribute name to attribute values, representing the primary key of the item to delete. All of the table's primary key attributes must be specified, and their data types must match those of the table's key schema.
- (*string*) –
- (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).

– **ItemCollectionMetrics** (*dict*) –

A list of tables that were processed by *BatchWriteItem* and, for each table, information about any item collections that were affected by individual *DeleteItem* or *PutItem* operations.

Each entry consists of the following subelements:

- * *ItemCollectionKey* - The hash key value of the item collection. This is the same as the hash key of the item.
- * *SizeEstimateRange* - An estimate of item collection size, expressed in GB. This is a two-element array containing a lower bound and an upper bound for the estimate. The estimate includes the size of all the items in the table, plus the size of all attributes projected into all of the local secondary indexes on the table. Use this estimate to measure whether a local secondary index is approaching its size limit. The estimate is subject to change over time; therefore, do not rely on the precision or accuracy of the estimate.

* (*string*) –

- (*list*) –
- (*dict*) –

Information about item collections, if any, that were affected by the operation. *ItemCollectionMetrics* is only returned if the request asked for it. If the table does not have any local secondary indexes, this information is not returned in the response.

· **ItemCollectionKey** (*dict*) –

The hash key value of the item collection. This value is the same as the hash key of the item.

- (*string*) –
- (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).
- **SizeEstimateRangeGB** (*list*) –

An estimate of item collection size, in gigabytes. This value is a two-element array containing a lower bound and an upper bound for the estimate. The estimate includes the size of all the items in the table, plus the size of all attributes projected into all of the local secondary indexes on that table. Use this

estimate to measure whether a local secondary index is approaching its size limit.

The estimate is subject to change over time; therefore, do not rely on the precision or accuracy of the estimate.

- *(float)* –

- **ConsumedCapacity** (*list*) –

The capacity units consumed by the operation.

Each element consists of:

- * *TableName* - The table that consumed the provisioned throughput.
- * *CapacityUnits* - The total number of capacity units consumed.

- * *(dict)* –

The capacity units consumed by an operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. *ConsumedCapacity* is only returned if the request asked for it. For more information, see [Provisioned Throughput](#) in the *Amazon DynamoDB Developer Guide*.

- **TableName** (*string*) –

The name of the table that was affected by the operation.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed by the operation.

- **Table** (*dict*) –

The amount of throughput consumed on the table affected by the operation.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

- **LocalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each local index affected by the operation.

- *(string)* –

- *(dict)* –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

- **GlobalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each global index affected by the operation.

- *(string)* –

- *(dict)* –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

create_table (**kwargs)

The *CreateTable* operation adds a new table to your account. In an AWS account, table names must be unique within each region. That is, you can have two tables with same name if you create the tables in different regions.

CreateTable is an asynchronous operation. Upon receiving a *CreateTable* request, DynamoDB immediately returns a response with a *TableStatus* of *CREATING*. After the table is created, DynamoDB sets the *TableStatus* to *ACTIVE*. You can perform read and write operations only on an *ACTIVE* table.

You can optionally define secondary indexes on the new table, as part of the *CreateTable* operation. If you want to create multiple tables with secondary indexes on them, you must create the tables sequentially. Only one table with secondary indexes can be in the *CREATING* state at any given time.

You can use the *DescribeTable* API to check the table status.

Request Syntax

```
table = dynamodb.create_table(
    AttributeDefinitions=[
        {
            'AttributeName': 'string',
            'AttributeType': 'S'|'N'|'B'
        },
    ],
    TableName='string',
    KeySchema=[
        {
            'AttributeName': 'string',
            'KeyType': 'HASH'|'RANGE'
        },
    ],
    LocalSecondaryIndexes=[
        {
            'IndexName': 'string',
            'KeySchema': [
                {
                    'AttributeName': 'string',
                    'KeyType': 'HASH'|'RANGE'
                },
            ],
            'Projection': {
                'ProjectionType': 'ALL'|'KEYS_ONLY'|'INCLUDE',
                'NonKeyAttributes': [
                    'string',
                ]
            }
        },
    ],
    GlobalSecondaryIndexes=[
        {
            'IndexName': 'string',
            'KeySchema': [
                {
                    'AttributeName': 'string',
                    'KeyType': 'HASH'|'RANGE'
                },
            ],
        },
    ],
)
```

```
    ],
    'Projection': {
        'ProjectionType': 'ALL' | 'KEYS_ONLY' | 'INCLUDE',
        'NonKeyAttributes': [
            'string',
        ]
    },
    'ProvisionedThroughput': {
        'ReadCapacityUnits': 123,
        'WriteCapacityUnits': 123
    }
},
ProvisionedThroughput={
    'ReadCapacityUnits': 123,
    'WriteCapacityUnits': 123
},
StreamSpecification={
    'StreamEnabled': True|False,
    'StreamViewType': 'NEW_IMAGE' | 'OLD_IMAGE' | 'NEW_AND_OLD_IMAGES' | 'KEYS_ONLY'
}
)
```

Parameters

- **AttributeDefinitions** (*list*) – [REQUIRED]

An array of attributes that describe the key schema for the table and indexes.

– (*dict*) –

Represents an attribute for describing the key schema for the table and indexes.

- * **AttributeName** (*string*) – [REQUIRED]

A name for the attribute.

- * **AttributeType** (*string*) – [REQUIRED]

The data type for the attribute.

- **TableName** (*string*) – [REQUIRED]

The name of the table to create.

- **KeySchema** (*list*) – [REQUIRED]

Specifies the attributes that make up the primary key for a table or an index. The attributes in *KeySchema* must also be defined in the *AttributeDefinitions* array. For more information, see [Data Model](#) in the *Amazon DynamoDB Developer Guide*.

Each *KeySchemaElement* in the array is composed of:

- *AttributeName* - The name of this key attribute.
- *KeyType* - Determines whether the key attribute is `HASH` or `RANGE`.

For a primary key that consists of a hash attribute, you must provide exactly one element with a *KeyType* of `HASH`.

For a primary key that consists of hash and range attributes, you must provide exactly two elements, in this order: The first element must have a *KeyType* of `HASH`, and the second element must have a *KeyType* of `RANGE`.

For more information, see [Specifying the Primary Key](#) in the *Amazon DynamoDB Developer Guide*.

– (*dict*) –

Represents *a single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement*. A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.

* **AttributeName** (*string*) – [REQUIRED]

The name of a key attribute.

* **KeyType** (*string*) – [REQUIRED]

The attribute data, consisting of the data type and the attribute value itself.

- **LocalSecondaryIndexes** (*list*) – One or more local secondary indexes (the maximum is five) to be created on the table. Each index is scoped to a given hash key value. There is a 10 GB size limit per hash key; otherwise, the size of a local secondary index is unconstrained.

Each local secondary index in the array includes the following:

- *IndexName* - The name of the local secondary index. Must be unique only for this table.
- *KeySchema* - Specifies the key schema for the local secondary index. The key schema must begin with the same hash key attribute as the table.
- *Projection* - Specifies attributes that are copied (projected) from the table into the index. These are in addition to the primary key attributes and index key attributes, which are automatically projected. Each attribute specification is composed of:
 - * *ProjectionType* - One of the following:
 - * KEYS_ONLY - Only the index and primary keys are projected into the index.
 - * INCLUDE - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes*.
 - * ALL - All of the table attributes are projected into the index.
 - * *NonKeyAttributes* - A list of one or more non-key attribute names that are projected into the secondary index. The total count of attributes provided in *NonKeyAttributes*, summed across all of the secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.
- (*dict*) –

Represents the properties of a local secondary index.

* **IndexName** (*string*) – [REQUIRED]

The name of the local secondary index. The name must be unique among all other indexes on this table.

* **KeySchema** (*list*) – [REQUIRED]

The complete key schema for the local secondary index, consisting of one or more pairs of attribute names and key types (HASH or RANGE).

· (*dict*) –

Represents *a single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement*. A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.

- **AttributeName** (*string*) – [REQUIRED]

The name of a key attribute.

- **KeyType** (*string*) – [REQUIRED]

The attribute data, consisting of the data type and the attribute value itself.

- * **Projection** (*dict*) – [REQUIRED]

Represents attributes that are copied (projected) from the table into an index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

- **ProjectionType** (*string*) –

The set of attributes that are projected into the index:

- KEYS_ONLY - Only the index and primary keys are projected into the index.
- INCLUDE - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes*.
- ALL - All of the table attributes are projected into the index.
- **NonKeyAttributes** (*list*) –

Represents the non-key attribute names which will be projected into the index.

For local secondary indexes, the total count of *NonKeyAttributes* summed across all of the local secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.

- (*string*) –

- **GlobalSecondaryIndexes** (*list*) – One or more global secondary indexes (the maximum is five) to be created on the table. Each global secondary index in the array includes the following:

- *IndexName* - The name of the global secondary index. Must be unique only for this table.
- *KeySchema* - Specifies the key schema for the global secondary index.
- *Projection* - Specifies attributes that are copied (projected) from the table into the index. These are in addition to the primary key attributes and index key attributes, which are automatically projected. Each attribute specification is composed of:
 - * *ProjectionType* - One of the following:
 - * KEYS_ONLY - Only the index and primary keys are projected into the index.
 - INCLUDE - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes*.
 - ALL - All of the table attributes are projected into the index.
 - * *NonKeyAttributes* - A list of one or more non-key attribute names that are projected into the secondary index. The total count of attributes provided in *NonKeyAttributes*, summed across all of the secondary indexes, must not exceed 20. If you project the same attribute into two different indexes,

- this counts as two distinct attributes when determining the total.
- *ProvisionedThroughput* - The provisioned throughput settings for the global secondary index, consisting of read and write capacity units.
 - (*dict*) –

Represents the properties of a global secondary index.

* **IndexName** (*string*) – [REQUIRED]

The name of the global secondary index. The name must be unique among all other indexes on this table.

* **KeySchema** (*list*) – [REQUIRED]

The complete key schema for a global secondary index, which consists of one or more pairs of attribute names and key types (HASH or RANGE).

· (*dict*) –

Represents a *single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement* . A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.

· **AttributeName** (*string*) – [REQUIRED]

The name of a key attribute.

· **KeyType** (*string*) – [REQUIRED]

The attribute data, consisting of the data type and the attribute value itself.

* **Projection** (*dict*) – [REQUIRED]

Represents attributes that are copied (projected) from the table into an index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

· **ProjectionType** (*string*) –

The set of attributes that are projected into the index:

- KEYS_ONLY - Only the index and primary keys are projected into the index.
- INCLUDE - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes* .
- ALL - All of the table attributes are projected into the index.
- **NonKeyAttributes** (*list*) –

Represents the non-key attribute names which will be projected into the index.

For local secondary indexes, the total count of *NonKeyAttributes* summed across all of the local secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.

· (*string*) –

* **ProvisionedThroughput** (*dict*) – [REQUIRED]

Represents the provisioned throughput settings for a specified table or index. The settings can be modified using the *UpdateTable* operation.

For current minimum and maximum provisioned throughput values, see [Limits](#) in the *Amazon DynamoDB Developer Guide* .

· **ReadCapacityUnits** (*integer*) – [REQUIRED]

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a *ThrottlingException* . For more information, see [Specifying Read and Write Requirements](#) in the *Amazon DynamoDB Developer Guide* .

· **WriteCapacityUnits** (*integer*) – [REQUIRED]

The maximum number of writes consumed per second before DynamoDB returns a *ThrottlingException* . For more information, see [Specifying Read and Write Requirements](#) in the *Amazon DynamoDB Developer Guide* .

• **ProvisionedThroughput** (*dict*) – [REQUIRED]

Represents the provisioned throughput settings for a specified table or index. The settings can be modified using the *UpdateTable* operation.

For current minimum and maximum provisioned throughput values, see [Limits](#) in the *Amazon DynamoDB Developer Guide* .

– **ReadCapacityUnits** (*integer*) – [REQUIRED]

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a *ThrottlingException* . For more information, see [Specifying Read and Write Requirements](#) in the *Amazon DynamoDB Developer Guide* .

– **WriteCapacityUnits** (*integer*) – [REQUIRED]

The maximum number of writes consumed per second before DynamoDB returns a *ThrottlingException* . For more information, see [Specifying Read and Write Requirements](#) in the *Amazon DynamoDB Developer Guide* .

• **StreamSpecification** (*dict*) – The settings for DynamoDB Streams on the table. These settings consist of:

- *StreamEnabled* - Indicates whether Streams is to be enabled (true) or disabled (false).
- *StreamViewType* - When an item in the table is modified, *StreamViewType* determines what information is written to the table's stream. Valid values for *StreamViewType* are: * *KEYS_ONLY* - Only the key attributes of the modified item are written to the stream.
- *NEW_IMAGE* - The entire item, as it appears after it was modified, is written to the stream.
- *OLD_IMAGE* - The entire item, as it appeared before it was modified, is written to the stream.
- *NEW_AND_OLD_IMAGES* - Both the new and the old item images of the item are written to the stream.

– **StreamEnabled** (*boolean*) –

Indicates whether DynamoDB Streams is enabled (true) or disabled (false) on the table.

– **StreamViewType** (*string*) –

The DynamoDB Streams settings for the table. These settings consist of:

- * *StreamEnabled* - Indicates whether DynamoDB Streams is enabled (true) or disabled (false) on the table.
- * *StreamViewType* - When an item in the table is modified, *StreamViewType* determines what information is written to the stream for this table. Valid values for *StreamViewType* are:
 - * *KEYS_ONLY* - Only the key attributes of the modified item are written to the stream.
 - * *NEW_IMAGE* - The entire item, as it appears after it was modified, is written to the stream.
 - * *OLD_IMAGE* - The entire item, as it appeared before it was modified, is written to the stream.
 - * *NEW_AND_OLD_IMAGES* - Both the new and the old item images of the item are written to the stream.

Return type `dynamodb.Table`

Returns Table resource

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Table (*name*)

Creates a Table resource.:

```
table = dynamodb.Table('name')
```

Parameters *name* (*string*) – The Table's name identifier. This **must** be set.

Return type `DynamoDB.Table`

Returns A Table resource

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

tables

all ()

Creates an iterable of all Table resources in the collection.

Request Syntax

```
table_iterator = dynamodb.tables.all()
```

Return type `list(dynamodb.Table)`

Returns A list of Table resources

filter (***kwargs*)

Creates an iterable of all Table resources in the collection filtered by kwargs passed to method.

Request Syntax

```
table_iterator = dynamodb.tables.filter(
    ExclusiveStartTableName='string',
    Limit=123
)
```

Parameters

- **ExclusiveStartTableName** (*string*) – The first table name that this operation will evaluate. Use the value that was returned for *LastEvaluatedTableName* in a previous operation, so that you can obtain the next page of results.
- **Limit** (*integer*) – A maximum number of table names to return. If this parameter is not specified, the limit is 100.

Return type `list(dynamodb.Table)`

Returns A list of Table resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Table resources in the collection.

Request Syntax

```
table_iterator = dynamodb.tables.limit(  
    count=123  
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(dynamodb.Table)`

Returns A list of Table resources

page_size (***kwargs*)

Creates an iterable of all Table resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
table_iterator = dynamodb.tables.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(dynamodb.Table)`

Returns A list of Table resources

Table

class `DynamoDB.Table` (*name*)

A resource representing an Amazon DynamoDB Table:

```
import boto3
```

```
dynamodb = boto3.resource('dynamodb')  
table = dynamodb.Table('name')
```

Parameters **name** (*string*) – The Table’s name identifier. This **must** be set.

These are the resource’s available identifiers:

- `name`

These are the resource’s available attributes:

- `attribute_definitions`
- `creation_date_time`
- `global_secondary_indexes`
- `item_count`
- `key_schema`

- latest_stream_arn
- latest_stream_label
- local_secondary_indexes
- provisioned_throughput
- stream_specification
- table_arn
- table_name
- table_size_bytes
- table_status

These are the resource's available actions:

- batch_writer()
- delete()
- delete_item()
- get_item()
- load()
- put_item()
- query()
- reload()
- scan()
- update()
- update_item()

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

name

(string) The Table's name identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

attribute_definitions

(list)

An array of *AttributeDefinition* objects. Each of these objects describes one attribute in the table and index key schema.

Each *AttributeDefinition* object in this array is composed of:

- AttributeName* - The name of the attribute.
- AttributeType* - The data type for the attribute.

creation_date_time

(datetime)

The date and time when the table was created, in [UNIX epoch time](#) format.

global_secondary_indexes

(list)

The global secondary indexes, if any, on the table. Each index is scoped to a given hash key value. Each element is composed of:

- Backfilling* - If true, then the index is currently in the backfilling phase. Backfilling occurs only when a new global secondary index is added to the table; it is the process by which DynamoDB populates the new index with data from the table. (This attribute does not appear for indexes that were created during a *CreateTable* operation.)
- IndexName* - The name of the global secondary index.

- IndexSizeBytes* - The total size of the global secondary index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.
- IndexStatus* - The current status of the global secondary index: * *CREATING* - The index is being created.
- UPDATING* - The index is being updated.
- DELETING* - The index is being deleted.
- ACTIVE* - The index is ready for use.
- ItemCount* - The number of items in the global secondary index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.
- KeySchema* - Specifies the complete index key schema. The attribute names in the key schema must be between 1 and 255 characters (inclusive). The key schema must begin with the same hash key attribute as the table.
- Projection* - Specifies attributes that are copied (projected) from the table into the index. These are in addition to the primary key attributes and index key attributes, which are automatically projected. Each attribute specification is composed of: * *ProjectionType* - One of the following: * *KEYS_ONLY* - Only the index and primary keys are projected into the index.
- INCLUDE* - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes* .
- ALL* - All of the table attributes are projected into the index.
- NonKeyAttributes* - A list of one or more non-key attribute names that are projected into the secondary index. The total count of attributes provided in *NonKeyAttributes* , summed across all of the secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.
- ProvisionedThroughput* - The provisioned throughput settings for the global secondary index, consisting of read and write capacity units, along with data about increases and decreases.

If the table is in the *DELETING* state, no information about indexes will be returned.

item_count
(integer)

The number of items in the specified table. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

key_schema
(list)

The primary key structure for the table. Each *KeySchemaElement* consists of:

- AttributeName* - The name of the attribute.
- KeyType* - The key type for the attribute. Can be either *HASH* or *RANGE* .

For more information about primary keys, see [Primary Key](#) in the *Amazon DynamoDB Developer Guide* .

latest_stream_arn
(string)

The Amazon Resource Name (ARN) that uniquely identifies the latest stream for this table.

latest_stream_label
(string)

A timestamp, in ISO 8601 format, for this stream.

Note that *LatestStreamLabel* is not a unique identifier for the stream, because it is possible that a stream from another table might have the same timestamp. However, the combination of the following three elements is guaranteed to be unique:

- the AWS customer ID.
- the table name.
- the *StreamLabel* .

local_secondary_indexes*(list)*

Represents one or more local secondary indexes on the table. Each index is scoped to a given hash key value. Tables with one or more local secondary indexes are subject to an item collection size limit, where the amount of data within a given item collection cannot exceed 10 GB. Each element is composed of:

- *IndexName* - The name of the local secondary index.
- *KeySchema* - Specifies the complete index key schema. The attribute names in the key schema must be between 1 and 255 characters (inclusive). The key schema must begin with the same hash key attribute as the table.
- *Projection* - Specifies attributes that are copied (projected) from the table into the index. These are in addition to the primary key attributes and index key attributes, which are automatically projected. Each attribute specification is composed of:
 - * *ProjectionType* - One of the following:
 - * *KEYS_ONLY* - Only the index and primary keys are projected into the index.
 - * *INCLUDE* - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes*.
 - * *ALL* - All of the table attributes are projected into the index.
- *NonKeyAttributes* - A list of one or more non-key attribute names that are projected into the secondary index. The total count of attributes provided in *NonKeyAttributes*, summed across all of the secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.
- *IndexSizeBytes* - Represents the total size of the index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.
- *ItemCount* - Represents the number of items in the index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

If the table is in the `DELETING` state, no information about indexes will be returned.

provisioned_throughput*(dict)*

The provisioned throughput settings for the table, consisting of read and write capacity units, along with data about increases and decreases.

stream_specification*(dict)*

The current DynamoDB Streams configuration for the table.

table_arn*(string)*

The Amazon Resource Name (ARN) that uniquely identifies the table.

table_name*(string)*

The name of the table.

table_size_bytes*(integer)*

The total size of the specified table, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

table_status*(string)*

The current state of the table:

- *CREATING* - The table is being created.
- *UPDATING* - The table is being updated.

- DELETING* - The table is being deleted.
- ACTIVE* - The table is ready for use.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

`batch_writer()`

Create a batch writer object.

This method creates a context manager for writing objects to Amazon DynamoDB in batch.

The batch writer will automatically handle buffering and sending items in batches. In addition, the batch writer will also automatically handle any unprocessed items and resend them as needed. All you need to do is call `put_item` for any items you want to add, and `delete_item` for any items you want to delete.

Example usage:

```
with table.batch_writer() as batch:
    for _ in xrange(1000000):
        batch.put_item(Item={'HashKey': '...',
                             'Otherstuff': '...'})
    # You can also delete items in a batch.
    batch.delete_item(Key={'HashKey': 'SomeHashKey'})
```

`delete()`

The *DeleteTable* operation deletes a table and all of its items. After a *DeleteTable* request, the specified table is in the *DELETING* state until DynamoDB completes the deletion. If the table is in the *ACTIVE* state, you can delete it. If a table is in *CREATING* or *UPDATING* states, then DynamoDB returns a *ResourceInUseException*. If the specified table does not exist, DynamoDB returns a *ResourceNotFoundException*. If table is already in the *DELETING* state, no error is returned.

Note: DynamoDB might continue to accept data read and write operations, such as *GetItem* and *PutItem*, on a table in the *DELETING* state until the table deletion is complete.

When you delete a table, any indexes on that table are also deleted.

If you have DynamoDB Streams enabled on the table, then the corresponding stream on that table goes into the *DISABLED* state, and the stream is automatically deleted after 24 hours.

Use the *DescribeTable* API to check the status of the table.

Request Syntax

```
response = table.delete()
```

Return type dict

Returns

Response Syntax

```
{
    'TableDescription': {
        'AttributeDefinitions': [
            {
                'AttributeName': 'string',
```

```

        'AttributeType': 'S'|'N'|'B'
    },
],
'TableName': 'string',
'KeySchema': [
    {
        'AttributeName': 'string',
        'KeyType': 'HASH'|'RANGE'
    },
],
'TableStatus': 'CREATING'|'UPDATING'|'DELETING'|'ACTIVE',
'CreationDateTime': datetime(2015, 1, 1),
'ProvisionedThroughput': {
    'LastIncreaseDateTime': datetime(2015, 1, 1),
    'LastDecreaseDateTime': datetime(2015, 1, 1),
    'NumberOfDecreasesToday': 123,
    'ReadCapacityUnits': 123,
    'WriteCapacityUnits': 123
},
'TableSizeBytes': 123,
'ItemCount': 123,
'TableArn': 'string',
'LocalSecondaryIndexes': [
    {
        'IndexName': 'string',
        'KeySchema': [
            {
                'AttributeName': 'string',
                'KeyType': 'HASH'|'RANGE'
            },
        ],
        'Projection': {
            'ProjectionType': 'ALL'|'KEYS_ONLY'|'INCLUDE',
            'NonKeyAttributes': [
                'string',
            ]
        },
        'IndexSizeBytes': 123,
        'ItemCount': 123,
        'IndexArn': 'string'
    },
],
'GlobalSecondaryIndexes': [
    {
        'IndexName': 'string',
        'KeySchema': [
            {
                'AttributeName': 'string',
                'KeyType': 'HASH'|'RANGE'
            },
        ],
        'Projection': {
            'ProjectionType': 'ALL'|'KEYS_ONLY'|'INCLUDE',
            'NonKeyAttributes': [
                'string',
            ]
        },
        'IndexStatus': 'CREATING'|'UPDATING'|'DELETING'|'ACTIVE',
    },
],

```

```
        'Backfilling': True|False,
        'ProvisionedThroughput': {
            'LastIncreaseDateTime': datetime(2015, 1, 1),
            'LastDecreaseDateTime': datetime(2015, 1, 1),
            'NumberOfDecreasesToday': 123,
            'ReadCapacityUnits': 123,
            'WriteCapacityUnits': 123
        },
        'IndexSizeBytes': 123,
        'ItemCount': 123,
        'IndexArn': 'string'
    },
],
'StreamSpecification': {
    'StreamEnabled': True|False,
    'StreamViewType': 'NEW_IMAGE' | 'OLD_IMAGE' | 'NEW_AND_OLD_IMAGES' | 'K
},
'LatestStreamLabel': 'string',
'LatestStreamArn': 'string'
}
}
```

Response Structure

- (*dict*) –

Represents the output of a *DeleteTable* operation.

- **TableDescription** (*dict*) –

Represents the properties of a table.

- * **AttributeDefinitions** (*list*) –

An array of *AttributeDefinition* objects. Each of these objects describes one attribute in the table and index key schema.

Each *AttributeDefinition* object in this array is composed of:

- *AttributeName* - The name of the attribute.
- *AttributeType* - The data type for the attribute.

- (*dict*) –

Represents an attribute for describing the key schema for the table and indexes.

- **AttributeName** (*string*) –

A name for the attribute.

- **AttributeType** (*string*) –

The data type for the attribute.

- * **TableName** (*string*) –

The name of the table.

- * **KeySchema** (*list*) –

The primary key structure for the table. Each *KeySchemaElement* consists of:

- *AttributeName* - The name of the attribute.
- *KeyType* - The key type for the attribute. Can be either `HASH` or `RANGE`.

For more information about primary keys, see [Primary Key](#) in the *Amazon DynamoDB Developer Guide*.

- (*dict*) –

Represents a *single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement*. A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.

- **AttributeName** (*string*) –

The name of a key attribute.

- **KeyType** (*string*) –

The attribute data, consisting of the data type and the attribute value itself.

- * **TableStatus** (*string*) –

The current state of the table:

- *CREATING* - The table is being created.
- *UPDATING* - The table is being updated.
- *DELETING* - The table is being deleted.
- *ACTIVE* - The table is ready for use.

- * **CreationDateTime** (*datetime*) –

The date and time when the table was created, in [UNIX epoch time](#) format.

- * **ProvisionedThroughput** (*dict*) –

The provisioned throughput settings for the table, consisting of read and write capacity units, along with data about increases and decreases.

- **LastIncreaseDateTime** (*datetime*) –

The date and time of the last provisioned throughput increase for this table.

- **LastDecreaseDateTime** (*datetime*) –

The date and time of the last provisioned throughput decrease for this table.

- **NumberOfDecreasesToday** (*integer*) –

The number of provisioned throughput decreases for this table during this UTC calendar day. For current maximums on provisioned throughput decreases, see [Limits](#) in the *Amazon DynamoDB Developer Guide*.

- **ReadCapacityUnits** (*integer*) –

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a *ThrottlingException*. Eventually consistent reads require less effort than strongly consistent reads, so a setting of 50 *ReadCapacityUnits* per second provides 100 eventually consistent *ReadCapacityUnits* per second.

- **WriteCapacityUnits** (*integer*) –

The maximum number of writes consumed per second before DynamoDB returns a *ThrottlingException*.

* **TableSizeBytes** (*integer*) –

The total size of the specified table, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

* **ItemCount** (*integer*) –

The number of items in the specified table. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

* **TableArn** (*string*) –

The Amazon Resource Name (ARN) that uniquely identifies the table.

* **LocalSecondaryIndexes** (*list*) –

Represents one or more local secondary indexes on the table. Each index is scoped to a given hash key value. Tables with one or more local secondary indexes are subject to an item collection size limit, where the amount of data within a given item collection cannot exceed 10 GB. Each element is composed of:

- *IndexName* - The name of the local secondary index.
- *KeySchema* - Specifies the complete index key schema. The attribute names in the key schema must be between 1 and 255 characters (inclusive). The key schema must begin with the same hash key attribute as the table.
- *Projection* - Specifies attributes that are copied (projected) from the table into the index. These are in addition to the primary key attributes and index key attributes, which are automatically projected. Each attribute specification is composed of: * *ProjectionType* - One of the following: * *KEYS_ONLY* - Only the index and primary keys are projected into the index.
- *INCLUDE* - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes*.
- *ALL* - All of the table attributes are projected into the index.
- *NonKeyAttributes* - A list of one or more non-key attribute names that are projected into the secondary index. The total count of attributes provided in *NonKeyAttributes*, summed across all of the secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.
- *IndexSizeBytes* - Represents the total size of the index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.
- *ItemCount* - Represents the number of items in the index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

If the table is in the `DELETING` state, no information about indexes will be returned.

- *(dict)* –

Represents the properties of a local secondary index.

- **IndexName** (*string*) –

Represents the name of the local secondary index.

- **KeySchema** (*list*) –

The complete index key schema, which consists of one or more pairs of attribute names and key types (`HASH` or `RANGE`).

- *(dict)* –

Represents a *single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement*. A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.

- **AttributeName** (*string*) –

The name of a key attribute.

- **KeyType** (*string*) –

The attribute data, consisting of the data type and the attribute value itself.

- **Projection** (*dict*) –

Represents attributes that are copied (projected) from the table into an index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

- **ProjectionType** (*string*) –

The set of attributes that are projected into the index:

- `KEYS_ONLY` - Only the index and primary keys are projected into the index.

- `INCLUDE` - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes*.

- `ALL` - All of the table attributes are projected into the index.

- **NonKeyAttributes** (*list*) –

Represents the non-key attribute names which will be projected into the index.

For local secondary indexes, the total count of *NonKeyAttributes* summed across all of the local secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.

- (*string*) –

- **IndexSizeBytes** (*integer*) –

The total size of the specified index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- **ItemCount** (*integer*) –

The number of items in the specified index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- **IndexArn** (*string*) –

The Amazon Resource Name (ARN) that uniquely identifies the index.

- * **GlobalSecondaryIndexes** (*list*) –

The global secondary indexes, if any, on the table. Each index is scoped to a given hash key value. Each element is composed of:

- *Backfilling* - If true, then the index is currently in the backfilling phase. Backfilling occurs only when a new global secondary index is added to the table; it is the process by which DynamoDB populates the new index with data from the table. (This attribute does not appear for indexes that were created during a *CreateTable* operation.)
- *IndexName* - The name of the global secondary index.
- *IndexSizeBytes* - The total size of the global secondary index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.
- *IndexStatus* - The current status of the global secondary index: * *CREATING* - The index is being created.
- *UPDATING* - The index is being updated.
- *DELETING* - The index is being deleted.
- *ACTIVE* - The index is ready for use.
- *ItemCount* - The number of items in the global secondary index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.
- *KeySchema* - Specifies the complete index key schema. The attribute names in the key schema must be between 1 and 255 characters (inclusive). The key schema must begin with the same hash key attribute as the table.
- *Projection* - Specifies attributes that are copied (projected) from the table into the index. These are in addition to the primary key attributes and index key attributes, which are automatically projected. Each attribute specification is composed of: * *ProjectionType* - One of the following: * *KEYS_ONLY* - Only the index and primary keys are projected into the index.
- *INCLUDE* - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes* .

- **ALL** - All of the table attributes are projected into the index.
- **NonKeyAttributes** - A list of one or more non-key attribute names that are projected into the secondary index. The total count of attributes provided in *NonKeyAttributes*, summed across all of the secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.
- **ProvisionedThroughput** - The provisioned throughput settings for the global secondary index, consisting of read and write capacity units, along with data about increases and decreases.

If the table is in the `DELETING` state, no information about indexes will be returned.

- *(dict)* –

Represents the properties of a global secondary index.

- **IndexName** (*string*) –

The name of the global secondary index.

- **KeySchema** (*list*) –

The complete key schema for the global secondary index, consisting of one or more pairs of attribute names and key types (`HASH` or `RANGE`).

- *(dict)* –

Represents a *single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement*. A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.

- **AttributeName** (*string*) –

The name of a key attribute.

- **KeyType** (*string*) –

The attribute data, consisting of the data type and the attribute value itself.

- **Projection** (*dict*) –

Represents attributes that are copied (projected) from the table into an index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

- **ProjectionType** (*string*) –

The set of attributes that are projected into the index:

- **KEYS_ONLY** - Only the index and primary keys are projected into the index.
- **INCLUDE** - Only the specified table attributes are projected into the index. The list of projected attributes are

in *NonKeyAttributes* .

- **ALL** - All of the table attributes are projected into the index.
- **NonKeyAttributes** (*list*) –

Represents the non-key attribute names which will be projected into the index.

For local secondary indexes, the total count of *NonKeyAttributes* summed across all of the local secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.

- (*string*) –
- **IndexStatus** (*string*) –

The current state of the global secondary index:

- **CREATING** - The index is being created.
- **UPDATING** - The index is being updated.
- **DELETING** - The index is being deleted.
- **ACTIVE** - The index is ready for use.
- **Backfilling** (*boolean*) –

Indicates whether the index is currently backfilling. *Backfilling* is the process of reading items from the table and determining whether they can be added to the index. (Not all items will qualify: For example, a hash key attribute cannot have any duplicates.) If an item can be added to the index, DynamoDB will do so. After all items have been processed, the backfilling operation is complete and *Backfilling* is false.

Note: For indexes that were created during a *CreateTable* operation, the *Backfilling* attribute does not appear in the *DescribeTable* output.

- **ProvisionedThroughput** (*dict*) –

Represents the provisioned throughput settings for the table, consisting of read and write capacity units, along with data about increases and decreases.

- **LastIncreaseDateTime** (*datetime*) –

The date and time of the last provisioned throughput increase for this table.

- **LastDecreaseDateTime** (*datetime*) –

The date and time of the last provisioned throughput decrease for this table.

- **NumberOfDecreasesToday** (*integer*) –

The number of provisioned throughput decreases for this table during this UTC calendar day. For current maximums on provisioned throughput decreases, see [Limits](#) in the *Amazon DynamoDB Developer Guide* .

- **ReadCapacityUnits** (*integer*) –

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a

ThrottlingException . Eventually consistent reads require less effort than strongly consistent reads, so a setting of 50 *ReadCapacityUnits* per second provides 100 eventually consistent *ReadCapacityUnits* per second.

- **WriteCapacityUnits** (*integer*) –

The maximum number of writes consumed per second before DynamoDB returns a *ThrottlingException* .

- **IndexSizeBytes** (*integer*) –

The total size of the specified index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- **ItemCount** (*integer*) –

The number of items in the specified index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- **IndexArn** (*string*) –

The Amazon Resource Name (ARN) that uniquely identifies the index.

- * **StreamSpecification** (*dict*) –

The current DynamoDB Streams configuration for the table.

- **StreamEnabled** (*boolean*) –

Indicates whether DynamoDB Streams is enabled (true) or disabled (false) on the table.

- **StreamViewType** (*string*) –

The DynamoDB Streams settings for the table. These settings consist of:

- *StreamEnabled* - Indicates whether DynamoDB Streams is enabled (true) or disabled (false) on the table.
- *StreamViewType* - When an item in the table is modified, *StreamViewType* determines what information is written to the stream for this table. Valid values for *StreamViewType* are: * *KEYS_ONLY* - Only the key attributes of the modified item are written to the stream.
- *NEW_IMAGE* - The entire item, as it appears after it was modified, is written to the stream.
- *OLD_IMAGE* - The entire item, as it appeared before it was modified, is written to the stream.
- *NEW_AND_OLD_IMAGES* - Both the new and the old item images of the item are written to the stream.

- * **LatestStreamLabel** (*string*) –

A timestamp, in ISO 8601 format, for this stream.

Note that *LatestStreamLabel* is not a unique identifier for the stream, because it is possible that a stream from another table might have the same timestamp. However, the combination of the following three elements is guaranteed to be unique:

- the AWS customer ID.

- the table name.
- the *StreamLabel*.

* **LatestStreamArn** (*string*) –

The Amazon Resource Name (ARN) that uniquely identifies the latest stream for this table.

delete_item (**kwargs)

Deletes a single item in a table by primary key. You can perform a conditional delete operation that deletes the item if it exists, or if it has an expected attribute value.

In addition to deleting an item, you can also return the item's attribute values in the same operation, using the *ReturnValues* parameter.

Unless you specify conditions, the *DeleteItem* is an idempotent operation; running it multiple times on the same item or attribute does *not* result in an error response.

Conditional deletes are useful for deleting items only if specific conditions are met. If those conditions are met, DynamoDB performs the delete. Otherwise, the item is not deleted.

Request Syntax

```
response = table.delete_item(  
    Key={  
        'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])|set([123])|set(  
    },  
    Expected={  
        'string': {  
            'Value': 'string'|123|Binary(b'bytes')|True|None|set(['string'])|set([123])|set(  
            'Exists': True|False,  
            'ComparisonOperator': 'EQ'|'NE'|'IN'|'LE'|'LT'|'GE'|'GT'|'BETWEEN'|'NOT_NULL'|'NOT_EQUAL'|  
            'AttributeValueList': [  
                'string'|123|Binary(b'bytes')|True|None|set(['string'])|set([123])|set(  
            ]  
        }  
    },  
    ConditionalOperator='AND'|'OR',  
    ReturnValues='NONE'|'ALL_OLD'|'UPDATED_OLD'|'ALL_NEW'|'UPDATED_NEW',  
    ReturnConsumedCapacity='INDEXES'|'TOTAL'|'NONE',  
    ReturnItemCollectionMetrics='SIZE'|'NONE',  
    ConditionExpression=Attr('myattribute').eq('myvalue'),  
    ExpressionAttributeNames={  
        'string': 'string'  
    },  
    ExpressionAttributeValues={  
        'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])|set([123])|set(  
    }  
)
```

Parameters

- **Key** (*dict*) – **[REQUIRED]**

A map of attribute names to *AttributeValue* objects, representing the primary key of the item to delete.

For the primary key, you must provide all of the attributes. For example, with a hash type primary key, you only need to provide the hash attribute. For a hash-and-range type primary key, you must provide both the hash attribute and the range attribute.

– (*string*) –

* (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).

- **Expected** (*dict*) –

Warning: This is a legacy parameter, for backward compatibility. New applications should use *ConditionExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception.

A map of attribute/condition pairs. *Expected* provides a conditional block for the *DeleteItem* operation.

Each element of *Expected* consists of an attribute name, a comparison operator, and one or more values. DynamoDB compares the attribute with the value(s) you supplied, using the comparison operator. For each *Expected* element, the result of the evaluation is either true or false.

If you specify more than one element in the *Expected* map, then by default all of the conditions must evaluate to true. In other words, the conditions are ANDed together. (You can use the *ConditionalOperator* parameter to OR the conditions instead. If you do this, then at least one of the conditions must evaluate to true, rather than all of them.)

If the *Expected* map evaluates to true, then the conditional operation succeeds; otherwise, it fails.

Expected contains the following:

- *AttributeValueList* - One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *ComparisonOperator* being used. For type Number, value comparisons are numeric. String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A , and a is greater than B . For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters . For type Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.
- *ComparisonOperator* - A comparator for evaluating attributes in the *AttributeValueList* . When performing the comparison, DynamoDB uses strongly consistent reads. The following comparison operators are available: EQ | NE | LE | LT | GE | GT | NOT_NULL | NULL | CONTAINS | NOT_CONTAINS | BEGINS_WITH | IN | BETWEEN The following are descriptions of each comparison operator. * EQ : Equal. EQ is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] } .
- NE : Not equal. NE is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* of a different type than the

one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not equal `{"NS": ["6", "2", "1"]}`.

- **LE** : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
- **LT** : Less than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
- **GE** : Greater than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
- **GT** : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
- **NOT_NULL** : The attribute exists. **NOT_NULL** is supported for all datatypes, including lists and maps. .. note::This operator tests for the existence of an attribute, not its data type. If the data type of attribute “a” is null, and you evaluate it using **NOT_NULL**, the result is a Boolean *true*. This result is because the attribute “a” exists; its data type is not relevant to the **NOT_NULL** comparison operator.
- **NULL** : The attribute does not exist. **NULL** is supported for all datatypes, including lists and maps. .. note::This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute “a” is null, and you evaluate it using **NULL**, the result is a Boolean *false*. This is because the attribute “a” exists; its data type is not relevant to the **NULL** comparison operator.
- **CONTAINS** : Checks for a subsequence, or value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS”, “NS”, or “BS”), then the operator evaluates to true if it finds an exact match with any member of the set. **CONTAINS** is supported for lists: When evaluating “a **CONTAINS** b”, “a” can be a list; however, “b” cannot be a set, a map, or a list.

- `NOT_CONTAINS` : Checks for absence of a subsequence, or absence of a value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS”, “NS”, or “BS”), then the operator evaluates to true if it *does not* find an exact match with any member of the set. `NOT_CONTAINS` is supported for lists: When evaluating “a NOT CONTAINS b”, “a” can be a list; however, “b” cannot be a set, a map, or a list.
- `BEGINS_WITH` : Checks for a prefix. *AttributeValueList* can contain only one *AttributeValue* of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).
- `IN` : Checks for matching elements within two sets. *AttributeValueList* can contain one or more *AttributeValue* elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
- `BETWEEN` : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, {“S”: “6”} does not compare to {“N”: “6”} . Also, {“N”: “6”} does not compare to {“NS”: [“6”, “2”, “1”]}

For usage examples of *AttributeValueList* and *ComparisonOperator* , see [Legacy Conditional Parameters](#) in the *Amazon DynamoDB Developer Guide* .

For backward compatibility with previous DynamoDB releases, the following parameters can be used instead of *AttributeValueList* and *ComparisonOperator* :

- *Value* - A value for DynamoDB to compare with an attribute.
- *Exists* - A Boolean value that causes DynamoDB to evaluate the value before attempting the conditional operation: * If *Exists* is `true` , DynamoDB will check to see if that attribute value already exists in the table. If it is found, then the condition evaluates to true; otherwise the condition evaluate to false.
- If *Exists* is `false` , DynamoDB assumes that the attribute value does *not* exist in the table. If in fact the value does not exist, then the assumption is valid and the condition evaluates to true. If the value is found, despite the assumption that it does not exist, the condition evaluates to false.

Note that the default value for *Exists* is `true` .

The *Value* and *Exists* parameters are incompatible with *AttributeValueList* and *ComparisonOperator* . Note that if you use both sets of parameters at once, DynamoDB will return a *ValidationException* exception.

Note: This parameter does not support attributes of type List or Map.

- (string) –
- * (dict) –

Represents a condition to be compared with an attribute value. This condition can be used with *DeleteItem*, *PutItem* or *UpdateItem* operations; if the comparison evaluates to true, the operation succeeds; if not, the operation fails. You can use *ExpectedAttributeValue* in one of two different ways:

- Use *AttributeValueList* to specify one or more values to compare against an attribute. Use *ComparisonOperator* to specify how you want to perform the comparison. If the comparison evaluates to true, then the conditional operation succeeds.
- Use *Value* to specify a value that DynamoDB will compare against an attribute. If the values match, then *ExpectedAttributeValue* evaluates to true and the conditional operation succeeds. Optionally, you can also set *Exists* to false, indicating that you *do not* expect to find the attribute value in the table. In this case, the conditional operation succeeds only if the comparison evaluates to false.

Value and *Exists* are incompatible with *AttributeValueList* and *ComparisonOperator*. Note that if you use both sets of parameters at once, DynamoDB will return a *ValidationException*.

- **Value** (valid *DynamoDB* type) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).
- **Exists** (boolean) –

Causes DynamoDB to evaluate the value before attempting a conditional operation:

- If *Exists* is `true`, DynamoDB will check to see if that attribute value already exists in the table. If it is found, then the operation succeeds. If it is not found, the operation fails with a *ConditionalCheckFailedException*.
- If *Exists* is `false`, DynamoDB assumes that the attribute value does not exist in the table. If in fact the value does not exist, then the assumption is valid and the operation succeeds. If the value is found, despite the assumption that it does not exist, the operation fails with a *ConditionalCheckFailedException*.

The default setting for *Exists* is `true`. If you supply a *Value* all by itself, DynamoDB assumes the attribute exists: You don't have to set *Exists* to `true`, because it is implied.

DynamoDB returns a *ValidationException* if:

- *Exists* is `true` but there is no *Value* to check. (You expect a value to exist, but don't specify what that value is.)

- *Exists* is *false* but you also provide a *Value* . (You cannot expect an attribute to have a value, while also expecting it not to exist.)
- **ComparisonOperator** (*string*) –

A comparator for evaluating attributes in the *Attribute-ValueList* . For example, equals, greater than, less than, etc.

The following comparison operators are available:

```
EQ | NE | LE | LT | GE | GT |
NOT_NULL | NULL | CONTAINS |
NOT_CONTAINS | BEGINS_WITH | IN |
BETWEEN
```

The following are descriptions of each comparison operator.

- **EQ** : Equal. **EQ** is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] } .
- **NE** : Not equal. **NE** is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] } .
- **LE** : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- **LT** : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- **GE** : Greater than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not

match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.

- **GT** : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
- **NOT_NULL** : The attribute exists. **NOT_NULL** is supported for all datatypes, including lists and maps. .. note::This operator tests for the existence of an attribute, not its data type. If the data type of attribute “a” is null, and you evaluate it using **NOT_NULL**, the result is a Boolean *true*. This result is because the attribute “a” exists; its data type is not relevant to the **NOT_NULL** comparison operator.
- **NULL** : The attribute does not exist. **NULL** is supported for all datatypes, including lists and maps. .. note::This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute “a” is null, and you evaluate it using **NULL**, the result is a Boolean *false*. This is because the attribute “a” exists; its data type is not relevant to the **NULL** comparison operator.
- **CONTAINS** : Checks for a subsequence, or value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS”, “NS”, or “BS”), then the operator evaluates to true if it finds an exact match with any member of the set. **CONTAINS** is supported for lists: When evaluating “a **CONTAINS** b”, “a” can be a list; however, “b” cannot be a set, a map, or a list.
- **NOT_CONTAINS** : Checks for absence of a subsequence, or absence of a value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS”, “NS”, or “BS”), then the operator evaluates to true if it *does not* find an exact match with any member of the set. **NOT_CONTAINS** is supported for lists: When evaluating “a **NOT CONTAINS** b”, “a” can be a list;

however, “b” cannot be a set, a map, or a list.

- **BEGINS_WITH** : Checks for a prefix. *AttributeValueList* can contain only one *AttributeValue* of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).
- **IN** : Checks for matching elements within two sets. *AttributeValueList* can contain one or more *AttributeValue* elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
- **BETWEEN** : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not compare to { "N" : "6" } . Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }
- **AttributeValueList** (*list*) –

One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *ComparisonOperator* being used.

For type Number, value comparisons are numeric.

String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A , and a is greater than B . For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters .

For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.

For information on specifying data types in JSON, see [JSON Data Format](#) in the *Amazon DynamoDB Developer Guide* .

- (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).

- **ConditionalOperator** (*string*) –

Warning: This is a legacy parameter, for backward compatibility. New applications should use *ConditionExpression* instead. To combine legacy parameters and expression parameters in a call; otherwise, DynamoDB will return a *ValidationException*.

A logical operator to apply to the conditions in the *Expected* map:

- AND - If all of the conditions evaluate to true, then the entire map evaluates to true.
- OR - If at least one of the conditions evaluate to true, then the entire map evaluates to true.

If you omit *ConditionalOperator* , then AND is the default.

The operation will succeed only if the entire map evaluates to true.

Note: This parameter does not support attributes of type List or Map.

- **ReturnValues** (*string*) – Use *ReturnValues* if you want to get the item attributes as they appeared before they were deleted. For *DeleteItem* , the valid values are:
 - NONE - If *ReturnValues* is not specified, or if its value is NONE , then nothing is returned. (This setting is the default for *ReturnValues* .)
 - ALL_OLD - The content of the old item is returned.
- **ReturnConsumedCapacity** (*string*) – Determines the level of detail about provisioned throughput consumption that is returned in the response:
 - INDEXES - The response includes the aggregate *ConsumedCapacity* for the operation, together with *ConsumedCapacity* for each table and secondary index that was accessed. Note that some operations, such as *GetItem* and *BatchGetItem* , do not access any indexes at all. In these cases, specifying *INDEXES* will only return *ConsumedCapacity* information for table(s).
 - TOTAL - The response includes only the aggregate *ConsumedCapacity* for the operation.
 - NONE - No *ConsumedCapacity* details are included in the response.
- **ReturnItemCollectionMetrics** (*string*) – Determines whether item collection metrics are returned. If set to *SIZE* , the response includes statistics about item collections, if any, that were modified during the operation are returned in the response. If set to *NONE* (the default), no statistics are returned.
- **ConditionExpression** (condition from `boto3.dynamodb.conditions.Attr` method) – The condition(s) an attribute(s) must meet. Valid conditions are listed in the *DynamoDB Reference Guide*.
- **ExpressionAttributeNames** (*dict*) – One or more substitution tokens for attribute names in an expression. The following are some use cases for using *ExpressionAttributeNames* :
 - To access an attribute whose name conflicts with a DynamoDB reserved word.
 - To create a placeholder for repeating occurrences of an attribute name in an expression.
 - To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the # character in an expression to dereference an attribute name. For example, consider the following attribute name:

- Percentile

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see *Reserved Words* in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for *ExpressionAttributeNames* :

- {"#P": "Percentile"}

You could then use this substitution in an expression, as in this example:

– #P = :val

Note: Tokens that begin with the `:` character are *expression attribute values*, which are placeholders for the actual value at runtime.

For more information on expression attribute names, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

– (string) –
 * (string) –

- **ExpressionAttributeValues** (dict) – One or more values that can be substituted in an expression.

Use the `:` (colon) character in an expression to dereference an attribute value. For example, suppose that you wanted to check whether the value of the *ProductStatus* attribute was one of the following:

```
Available | Backordered | Discontinued
```

You would first need to specify *ExpressionAttributeValues* as follows:

```
{ ":avail":{"S":"Available"},
  ":back":{"S":"Backordered"},
  ":disc":{"S":"Discontinued"} }
```

You could then use these values in an expression, such as this:

```
ProductStatus IN (:avail, :back, :disc)
```

For more information on expression attribute values, see [Specifying Conditions](#) in the *Amazon DynamoDB Developer Guide*.

– (string) –
 * (valid *DynamoDB* type) – - The value of the attribute. The valid value types are listed in the *DynamoDB Reference Guide*.

Return type dict

Returns

Response Syntax

```
{
  'Attributes': {
    'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])|set
  },
  'ConsumedCapacity': {
    'TableName': 'string',
    'CapacityUnits': 123.0,
    'Table': {
      'CapacityUnits': 123.0
    },
    'LocalSecondaryIndexes': {
      'string': {
        'CapacityUnits': 123.0
      }
    },
    'GlobalSecondaryIndexes': {
      'string': {
        'CapacityUnits': 123.0
      }
    }
  },
}
```

```
'ItemCollectionMetrics': {
  'ItemCollectionKey': {
    'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])
  },
  'SizeEstimateRangeGB': [
    123.0,
  ]
}
```

Response Structure

- (*dict*) –

Represents the output of a *DeleteItem* operation.

- **Attributes** (*dict*) –

A map of attribute names to *AttributeValue* objects, representing the item as it appeared before the *DeleteItem* operation. This map appears in the response only if *ReturnValues* was specified as `ALL_OLD` in the request.

- * (*string*) –

- (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).

- **ConsumedCapacity** (*dict*) –

The capacity units consumed by an operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. *ConsumedCapacity* is only returned if the request asked for it. For more information, see [Provisioned Throughput](#) in the *Amazon DynamoDB Developer Guide*.

- * **TableName** (*string*) –

The name of the table that was affected by the operation.

- * **CapacityUnits** (*float*) –

The total number of capacity units consumed by the operation.

- * **Table** (*dict*) –

The amount of throughput consumed on the table affected by the operation.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

- * **LocalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each local index affected by the operation.

- (*string*) –

- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

* **GlobalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each global index affected by the operation.

- (*string*) –
- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

– **ItemCollectionMetrics** (*dict*) –

Information about item collections, if any, that were affected by the operation. *ItemCollectionMetrics* is only returned if the request asked for it. If the table does not have any local secondary indexes, this information is not returned in the response.

Each *ItemCollectionMetrics* element consists of:

- * *ItemCollectionKey* - The hash key value of the item collection. This is the same as the hash key of the item.
- * *SizeEstimateRange* - An estimate of item collection size, in gigabytes. This value is a two-element array containing a lower bound and an upper bound for the estimate. The estimate includes the size of all the items in the table, plus the size of all attributes projected into all of the local secondary indexes on that table. Use this estimate to measure whether a local secondary index is approaching its size limit. The estimate is subject to change over time; therefore, do not rely on the precision or accuracy of the estimate.

* **ItemCollectionKey** (*dict*) –

The hash key value of the item collection. This value is the same as the hash key of the item.

- (*string*) –
- (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).

* **SizeEstimateRangeGB** (*list*) –

An estimate of item collection size, in gigabytes. This value is a two-element array containing a lower bound and an upper bound for the estimate. The estimate includes the size of all the items in the table, plus the size of all attributes projected into all of the local secondary indexes on that table. Use this estimate to measure whether a local secondary index is approaching its size limit.

The estimate is subject to change over time; therefore, do not rely on the precision or accuracy of the estimate.

- (*float*) –

get_item (**kwargs)

The *GetItem* operation returns a set of attributes for the item with the given primary key. If there is

no matching item, *GetItem* does not return any data.

GetItem provides an eventually consistent read by default. If your application requires a strongly consistent read, set *ConsistentRead* to `true`. Although a strongly consistent read might take more time than an eventually consistent read, it always returns the last updated value.

Request Syntax

```
response = table.get_item(
    Key={
        'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])|set([123])|set(['string'])
    },
    AttributesToGet=[
        'string',
    ],
    ConsistentRead=True|False,
    ReturnConsumedCapacity='INDEXES'|'TOTAL'|'NONE',
    ProjectionExpression='string',
    ExpressionAttributeNames={
        'string': 'string'
    }
)
```

Parameters

- **Key** (*dict*) – [REQUIRED]

A map of attribute names to *AttributeValue* objects, representing the primary key of the item to retrieve.

For the primary key, you must provide all of the attributes. For example, with a hash type primary key, you only need to provide the hash attribute. For a hash-and-range type primary key, you must provide both the hash attribute and the range attribute.

- (*string*) –

- * (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the *DynamoDB Reference Guide*.

- **AttributesToGet** (*list*) –

The names of one or more attributes to retrieve. If no attribute names are provided, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

Note that *AttributesToGet* has no effect on provisioned throughput consumption. DynamoDB determines capacity units consumed based on item size, not on the amount of data that is returned to an application.

- (string) –

- **ConsistentRead** (*boolean*) – Determines the read consistency model: If set to `true`, then the operation uses strongly consistent reads; otherwise, the operation uses eventually consistent reads.

Warning: This is a legacy parameter, for backward compatibility. New applications should use *ProjectionExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception.

This parameter allows you to retrieve attributes of type List or Map, however, it cannot retrieve individual elements within a List or a Map.

- **ReturnConsumedCapacity** (*string*) – Determines the level of detail about provisioned throughput consumption that is returned in the response:
 - *INDEXES* - The response includes the aggregate *ConsumedCapacity* for the operation, together with *ConsumedCapacity* for each table and secondary index that was accessed. Note that some operations, such as *GetItem* and *BatchGetItem*, do not access any indexes at all. In these cases, specifying *INDEXES* will only return *ConsumedCapacity* information for table(s).
 - *TOTAL* - The response includes only the aggregate *ConsumedCapacity* for the operation.
 - *NONE* - No *ConsumedCapacity* details are included in the response.
- **ProjectionExpression** (*string*) – A string that identifies one or more attributes to retrieve from the table. These attributes can include scalars, sets, or elements of a JSON document. The attributes in the expression must be separated by commas.

If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

For more information, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

Note: *ProjectionExpression* replaces the legacy *AttributesToGet* parameter.

- **ExpressionAttributeNames** (*dict*) – One or more substitution tokens for attribute names in an expression. The following are some use cases for using *ExpressionAttributeNames*:
 - To access an attribute whose name conflicts with a DynamoDB reserved word.
 - To create a placeholder for repeating occurrences of an attribute name in an expression.
 - To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the `#` character in an expression to dereference an attribute name. For example, consider the following attribute name:

- `Percentile`

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see [Reserved Words](#) in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for *ExpressionAttributeNames*:

- `{"#P": "Percentile"}`

You could then use this substitution in an expression, as in this example:

- `#P = :val`

Note: Tokens that begin with the `:` character are *expression attribute values*, which are placeholders for the actual value at runtime.

For more information on expression attribute names, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

- (*string*) –
- * (*string*) –

Return type dict
Returns

Response Syntax

```
{
  'Item': {
    'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])|set
  },
  'ConsumedCapacity': {
    'TableName': 'string',
    'CapacityUnits': 123.0,
    'Table': {
      'CapacityUnits': 123.0
    },
    'LocalSecondaryIndexes': {
      'string': {
        'CapacityUnits': 123.0
      }
    },
    'GlobalSecondaryIndexes': {
      'string': {
        'CapacityUnits': 123.0
      }
    }
  }
}
```

Response Structure

- (*dict*) –

Represents the output of a *GetItem* operation.

- **Item** (*dict*) –

A map of attribute names to *AttributeValue* objects, as specified by *AttributesToGet*.

- * (*string*) –

- (*valid DynamoDB type*) – - The value of the attribute.
The valid value types are listed in the *DynamoDB Reference Guide*.

- **ConsumedCapacity** (*dict*) –

The capacity units consumed by an operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. *ConsumedCapacity* is only returned if the request asked for it. For more information, see [Provisioned Throughput](#) in the *Amazon DynamoDB Developer Guide*.

- * **TableName** (*string*) –

The name of the table that was affected by the operation.

- * **CapacityUnits** (*float*) –

The total number of capacity units consumed by the operation.

- * **Table** (*dict*) –

The amount of throughput consumed on the table affected by the operation.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

* **LocalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each local index affected by the operation.

- (*string*) –
- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

* **GlobalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each global index affected by the operation.

- (*string*) –
- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

load()

Calls `dynamodb.Client.describe_table()` to update the attributes of the Table resource

Request Syntax

```
table.load()
```

Returns None

put_item(kwargs)**

Creates a new item, or replaces an old item with a new item. If an item that has the same primary key as the new item already exists in the specified table, the new item completely replaces the existing item. You can perform a conditional put operation (add a new item if one with the specified primary key doesn't exist), or replace an existing item if it has certain attribute values.

In addition to putting an item, you can also return the item's attribute values in the same operation, using the *ReturnValues* parameter.

When you add an item, the primary key attribute(s) are the only required attributes. Attribute values cannot be null. String and Binary type attributes must have lengths greater than zero. Set type attributes cannot be empty. Requests with empty values will be rejected with a *ValidationException* exception.

You can request that *PutItem* return either a copy of the original item (before the update) or a copy of the updated item (after the update). For more information, see the *ReturnValues* description below.

Note: To prevent a new item from replacing an existing item, use a conditional put operation with *ComparisonOperator* set to `NULL` for the primary key attribute, or attributes.

For more information about using this API, see [Working with Items](#) in the *Amazon DynamoDB Developer Guide*.

Request Syntax

```
response = table.put_item(
    Item={
        'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])|set([123])|se
    },
    Expected={
        'string': {
            'Value': 'string'|123|Binary(b'bytes')|True|None|set(['string'])|set([123])
            'Exists': True|False,
            'ComparisonOperator': 'EQ'|'NE'|'IN'|'LE'|'LT'|'GE'|'GT'|'BETWEEN'|'NOT_NULL'
            'AttributeValueList': [
                'string'|123|Binary(b'bytes')|True|None|set(['string'])|set([123])|set
            ]
        }
    },
    ReturnValues='NONE'|'ALL_OLD'|'UPDATED_OLD'|'ALL_NEW'|'UPDATED_NEW',
    ReturnConsumedCapacity='INDEXES'|'TOTAL'|'NONE',
    ReturnItemCollectionMetrics='SIZE'|'NONE',
    ConditionalOperator='AND'|'OR',
    ConditionExpression=Attr('myattribute').eq('myvalue'),
    ExpressionAttributeNames={
        'string': 'string'
    },
    ExpressionAttributeValues={
        'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])|set([123])|se
    }
)
```

Parameters

- **Item (*dict*) – [REQUIRED]**

A map of attribute name/value pairs, one for each attribute. Only the primary key attributes are required; you can optionally provide other attribute name-value pairs for the item.

You must provide all of the attributes for the primary key. For example, with a hash type primary key, you only need to provide the hash attribute. For a hash-and-range type primary key, you must provide both the hash attribute and the range attribute.

If you specify any attributes that are part of an index key, then the data types for those attributes must match those of the schema in the table's attribute definition.

For more information about primary keys, see [Primary Key](#) in the *Amazon DynamoDB Developer Guide*.

Each element in the *Item* map is an *AttributeValue* object.

- (string) -
 - * (valid DynamoDB type) - - The value of the attribute. The valid value types are listed in the *DynamoDB Reference Guide*.

- **Expected** (*dict*) –

Warning: This is a legacy parameter, for backward compatibility. New applications should use *ConditionExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception.

A map of attribute/condition pairs. *Expected* provides a conditional block for the *PutItem* operation.

Note: This parameter does not support attributes of type List or Map.

Each element of *Expected* consists of an attribute name, a comparison operator, and one or more values. DynamoDB compares the attribute with the value(s) you supplied, using the comparison operator. For each *Expected* element, the result of the evaluation is either true or false.

If you specify more than one element in the *Expected* map, then by default all of the conditions must evaluate to true. In other words, the conditions are ANDed together. (You can use the *ConditionalOperator* parameter to OR the conditions instead. If you do this, then at least one of the conditions must evaluate to true, rather than all of them.)

If the *Expected* map evaluates to true, then the conditional operation succeeds; otherwise, it fails.

Expected contains the following:

- *AttributeValueList* - One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *ComparisonOperator* being used. For type Number, value comparisons are numeric. String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A , and a is greater than B . For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters . For type Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.
- *ComparisonOperator* - A comparator for evaluating attributes in the *AttributeValueList* . When performing the comparison, DynamoDB uses strongly consistent reads. The following comparison operators are available: EQ | NE | LE | LT | GE | GT | NOT_NULL | NULL | CONTAINS | NOT_CONTAINS | BEGINS_WITH | IN | BETWEEN The following are descriptions of each comparison operator. * EQ : Equal. EQ is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] } .
- NE : Not equal. NE is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* of a different type than the one provided in the request, the value does not match. For example,

- `{"S": "6"}` does not equal `{"N": "6"}` . Also, `{"N": "6"}` does not equal `{"NS": ["6", "2", "1"]}` .
- **LE** : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}` . Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}` .
 - **LT** : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}` . Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}` .
 - **GE** : Greater than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}` . Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}` .
 - **GT** : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}` . Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}` .
 - **NOT_NULL** : The attribute exists. **NOT_NULL** is supported for all datatypes, including lists and maps. .. note::This operator tests for the existence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using **NOT_NULL** , the result is a Boolean *true* . This result is because the attribute “a ” exists; its data type is not relevant to the **NOT_NULL** comparison operator.
 - **NULL** : The attribute does not exist. **NULL** is supported for all datatypes, including lists and maps. .. note::This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using **NULL** , the result is a Boolean *false* . This is because the attribute “a ” exists; its data type is not relevant to the **NULL** comparison operator.
 - **CONTAINS** : Checks for a subsequence, or value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS ”, “NS ”, or “BS ”), then the operator evaluates to true if it finds an exact match with any member of the set. **CONTAINS** is supported for lists: When evaluating “a **CONTAINS** b ”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
 - **NOT_CONTAINS** : Checks for absence of a subsequence, or ab-

sence of a value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set ("SS ", "NS ", or "BS "), then the operator evaluates to true if it *does not* find an exact match with any member of the set. NOT_CONTAINS is supported for lists: When evaluating "a NOT CONTAINS b", "a " can be a list; however, "b " cannot be a set, a map, or a list.

- **BEGINS_WITH** : Checks for a prefix. *AttributeValueList* can contain only one *AttributeValue* of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).
- **IN** : Checks for matching elements within two sets. *AttributeValueList* can contain one or more *AttributeValue* elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
- **BETWEEN** : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, {"S": "6"} does not compare to {"N": "6"} . Also, {"N": "6"} does not compare to {"NS": ["6", "2", "1"]}

For usage examples of *AttributeValueList* and *ComparisonOperator* , see [Legacy Conditional Parameters](#) in the *Amazon DynamoDB Developer Guide* .

For backward compatibility with previous DynamoDB releases, the following parameters can be used instead of *AttributeValueList* and *ComparisonOperator* :

- *Value* - A value for DynamoDB to compare with an attribute.
- *Exists* - A Boolean value that causes DynamoDB to evaluate the value before attempting the conditional operation: * If *Exists* is `true` , DynamoDB will check to see if that attribute value already exists in the table. If it is found, then the condition evaluates to true; otherwise the condition evaluate to false.
- If *Exists* is `false` , DynamoDB assumes that the attribute value does *not* exist in the table. If in fact the value does not exist, then the assumption is valid and the condition evaluates to true. If the value is found, despite the assumption that it does not exist, the condition evaluates to false.

Note that the default value for *Exists* is `true` .

The *Value* and *Exists* parameters are incompatible with *AttributeValueList* and *ComparisonOperator* . Note that if you use both sets of parameters at once, DynamoDB will return a *ValidationException* exception.

- (string) -

* (*dict*) –

Represents a condition to be compared with an attribute value. This condition can be used with *DeleteItem* , *PutItem* or *UpdateItem* operations; if the comparison evaluates to true, the operation succeeds; if not, the operation fails. You can use *ExpectedAttributeValue* in one of two different ways:

- Use *AttributeValueList* to specify one or more values to compare against an attribute. Use *ComparisonOperator* to specify how you want to perform the comparison. If the comparison evaluates to true, then the conditional operation succeeds.
- Use *Value* to specify a value that DynamoDB will compare against an attribute. If the values match, then *ExpectedAttributeValue* evaluates to true and the conditional operation succeeds. Optionally, you can also set *Exists* to false, indicating that you *do not* expect to find the attribute value in the table. In this case, the conditional operation succeeds only if the comparison evaluates to false.

Value and *Exists* are incompatible with *AttributeValueList* and *ComparisonOperator* . Note that if you use both sets of parameters at once, DynamoDB will return a *ValidationException* exception.

- **Value** (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).
- **Exists** (*boolean*) –

Causes DynamoDB to evaluate the value before attempting a conditional operation:

- If *Exists* is `true` , DynamoDB will check to see if that attribute value already exists in the table. If it is found, then the operation succeeds. If it is not found, the operation fails with a *ConditionalCheckFailedException* .
- If *Exists* is `false` , DynamoDB assumes that the attribute value does not exist in the table. If in fact the value does not exist, then the assumption is valid and the operation succeeds. If the value is found, despite the assumption that it does not exist, the operation fails with a *ConditionalCheckFailedException* .

The default setting for *Exists* is `true` . If you supply a *Value* all by itself, DynamoDB assumes the attribute exists: You don't have to set *Exists* to `true` , because it is implied.

DynamoDB returns a *ValidationException* if:

- *Exists* is `true` but there is no *Value* to check. (You expect a value to exist, but don't specify what that value is.)
- *Exists* is `false` but you also provide a *Value* . (You cannot expect an attribute to have a value, while also expecting it not to exist.)
- **ComparisonOperator** (*string*) –

A comparator for evaluating attributes in the *AttributeValueList*. For example, equals, greater than, less than, etc.

The following comparison operators are available:

```
EQ | NE | LE | LT | GE | GT |
NOT_NULL | NULL | CONTAINS |
NOT_CONTAINS | BEGINS_WITH | IN |
BETWEEN
```

The following are descriptions of each comparison operator.

- **EQ** : Equal. **EQ** is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" }. Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] }.
- **NE** : Not equal. **NE** is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" }. Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] }.
- **LE** : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" }. Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] }.
- **LT** : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" }. Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] }.
- **GE** : Greater than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" }. Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] }.
- **GT** : Greater than. *AttributeValueList* can contain only

one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" } . Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] } .

- NOT_NULL : The attribute exists. NOT_NULL is supported for all datatypes, including lists and maps. .. note::This operator tests for the existence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using NOT_NULL , the result is a Boolean *true* . This result is because the attribute “a ” exists; its data type is not relevant to the NOT_NULL comparison operator.
- NULL : The attribute does not exist. NULL is supported for all datatypes, including lists and maps. .. note::This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using NULL , the result is a Boolean *false* . This is because the attribute “a ” exists; its data type is not relevant to the NULL comparison operator.
- CONTAINS : Checks for a subsequence, or value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS ”, “NS ”, or “BS ”), then the operator evaluates to true if it finds an exact match with any member of the set. CONTAINS is supported for lists: When evaluating “a CONTAINS b ”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
- NOT_CONTAINS : Checks for absence of a subsequence, or absence of a value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS ”, “NS ”, or “BS ”), then the operator evaluates to true if it *does not* find an exact match with any member of the set. NOT_CONTAINS is supported for lists: When evaluating “a NOT CONTAINS b ”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
- BEGINS_WITH : Checks for a prefix. *AttributeValueList* can contain only one *AttributeValue* of type String or Binary (not a Number or a set type). The tar-

get attribute of the comparison must be of type String or Binary (not a Number or a set type).

- **IN** : Checks for matching elements within two sets. *AttributeValueList* can contain one or more *AttributeValue* elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
- **BETWEEN** : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not compare to { "N" : "6" } . Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }
- **AttributeValueList** (*list*) –

One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *ComparisonOperator* being used.

For type Number, value comparisons are numeric.

String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A , and a is greater than B . For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters .

For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.

For information on specifying data types in JSON, see [JSON Data Format](#) in the *Amazon DynamoDB Developer Guide* .

- (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the *DynamoDB Reference Guide*.
- **ReturnValues** (*string*) – Use *ReturnValues* if you want to get the item attributes as they appeared before they were updated with the *PutItem* request. For *PutItem* , the valid values are:
 - NONE - If *ReturnValues* is not specified, or if its value is NONE , then nothing is returned. (This setting is the default for *ReturnValues* .)
 - ALL_OLD - If *PutItem* overwrote an attribute name-value pair, then the content of the old item is returned.

Note: Other “Valid Values” are not relevant to *PutItem*.

- **ReturnConsumedCapacity** (*string*) – Determines the level of detail about provisioned throughput consumption that is returned in the response:

- *INDEXES* - The response includes the aggregate *ConsumedCapacity* for the operation, together with *ConsumedCapacity* for each table and secondary index that was accessed. Note that some operations, such as *GetItem* and *BatchGetItem*, do not access any indexes at all. In these cases, specifying *INDEXES* will only return *ConsumedCapacity* information for table(s).
- *TOTAL* - The response includes only the aggregate *ConsumedCapacity* for the operation.
- *NONE* - No *ConsumedCapacity* details are included in the response.
- **ReturnItemCollectionMetrics** (*string*) – Determines whether item collection metrics are returned. If set to *SIZE*, the response includes statistics about item collections, if any, that were modified during the operation are returned in the response. If set to *NONE* (the default), no statistics are returned.

Warning: This is a legacy parameter, for backward compatibility. New applications should use *ConditionExpression* instead. To combine legacy parameters and expression parameters in a single call; otherwise, DynamoDB will return a *ValidationException*.

- **ConditionalOperator** (*string*) –

A logical operator to apply to the conditions in the *Expected* map:

- *AND* - If all of the conditions evaluate to true, then the entire map evaluates to true.
- *OR* - If at least one of the conditions evaluate to true, then the entire map evaluates to true.

If you omit *ConditionalOperator*, then *AND* is the default.

The operation will succeed only if the entire map evaluates to true.

Note: This parameter does not support attributes of type List or Map.

- **ConditionExpression** (condition from `boto3.dynamodb.conditions.Attr` method) – The condition(s) an attribute(s) must meet. Valid conditions are listed in the *DynamoDB Reference Guide*.
- **ExpressionAttributeNames** (*dict*) – One or more substitution tokens for attribute names in an expression. The following are some use cases for using *ExpressionAttributeNames*:
 - To access an attribute whose name conflicts with a DynamoDB reserved word.
 - To create a placeholder for repeating occurrences of an attribute name in an expression.
 - To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the *#* character in an expression to dereference an attribute name. For example, consider the following attribute name:

- Percentile

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see *Reserved Words* in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for *ExpressionAttributeNames*:

- {"#P": "Percentile"}

You could then use this substitution in an expression, as in this example:

– #P = :val

Note: Tokens that begin with the `:` character are *expression attribute values*, which are placeholders for the actual value at runtime.

For more information on expression attribute names, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

– (string) –
 * (string) –

- **ExpressionAttributeValues** (*dict*) – One or more values that can be substituted in an expression.

Use the `:` (colon) character in an expression to dereference an attribute value. For example, suppose that you wanted to check whether the value of the *ProductStatus* attribute was one of the following:

```
Available | Backordered | Discontinued
```

You would first need to specify *ExpressionAttributeValues* as follows:

```
{ ":avail":{"S":"Available"},
  ":back":{"S":"Backordered"},
  ":disc":{"S":"Discontinued"} }
```

You could then use these values in an expression, such as this:

```
ProductStatus IN (:avail, :back, :disc)
```

For more information on expression attribute values, see [Specifying Conditions](#) in the *Amazon DynamoDB Developer Guide*.

– (string) –
 * (valid *DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).

Return type dict

Returns

Response Syntax

```
{
  'Attributes': {
    'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])|set
  },
  'ConsumedCapacity': {
    'TableName': 'string',
    'CapacityUnits': 123.0,
    'Table': {
      'CapacityUnits': 123.0
    },
    'LocalSecondaryIndexes': {
      'string': {
        'CapacityUnits': 123.0
      }
    },
    'GlobalSecondaryIndexes': {
      'string': {
        'CapacityUnits': 123.0
      }
    }
  },
}
```

```
'ItemCollectionMetrics': {
    'ItemCollectionKey': {
        'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])
    },
    'SizeEstimateRangeGB': [
        123.0,
    ]
}
```

Response Structure

- (*dict*) –

Represents the output of a *PutItem* operation.

- **Attributes** (*dict*) –

The attribute values as they appeared before the *PutItem* operation, but only if *ReturnValues* is specified as `ALL_OLD` in the request. Each element consists of an attribute name and an attribute value.

- * (*string*) –
 - (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).

- **ConsumedCapacity** (*dict*) –

The capacity units consumed by an operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. *ConsumedCapacity* is only returned if the request asked for it. For more information, see [Provisioned Throughput](#) in the *Amazon DynamoDB Developer Guide*.

- * **TableName** (*string*) –

The name of the table that was affected by the operation.

- * **CapacityUnits** (*float*) –

The total number of capacity units consumed by the operation.

- * **Table** (*dict*) –

The amount of throughput consumed on the table affected by the operation.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

- * **LocalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each local index affected by the operation.

- (*string*) –
- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

* **GlobalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each global index affected by the operation.

- (*string*) –
- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

– **ItemCollectionMetrics** (*dict*) –

Information about item collections, if any, that were affected by the operation. *ItemCollectionMetrics* is only returned if the request asked for it. If the table does not have any local secondary indexes, this information is not returned in the response.

Each *ItemCollectionMetrics* element consists of:

- * **ItemCollectionKey** - The hash key value of the item collection. This is the same as the hash key of the item.
- * **SizeEstimateRange** - An estimate of item collection size, in gigabytes. This value is a two-element array containing a lower bound and an upper bound for the estimate. The estimate includes the size of all the items in the table, plus the size of all attributes projected into all of the local secondary indexes on that table. Use this estimate to measure whether a local secondary index is approaching its size limit. The estimate is subject to change over time; therefore, do not rely on the precision or accuracy of the estimate.

* **ItemCollectionKey** (*dict*) –

The hash key value of the item collection. This value is the same as the hash key of the item.

- (*string*) –
- (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).

* **SizeEstimateRangeGB** (*list*) –

An estimate of item collection size, in gigabytes. This value is a two-element array containing a lower bound and an upper bound for the estimate. The estimate includes the size of all the items in the table, plus the size of all attributes projected into all of the local secondary indexes on that table. Use this estimate to measure whether a local secondary index is approaching its size limit.

The estimate is subject to change over time; therefore, do not rely on the precision or accuracy of the estimate.

- (*float*) –

query (***kwargs*)

A *Query* operation uses the primary key of a table or a secondary index to directly access items from that table or index.

Use the *KeyConditionExpression* parameter to provide a specific hash key value. The *Query* operation will return all of the items from the table or index with that hash key value. You can optionally narrow the scope of the *Query* operation by specifying a range key value and a comparison operator in *KeyConditionExpression*. You can use the *ScanIndexForward* parameter to get results in forward or reverse order, by range key or by index key.

Queries that do not return results consume the minimum number of read capacity units for that type of read operation.

If the total number of items meeting the query criteria exceeds the result set size limit of 1 MB, the query stops and results are returned to the user with the *LastEvaluatedKey* element to continue the query in a subsequent operation. Unlike a *Scan* operation, a *Query* operation never returns both an empty result set and a *LastEvaluatedKey* value. *LastEvaluatedKey* is only provided if the results exceed 1 MB, or if you have used the *Limit* parameter.

You can query a table, a local secondary index, or a global secondary index. For a query on a table or on a local secondary index, you can set the *ConsistentRead* parameter to `true` and obtain a strongly consistent result. Global secondary indexes support eventually consistent reads only, so do not specify *ConsistentRead* when querying a global secondary index.

Request Syntax

```
response = table.query(
    IndexName='string',
    Select='ALL_ATTRIBUTES' | 'ALL_PROJECTED_ATTRIBUTES' | 'SPECIFIC_ATTRIBUTES' | 'COUNT',
    AttributesToGet=[
        'string',
    ],
    Limit=123,
    ConsistentRead=True|False,
    KeyConditions={
        'string': {
            'AttributeValueList': [
                'string' | 123 | Binary(b'bytes') | True|None|set(['string']) | set([123]) | set
            ],
            'ComparisonOperator': 'EQ' | 'NE' | 'IN' | 'LE' | 'LT' | 'GE' | 'GT' | 'BETWEEN' | 'NOT_NULL'
        }
    },
    QueryFilter={
        'string': {
            'AttributeValueList': [
                'string' | 123 | Binary(b'bytes') | True|None|set(['string']) | set([123]) | set
            ],
            'ComparisonOperator': 'EQ' | 'NE' | 'IN' | 'LE' | 'LT' | 'GE' | 'GT' | 'BETWEEN' | 'NOT_NULL'
        }
    },
    ConditionalOperator='AND' | 'OR',
    ScanIndexForward=True|False,
    ExclusiveStartKey={
        'string': 'string' | 123 | Binary(b'bytes') | True|None|set(['string']) | set([123]) | set
    },
    ReturnConsumedCapacity='INDEXES' | 'TOTAL' | 'NONE',
    ProjectionExpression='string',
    FilterExpression=Attr('myattribute').eq('myvalue'),
    KeyConditionExpression=Key('mykey').eq('myvalue'),
    ExpressionAttributeNames={
        'string': 'string'
    },
    ExpressionAttributeValues={
```



```

    'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])|set([123])|se
  }
)

```

Parameters

- **IndexName** (*string*) – The name of an index to query. This index can be any local secondary index or global secondary index on the table. Note that if you use the *IndexName* parameter, you must also provide *TableName*.
- **Select** (*string*) – The attributes to be returned in the result. You can retrieve all item attributes, specific item attributes, the count of matching items, or in the case of an index, some or all of the attributes projected into the index.
 - **ALL_ATTRIBUTES** - Returns all of the item attributes from the specified table or index. If you query a local secondary index, then for each matching item in the index DynamoDB will fetch the entire item from the parent table. If the index is configured to project all item attributes, then all of the data can be obtained from the local secondary index, and no fetching is required.
 - **ALL_PROJECTED_ATTRIBUTES** - Allowed only when querying an index. Retrieves all attributes that have been projected into the index. If the index is configured to project all attributes, this return value is equivalent to specifying **ALL_ATTRIBUTES**.
 - **COUNT** - Returns the number of matching items, rather than the matching items themselves.
 - **SPECIFIC_ATTRIBUTES** - Returns only the attributes listed in *AttributesToGet*. This return value is equivalent to specifying *AttributesToGet* without specifying any value for *Select*. If you query a local secondary index and request only attributes that are projected into that index, the operation will read only the index and not the table. If any of the requested attributes are not projected into the local secondary index, DynamoDB will fetch each of these attributes from the parent table. This extra fetching incurs additional throughput cost and latency. If you query a global secondary index, you can only request attributes that are projected into the index. Global secondary index queries cannot fetch attributes from the parent table.

If neither *Select* nor *AttributesToGet* are specified, DynamoDB defaults to **ALL_ATTRIBUTES** when accessing a table, and **ALL_PROJECTED_ATTRIBUTES** when accessing an index. You cannot use both *Select* and *AttributesToGet* together in a single request, unless the value for *Select* is **SPECIFIC_ATTRIBUTES**. (This usage is equivalent to specifying *AttributesToGet* without any value for *Select*.)

Note: If you use the *ProjectionExpression* parameter, then the value for *Select* can only be **SPECIFIC_ATTRIBUTES**. Any other value for *Select* will return an error.

- **AttributesToGet** (*list*) –

Warning: This is a legacy parameter, for backward compatibility. New applications should use *ProjectionExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception.

This parameter allows you to retrieve attributes of type List or Map; however, it cannot retrieve individual elements within a List or a Map.

The names of one or more attributes to retrieve. If no attribute names are provided, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

Note that *AttributesToGet* has no effect on provisioned throughput consumption. DynamoDB determines capacity units consumed based on item size, not on the amount of data that is returned to an application.

You cannot use both *AttributesToGet* and *Select* together in a *Query* request, *unless* the value for *Select* is `SPECIFIC_ATTRIBUTES`. (This usage is equivalent to specifying *AttributesToGet* without any value for *Select*.)

If you query a local secondary index and request only attributes that are projected into that index, the operation will read only the index and not the table. If any of the requested attributes are not projected into the local secondary index, DynamoDB will fetch each of these attributes from the parent table. This extra fetching incurs additional throughput cost and latency.

If you query a global secondary index, you can only request attributes that are projected into the index. Global secondary index queries cannot fetch attributes from the parent table.

– (string) –

- **Limit** (*integer*) – The maximum number of items to evaluate (not necessarily the number of matching items). If DynamoDB processes the number of items up to the limit while processing the results, it stops the operation and returns the matching values up to that point, and a key in *LastEvaluatedKey* to apply in a subsequent operation, so that you can pick up where you left off. Also, if the processed data set size exceeds 1 MB before DynamoDB reaches this limit, it stops the operation and returns the matching values up to the limit, and a key in *LastEvaluatedKey* to apply in a subsequent operation to continue the operation. For more information, see [Query and Scan](#) in the *Amazon DynamoDB Developer Guide*.
- **ConsistentRead** (*boolean*) – Determines the read consistency model: If set to `true`, then the operation uses strongly consistent reads; otherwise, the operation uses eventually consistent reads.

Strongly consistent reads are not supported on global secondary indexes. If you query a global secondary index with *ConsistentRead* set to `true`, you will receive a *ValidationException*.

- **KeyConditions** (*dict*) –

Warning: This is a legacy parameter, for backward compatibility. New applications should use *KeyConditionExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception.

The selection criteria for the query. For a query on a table, you can have conditions only on the table primary key attributes. You must provide the hash key attribute name and value as an EQ condition. You can optionally provide a second condition, referring to the range key attribute.

Note: If you don't provide a range key condition, all of the items that match the hash key will be retrieved. If a *FilterExpression* or *QueryFilter* is present, it will be applied after the items are retrieved.

For a query on an index, you can have conditions only on the index key attributes. You must provide the index hash attribute name and value as an EQ condition. You can optionally provide a second condition, referring to the index key range attribute.

Each *KeyConditions* element consists of an attribute name to compare, along with the following:

- *AttributeValueList* - One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *ComparisonOperator* being used. For type Number, value comparisons are numeric. String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A, and a is greater than B. For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters. For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.
- *ComparisonOperator* - A comparator for evaluating attributes, for example, equals, greater than, less than, and so on. For *KeyConditions*, only the following comparison operators are supported: EQ | LE | LT | GE | GT | BEGINS_WITH | BETWEEN. The following are descriptions of these comparison operators.
 - * EQ : Equal. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one specified in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" }. Also, { "N" : "6" } does not equal { "NS" : ["6", "2", "1"] }.
 - LE : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" }. Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }.
 - LT : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" }. Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }.
 - GE : Greater than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" }. Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }.
 - GT : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" }. Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }.

-
- **BEGINS_WITH** : Checks for a prefix. *AttributeValueList* can contain only one *AttributeValue* of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).
- **BETWEEN** : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not compare to { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] }

For usage examples of *AttributeValueList* and *ComparisonOperator* , see [Legacy Conditional Parameters](#) in the *Amazon DynamoDB Developer Guide* .

- (string) –
- * (dict) –

Represents the selection criteria for a *Query* or *Scan* operation:

- For a *Query* operation, *Condition* is used for specifying the *KeyConditions* to use when querying a table or an index. For *KeyConditions* , only the following comparison operators are supported: EQ | LE | LT | GE | GT | **BEGINS_WITH** | **BETWEEN** *Condition* is also used in a *QueryFilter* , which evaluates the query results and returns only the desired values.
- For a *Scan* operation, *Condition* is used in a *ScanFilter* , which evaluates the scan results and returns only the desired values.

- **AttributeValueList** (list) –

One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *ComparisonOperator* being used.

For type Number, value comparisons are numeric.

String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A , and a is greater than B . For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters .

For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.

- (valid *DynamoDB* type) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).
- **ComparisonOperator** (string) – **[REQUIRED]**

A comparator for evaluating attributes. For example,

equals, greater than, less than, etc.

The following comparison operators are available:

```
EQ | NE | LE | LT | GE | GT |
NOT_NULL | NULL | CONTAINS |
NOT_CONTAINS | BEGINS_WITH | IN |
BETWEEN
```

The following are descriptions of each comparison operator.

- **EQ** : Equal. **EQ** is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] } .
- **NE** : Not equal. **NE** is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] } .
- **LE** : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- **LT** : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- **GE** : Greater than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- **GT** : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *Attribute-*

Value element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .

- **NOT_NULL** : The attribute exists. NOT_NULL is supported for all datatypes, including lists and maps. .. note::This operator tests for the existence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using NOT_NULL , the result is a Boolean *true* . This result is because the attribute “a ” exists; its data type is not relevant to the NOT_NULL comparison operator.
- **NULL** : The attribute does not exist. NULL is supported for all datatypes, including lists and maps. .. note::This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using NULL , the result is a Boolean *false* . This is because the attribute “a ” exists; its data type is not relevant to the NULL comparison operator.
- **CONTAINS** : Checks for a subsequence, or value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS ”, “NS ”, or “BS ”), then the operator evaluates to true if it finds an exact match with any member of the set. CONTAINS is supported for lists: When evaluating “a CONTAINS b ”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
- **NOT_CONTAINS** : Checks for absence of a subsequence, or absence of a value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS ”, “NS ”, or “BS ”), then the operator evaluates to true if it *does not* find an exact match with any member of the set. NOT_CONTAINS is supported for lists: When evaluating “a NOT CONTAINS b ”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
- **BEGINS_WITH** : Checks for a prefix. *AttributeValueList* can contain only one *AttributeValue* of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).

- **IN** : Checks for matching elements within two sets. *AttributeValueList* can contain one or more *AttributeValue* elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
- **BETWEEN** : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not compare to { "N" : "6" } . Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }

For usage examples of *AttributeValueList* and *ComparisonOperator* , see [Legacy Conditional Parameters](#) in the *Amazon DynamoDB Developer Guide* .

• **QueryFilter** (*dict*) –

Warning: This is a legacy parameter, for backward compatibility. New applications should use *FilterExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception.

A condition that evaluates the query results after the items are read and returns only the desired values.

This parameter does not support attributes of type List or Map.

Note: A *QueryFilter* is applied after the items have already been read; the process of filtering does not consume any additional read capacity units.

If you provide more than one condition in the *QueryFilter* map, then by default all of the conditions must evaluate to true. In other words, the conditions are ANDed together. (You can use the *ConditionalOperator* parameter to OR the conditions instead. If you do this, then at least one of the conditions must evaluate to true, rather than all of them.)

Note that *QueryFilter* does not allow key attributes. You cannot define a filter condition on a hash key or range key.

Each *QueryFilter* element consists of an attribute name to compare, along with the following:

- *AttributeValueList* - One or more values to evaluate against the supplied attribute. The number of values in the list depends on the operator specified in *ComparisonOperator* . For type Number, value comparisons are numeric. String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A , and a is greater than B . For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters .

For type Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values. For information on specifying data types in JSON, see [JSON Data Format](#) in the *Amazon DynamoDB Developer Guide*.

- *ComparisonOperator* - A comparator for evaluating attributes. For example, equals, greater than, less than, etc. The following comparison operators are available: EQ | NE | LE | LT | GE | GT | NOT_NULL | NULL | CONTAINS | NOT_CONTAINS | BEGINS_WITH | IN | BETWEEN. For complete descriptions of all comparison operators, see the [Condition](#) data type.

- *(string)* –

- * *(dict)* –

Represents the selection criteria for a *Query* or *Scan* operation:

- For a *Query* operation, *Condition* is used for specifying the *KeyConditions* to use when querying a table or an index. For *KeyConditions*, only the following comparison operators are supported: EQ | LE | LT | GE | GT | BEGINS_WITH | BETWEEN. *Condition* is also used in a *QueryFilter*, which evaluates the query results and returns only the desired values.
- For a *Scan* operation, *Condition* is used in a *ScanFilter*, which evaluates the scan results and returns only the desired values.

- **AttributeValueList** (*list*) –

One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *ComparisonOperator* being used.

For type Number, value comparisons are numeric.

String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A, and a is greater than B. For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters.

For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.

- *(valid DynamoDB type)* – - The value of the attribute. The valid value types are listed in the *DynamoDB Reference Guide*.
- **ComparisonOperator** (*string*) – **[REQUIRED]**

A comparator for evaluating attributes. For example, equals, greater than, less than, etc.

The following comparison operators are available:

EQ | NE | LE | LT | GE | GT |
NOT_NULL | NULL | CONTAINS |
NOT_CONTAINS | BEGINS_WITH | IN |
BETWEEN

The following are descriptions of each comparison operator.

- EQ : Equal. EQ is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] } .
- NE : Not equal. NE is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] } .
- LE : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- LT : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- GE : Greater than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- GT : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- NOT_NULL : The attribute exists. NOT_NULL is supported for all datatypes, including lists and maps. .. note::This operator tests for the existence of an at-

tribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using `NOT_NULL` , the result is a Boolean *true* . This result is because the attribute “a ” exists; its data type is not relevant to the `NOT_NULL` comparison operator.

- `NULL` : The attribute does not exist. `NULL` is supported for all datatypes, including lists and maps. .. note::This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using `NULL` , the result is a Boolean *false* . This is because the attribute “a ” exists; its data type is not relevant to the `NULL` comparison operator.
- `CONTAINS` : Checks for a subsequence, or value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS ”, “NS ”, or “BS ”), then the operator evaluates to true if it finds an exact match with any member of the set. `CONTAINS` is supported for lists: When evaluating “a `CONTAINS` b ”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
- `NOT_CONTAINS` : Checks for absence of a subsequence, or absence of a value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS ”, “NS ”, or “BS ”), then the operator evaluates to true if it *does not* find an exact match with any member of the set. `NOT_CONTAINS` is supported for lists: When evaluating “a `NOT CONTAINS` b ”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
- `BEGINS_WITH` : Checks for a prefix. *AttributeValueList* can contain only one *AttributeValue* of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).
- `IN` : Checks for matching elements within two sets. *AttributeValueList* can contain one or more *AttributeValue* elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
- `BETWEEN` : Greater than or equal to the first value,

and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, `{"S": "6"}` does not compare to `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`

For usage examples of *AttributeValueList* and *ComparisonOperator*, see [Legacy Conditional Parameters](#) in the *Amazon DynamoDB Developer Guide*.

- **ConditionalOperator** (*string*) –

Warning: This is a legacy parameter, for backward compatibility. New applications should use *FilterExpression* instead. Do not combine legacy parameters and expression parameters in a single request; otherwise, DynamoDB will return a *ValidationException*.

A logical operator to apply to the conditions in a *QueryFilter* map:

- AND - If all of the conditions evaluate to true, then the entire map evaluates to true.
- OR - If at least one of the conditions evaluate to true, then the entire map evaluates to true.

If you omit *ConditionalOperator*, then AND is the default.

The operation will succeed only if the entire map evaluates to true.

Note: This parameter does not support attributes of type List or Map.

- **ScanIndexForward** (*boolean*) – Specifies the order in which to return the query results - either ascending (`true`) or descending (`false`).

Items with the same hash key are stored in sorted order by range key. If the range key data type is Number, the results are stored in numeric order. For type String, the results are returned in order of ASCII character code values. For type Binary, DynamoDB treats each byte of the binary data as unsigned.

If *ScanIndexForward* is `true`, DynamoDB returns the results in order, by range key. This is the default behavior.

If *ScanIndexForward* is `false`, DynamoDB sorts the results in descending order by range key, and then returns the results to the client.

- **ExclusiveStartKey** (*dict*) – The primary key of the first item that this operation will evaluate. Use the value that was returned for *LastEvaluatedKey* in the previous operation.

The data type for *ExclusiveStartKey* must be String, Number or Binary. No set data types are allowed.

– (*string*) –

* (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).

- **ReturnConsumedCapacity** (*string*) – Determines the level of detail about

provisioned throughput consumption that is returned in the response:

- *INDEXES* - The response includes the aggregate *ConsumedCapacity* for the operation, together with *ConsumedCapacity* for each table and secondary index that was accessed. Note that some operations, such as *GetItem* and *BatchGetItem*, do not access any indexes at all. In these cases, specifying *INDEXES* will only return *ConsumedCapacity* information for table(s).
- *TOTAL* - The response includes only the aggregate *ConsumedCapacity* for the operation.
- *NONE* - No *ConsumedCapacity* details are included in the response.
- **ProjectionExpression** (*string*) – A string that identifies one or more attributes to retrieve from the table. These attributes can include scalars, sets, or elements of a JSON document. The attributes in the expression must be separated by commas.

If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

For more information, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

Note: *ProjectionExpression* replaces the legacy *AttributesToGet* parameter.

- **FilterExpression** (condition from `boto3.dynamodb.conditions.Attr` method) – The condition(s) an attribute(s) must meet. Valid conditions are listed in the [DynamoDB Reference Guide](#).
- **KeyConditionExpression** (condition from `boto3.dynamodb.conditions.Key` method) – The condition(s) a key(s) must meet. Valid conditions are listed in the [DynamoDB Reference Guide](#).
- **ExpressionAttributeNames** (*dict*) – One or more substitution tokens for attribute names in an expression. The following are some use cases for using *ExpressionAttributeNames*:
 - To access an attribute whose name conflicts with a DynamoDB reserved word.
 - To create a placeholder for repeating occurrences of an attribute name in an expression.
 - To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the `#` character in an expression to dereference an attribute name. For example, consider the following attribute name:

- Percentile

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see [Reserved Words](#) in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for *ExpressionAttributeNames*:

- `{"#P": "Percentile"}`

You could then use this substitution in an expression, as in this example:

- `#P = :val`

Note: Tokens that begin with the `:` character are *expression attribute values*, which are placeholders for the actual value at runtime.

For more information on expression attribute names, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

- (string) –
- * (string) –
- **ExpressionAttributeValues** (dict) – One or more values that can be substituted in an expression.

Use the `:` (colon) character in an expression to dereference an attribute value. For example, suppose that you wanted to check whether the value of the *ProductStatus* attribute was one of the following:

Available | Backordered | Discontinued

You would first need to specify *ExpressionAttributeValues* as follows:

```
{ ":avail":{"S":"Available"},
  ":back":{"S":"Backordered"},
  ":disc":{"S":"Discontinued"} }
```

You could then use these values in an expression, such as this:

```
ProductStatus IN (:avail, :back, :disc)
```

For more information on expression attribute values, see [Specifying Conditions](#) in the *Amazon DynamoDB Developer Guide*.

- (string) –
- * (valid DynamoDB type) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).

Return type dict

Returns

Response Syntax

```
{
  'Items': [
    {
      'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])
    },
  ],
  'Count': 123,
  'ScannedCount': 123,
  'LastEvaluatedKey': {
    'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])|set
  },
  'ConsumedCapacity': {
    'TableName': 'string',
    'CapacityUnits': 123.0,
    'Table': {
      'CapacityUnits': 123.0
    },
    'LocalSecondaryIndexes': {
      'string': {
        'CapacityUnits': 123.0
      }
    },
    'GlobalSecondaryIndexes': {
      'string': {
        'CapacityUnits': 123.0
      }
    }
  }
}
```

```
    }  
  }  
}
```

Response Structure

- (*dict*) –

Represents the output of a *Query* operation.

- **Items** (*list*) –

An array of item attributes that match the query criteria. Each element in this array consists of an attribute name and the value for that attribute.

- * (*dict*) –

- (*string*) –

- (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).

- **Count** (*integer*) –

The number of items in the response.

If you used a *QueryFilter* in the request, then *Count* is the number of items returned after the filter was applied, and *ScannedCount* is the number of matching items before the filter was applied.

If you did not use a filter in the request, then *Count* and *ScannedCount* are the same.

- **ScannedCount** (*integer*) –

The number of items evaluated, before any *QueryFilter* is applied. A high *ScannedCount* value with few, or no, *Count* results indicates an inefficient *Query* operation. For more information, see [Count and ScannedCount](#) in the *Amazon DynamoDB Developer Guide*.

If you did not use a filter in the request, then *ScannedCount* is the same as *Count*.

- **LastEvaluatedKey** (*dict*) –

The primary key of the item where the operation stopped, inclusive of the previous result set. Use this value to start a new operation, excluding this value in the new request.

If *LastEvaluatedKey* is empty, then the “last page” of results has been processed and there is no more data to be retrieved.

If *LastEvaluatedKey* is not empty, it does not necessarily mean that there is more data in the result set. The only way to know when you have reached the end of the result set is when *LastEvaluatedKey* is empty.

- * (*string*) –

- (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).

- **ConsumedCapacity** (*dict*) –

The capacity units consumed by an operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation.

ConsumedCapacity is only returned if the request asked for it. For more information, see [Provisioned Throughput](#) in the *Amazon DynamoDB Developer Guide*.

* **TableName** (*string*) –

The name of the table that was affected by the operation.

* **CapacityUnits** (*float*) –

The total number of capacity units consumed by the operation.

* **Table** (*dict*) –

The amount of throughput consumed on the table affected by the operation.

· **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

* **LocalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each local index affected by the operation.

· (*string*) –

· (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

· **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

* **GlobalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each global index affected by the operation.

· (*string*) –

· (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

· **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

reload()

Calls `dynamodb.Client.describe_table()` to update the attributes of the Table resource

Request Syntax

```
table.reload()
```

Returns None

scan (***kwargs*)

The *Scan* operation returns one or more items and item attributes by accessing every item in a table or a secondary index. To have DynamoDB return fewer items, you can provide a *ScanFilter* operation.

If the total number of scanned items exceeds the maximum data set size limit of 1 MB, the scan stops and results are returned to the user as a *LastEvaluatedKey* value to continue the scan in a

subsequent operation. The results also include the number of items exceeding the limit. A scan can result in no table data meeting the filter criteria.

By default, *Scan* operations proceed sequentially; however, for faster performance on a large table or secondary index, applications can request a parallel *Scan* operation by providing the *Segment* and *TotalSegments* parameters. For more information, see [Parallel Scan](#) in the *Amazon DynamoDB Developer Guide* .

By default, *Scan* uses eventually consistent reads when accessing the data in the table or local secondary index. However, you can use strongly consistent reads instead by setting the *ConsistentRead* parameter to *true* .

Request Syntax

```
response = table.scan(
    IndexName='string',
    AttributesToGet=[
        'string',
    ],
    Limit=123,
    Select='ALL_ATTRIBUTES' | 'ALL_PROJECTED_ATTRIBUTES' | 'SPECIFIC_ATTRIBUTES' | 'COUNT',
    ScanFilter={
        'string': {
            'AttributeValueList': [
                'string' | 123 | Binary(b'bytes') | True | None | set(['string']) | set([123]) | set
            ],
            'ComparisonOperator': 'EQ' | 'NE' | 'IN' | 'LE' | 'LT' | 'GE' | 'GT' | 'BETWEEN' | 'NOT_NULL'
        }
    },
    ConditionalOperator='AND' | 'OR',
    ExclusiveStartKey={
        'string': 'string' | 123 | Binary(b'bytes') | True | None | set(['string']) | set([123]) | set
    },
    ReturnConsumedCapacity='INDEXES' | 'TOTAL' | 'NONE',
    TotalSegments=123,
    Segment=123,
    ProjectionExpression='string',
    FilterExpression=Attr('myattribute').eq('myvalue'),
    ExpressionAttributeNames={
        'string': 'string'
    },
    ExpressionAttributeValues={
        'string': 'string' | 123 | Binary(b'bytes') | True | None | set(['string']) | set([123]) | set
    },
    ConsistentRead=True | False
)
```

Parameters

- **IndexName** (*string*) – The name of a secondary index to scan. This index can be any local secondary index or global secondary index. Note that if you use the *IndexName* parameter, you must also provide *TableName* .

- **AttributesToGet** (*list*) –

Warning: This is a legacy parameter, for backward compatibility. New applications should use *ProjectionExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception.

This parameter allows you to retrieve attributes of type List or Map; however, it cannot retrieve individual elements within a List or a Map.

The names of one or more attributes to retrieve. If no attribute names are provided, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

Note that *AttributesToGet* has no effect on provisioned throughput consumption. DynamoDB determines capacity units consumed based on item size, not on the amount of data that is returned to an application.

- (*string*) –
- **Limit** (*integer*) – The maximum number of items to evaluate (not necessarily the number of matching items). If DynamoDB processes the number of items up to the limit while processing the results, it stops the operation and returns the matching values up to that point, and a key in *LastEvaluatedKey* to apply in a subsequent operation, so that you can pick up where you left off. Also, if the processed data set size exceeds 1 MB before DynamoDB reaches this limit, it stops the operation and returns the matching values up to the limit, and a key in *LastEvaluatedKey* to apply in a subsequent operation to continue the operation. For more information, see [Query and Scan](#) in the *Amazon DynamoDB Developer Guide*.
- **Select** (*string*) – The attributes to be returned in the result. You can retrieve all item attributes, specific item attributes, or the count of matching items.
 - `ALL_ATTRIBUTES` - Returns all of the item attributes.
 - `COUNT` - Returns the number of matching items, rather than the matching items themselves.
 - `SPECIFIC_ATTRIBUTES` - Returns only the attributes listed in *AttributesToGet*. This return value is equivalent to specifying *AttributesToGet* without specifying any value for *Select*.

If neither *Select* nor *AttributesToGet* are specified, DynamoDB defaults to `ALL_ATTRIBUTES`. You cannot use both *AttributesToGet* and *Select* together in a single request, unless the value for *Select* is `SPECIFIC_ATTRIBUTES`. (This usage is equivalent to specifying *AttributesToGet* without any value for *Select*.)

- **ScanFilter** (*dict*) –

Warning: This is a legacy parameter, for backward compatibility. New applications should use *FilterExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception.

A condition that evaluates the scan results and returns only the desired values.

Note: This parameter does not support attributes of type List or Map.

If you specify more than one condition in the *ScanFilter* map, then by default all of the conditions must evaluate to true. In other words, the conditions are ANDed together. (You can use the *ConditionalOperator* parameter to OR the conditions instead. If you do this, then at least one of

the conditions must evaluate to true, rather than all of them.)

Each *ScanFilter* element consists of an attribute name to compare, along with the following:

- *AttributeValueList* - One or more values to evaluate against the supplied attribute. The number of values in the list depends on the operator specified in *ComparisonOperator* . For type Number, value comparisons are numeric. String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A , and a is greater than B . For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters . For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values. For information on specifying data types in JSON, see [JSON Data Format](#) in the *Amazon DynamoDB Developer Guide* .
- *ComparisonOperator* - A comparator for evaluating attributes. For example, equals, greater than, less than, etc. The following comparison operators are available: EQ | NE | LE | LT | GE | GT | NOT_NULL | NULL | CONTAINS | NOT_CONTAINS | BEGINS_WITH | IN | BETWEEN For complete descriptions of all comparison operators, see [Condition](#) .
- (string) -
 - * (dict) -

Represents the selection criteria for a *Query* or *Scan* operation:

- For a *Query* operation, *Condition* is used for specifying the *KeyConditions* to use when querying a table or an index. For *KeyConditions* , only the following comparison operators are supported: EQ | LE | LT | GE | GT | BEGINS_WITH | BETWEEN *Condition* is also used in a *QueryFilter* , which evaluates the query results and returns only the desired values.
- For a *Scan* operation, *Condition* is used in a *ScanFilter* , which evaluates the scan results and returns only the desired values.
- **AttributeValueList** (*list*) -

One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *ComparisonOperator* being used.

For type Number, value comparisons are numeric.

String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A , and a is greater than B . For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters .

For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.

- (*valid DynamoDB type*) - - The value of the attribute. The valid value types are listed in the [DynamoDB Ref](#)

erence Guide.

- **ComparisonOperator** (*string*) – [REQUIRED]

A comparator for evaluating attributes. For example, equals, greater than, less than, etc.

The following comparison operators are available:

```
EQ | NE | LE | LT | GE | GT |
NOT_NULL | NULL | CONTAINS |
NOT_CONTAINS | BEGINS_WITH | IN |
BETWEEN
```

The following are descriptions of each comparison operator.

- **EQ** : Equal. EQ is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] } .
- **NE** : Not equal. NE is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] } .
- **LE** : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- **LT** : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- **GE** : Greater than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .

- **GT** : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" } . Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] } .
- **NOT_NULL** : The attribute exists. **NOT_NULL** is supported for all datatypes, including lists and maps. .. note::This operator tests for the existence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using **NOT_NULL** , the result is a Boolean *true* . This result is because the attribute “a ” exists; its data type is not relevant to the **NOT_NULL** comparison operator.
- **NULL** : The attribute does not exist. **NULL** is supported for all datatypes, including lists and maps. .. note::This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using **NULL** , the result is a Boolean *false* . This is because the attribute “a ” exists; its data type is not relevant to the **NULL** comparison operator.
- **CONTAINS** : Checks for a subsequence, or value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS ”, “NS ”, or “BS ”), then the operator evaluates to true if it finds an exact match with any member of the set. **CONTAINS** is supported for lists: When evaluating “a **CONTAINS** b ”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
- **NOT_CONTAINS** : Checks for absence of a subsequence, or absence of a value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS ”, “NS ”, or “BS ”), then the operator evaluates to true if it *does not* find an exact match with any member of the set. **NOT_CONTAINS** is supported for lists: When evaluating “a **NOT CONTAINS** b ”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
- **BEGINS_WITH** : Checks for a prefix. *AttributeValueList* can contain only one *AttributeValue* of type

String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).

- **IN** : Checks for matching elements within two sets. *AttributeValueList* can contain one or more *AttributeValue* elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
- **BETWEEN** : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not compare to { "N" : "6" } . Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }

For usage examples of *AttributeValueList* and *ComparisonOperator* , see [Legacy Conditional Parameters](#) in the *Amazon DynamoDB Developer Guide* .

- **ConditionalOperator** (*string*) –

Warning: This is a legacy parameter, for backward compatibility. New applications should use *FilterExpression* instead. Do not combine legacy parameters and expression parameters in a single request; otherwise, DynamoDB will return a *ValidationException*.

A logical operator to apply to the conditions in a *ScanFilter* map:

- **AND** - If all of the conditions evaluate to true, then the entire map evaluates to true.
- **OR** - If at least one of the conditions evaluate to true, then the entire map evaluates to true.

If you omit *ConditionalOperator* , then AND is the default.

The operation will succeed only if the entire map evaluates to true.

Note: This parameter does not support attributes of type List or Map.

- **ExclusiveStartKey** (*dict*) – The primary key of the first item that this operation will evaluate. Use the value that was returned for *LastEvaluatedKey* in the previous operation.

The data type for *ExclusiveStartKey* must be String, Number or Binary. No set data types are allowed.

In a parallel scan, a *Scan* request that includes *ExclusiveStartKey* must specify the same segment whose previous *Scan* returned the corresponding value of *LastEvaluatedKey* .

- (*string*) –
 - * (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference](#)

Guide.

- **ReturnConsumedCapacity** (*string*) – Determines the level of detail about provisioned throughput consumption that is returned in the response:
 - *INDEXES* - The response includes the aggregate *ConsumedCapacity* for the operation, together with *ConsumedCapacity* for each table and secondary index that was accessed. Note that some operations, such as *GetItem* and *BatchGetItem*, do not access any indexes at all. In these cases, specifying *INDEXES* will only return *ConsumedCapacity* information for table(s).
 - *TOTAL* - The response includes only the aggregate *ConsumedCapacity* for the operation.
 - *NONE* - No *ConsumedCapacity* details are included in the response.
- **TotalSegments** (*integer*) – For a parallel *Scan* request, *TotalSegments* represents the total number of segments into which the *Scan* operation will be divided. The value of *TotalSegments* corresponds to the number of application workers that will perform the parallel scan. For example, if you want to use four application threads to scan a table or an index, specify a *TotalSegments* value of 4.

The value for *TotalSegments* must be greater than or equal to 1, and less than or equal to 1000000. If you specify a *TotalSegments* value of 1, the *Scan* operation will be sequential rather than parallel.

If you specify *TotalSegments*, you must also specify *Segment*.

- **Segment** (*integer*) – For a parallel *Scan* request, *Segment* identifies an individual segment to be scanned by an application worker.

Segment IDs are zero-based, so the first segment is always 0. For example, if you want to use four application threads to scan a table or an index, then the first thread specifies a *Segment* value of 0, the second thread specifies 1, and so on.

The value of *LastEvaluatedKey* returned from a parallel *Scan* request must be used as *ExclusiveStartKey* with the same segment ID in a subsequent *Scan* operation.

The value for *Segment* must be greater than or equal to 0, and less than the value provided for *TotalSegments*.

If you provide *Segment*, you must also provide *TotalSegments*.

- **ProjectionExpression** (*string*) – A string that identifies one or more attributes to retrieve from the specified table or index. These attributes can include scalars, sets, or elements of a JSON document. The attributes in the expression must be separated by commas.

If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

For more information, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide*.

Note: *ProjectionExpression* replaces the legacy *AttributesToGet* parameter.

- **FilterExpression** (condition from `boto3.dynamodb.conditions.Attr` method) – The condition(s) an attribute(s) must meet. Valid conditions are listed in the [DynamoDB Reference Guide](#).

- **ExpressionAttributeNames** (*dict*) – One or more substitution tokens for attribute names in an expression. The following are some use cases for using *ExpressionAttributeNames* :

- To access an attribute whose name conflicts with a DynamoDB reserved word.
- To create a placeholder for repeating occurrences of an attribute name in an expression.
- To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the `#` character in an expression to dereference an attribute name. For example, consider the following attribute name:

- Percentile

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see [Reserved Words](#) in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for *ExpressionAttributeNames* :

- { "#P": "Percentile" }

You could then use this substitution in an expression, as in this example:

- #P = :val

Note: Tokens that begin with the `:` character are *expression attribute values* , which are placeholders for the actual value at runtime.

For more information on expression attribute names, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide* .

- (*string*) –
- * (*string*) –

- **ExpressionAttributeValues** (*dict*) – One or more values that can be substituted in an expression.

Use the `:` (colon) character in an expression to dereference an attribute value. For example, suppose that you wanted to check whether the value of the *ProductStatus* attribute was one of the following:

```
Available | Backordered | Discontinued
```

You would first need to specify *ExpressionAttributeValues* as follows:

```
{ ":avail": {"S": "Available"},
  ":back": {"S": "Backordered"},
  ":disc": {"S": "Discontinued"} }
```

You could then use these values in an expression, such as this:

```
ProductStatus IN (:avail, :back, :disc)
```

For more information on expression attribute values, see [Specifying Conditions](#) in the *Amazon DynamoDB Developer Guide* .

- (*string*) –
- * (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).

- **ConsistentRead** (*boolean*) – A Boolean value that determines the read consistency model during the scan:

- If *ConsistentRead* is `false` , then *Scan* will use eventually consistent reads. The data returned from *Scan* might not contain the results

of other recently completed write operations (PutItem, UpdateItem or DeleteItem). The *Scan* response might include some stale data.

- If *ConsistentRead* is `true`, then *Scan* will use strongly consistent reads. All of the write operations that completed before the *Scan* began are guaranteed to be contained in the *Scan* response.

The default setting for *ConsistentRead* is `false`, meaning that eventually consistent reads will be used.

Strongly consistent reads are not supported on global secondary indexes. If you scan a global secondary index with *ConsistentRead* set to `true`, you will receive a *ValidationException*.

Return type dict

Returns

Response Syntax

```
{
  'Items': [
    {
      'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])
    },
  ],
  'Count': 123,
  'ScannedCount': 123,
  'LastEvaluatedKey': {
    'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])|set
  },
  'ConsumedCapacity': {
    'TableName': 'string',
    'CapacityUnits': 123.0,
    'Table': {
      'CapacityUnits': 123.0
    },
  },
  'LocalSecondaryIndexes': {
    'string': {
      'CapacityUnits': 123.0
    }
  },
  'GlobalSecondaryIndexes': {
    'string': {
      'CapacityUnits': 123.0
    }
  }
}
```

Response Structure

- (*dict*) –

Represents the output of a *Scan* operation.

- **Items** (*list*) –

An array of item attributes that match the scan criteria. Each element in this array consists of an attribute name and the value for that attribute.

- * (*dict*) –

- (*string*) –

- (valid *DynamoDB* type) – - The value of the attribute. The valid value types are listed in the *DynamoDB Reference Guide*.

– **Count** (*integer*) –

The number of items in the response.

If you set *ScanFilter* in the request, then *Count* is the number of items returned after the filter was applied, and *ScannedCount* is the number of matching items before the filter was applied.

If you did not use a filter in the request, then *Count* is the same as *ScannedCount*.

– **ScannedCount** (*integer*) –

The number of items evaluated, before any *ScanFilter* is applied. A high *ScannedCount* value with few, or no, *Count* results indicates an inefficient *Scan* operation. For more information, see *Count and ScannedCount* in the *Amazon DynamoDB Developer Guide*.

If you did not use a filter in the request, then *ScannedCount* is the same as *Count*.

– **LastEvaluatedKey** (*dict*) –

The primary key of the item where the operation stopped, inclusive of the previous result set. Use this value to start a new operation, excluding this value in the new request.

If *LastEvaluatedKey* is empty, then the “last page” of results has been processed and there is no more data to be retrieved.

If *LastEvaluatedKey* is not empty, it does not necessarily mean that there is more data in the result set. The only way to know when you have reached the end of the result set is when *LastEvaluatedKey* is empty.

* (*string*) –

- (valid *DynamoDB* type) – - The value of the attribute. The valid value types are listed in the *DynamoDB Reference Guide*.

– **ConsumedCapacity** (*dict*) –

The capacity units consumed by an operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. *ConsumedCapacity* is only returned if the request asked for it. For more information, see *Provisioned Throughput* in the *Amazon DynamoDB Developer Guide*.

* **TableName** (*string*) –

The name of the table that was affected by the operation.

* **CapacityUnits** (*float*) –

The total number of capacity units consumed by the operation.

* **Table** (*dict*) –

The amount of throughput consumed on the table affected by the operation.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

* **LocalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each local index affected by the operation.

- (*string*) –
- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

· **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

* **GlobalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each global index affected by the operation.

- (*string*) –
- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

· **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

update (***kwargs*)

Modifies the provisioned throughput settings, global secondary indexes, or DynamoDB Streams settings for a given table.

You can only perform one of the following operations at once:

- Modify the provisioned throughput settings of the table.
- Enable or disable Streams on the table.
- Remove a global secondary index from the table.
- Create a new global secondary index on the table. Once the index begins backfilling, you can use *UpdateTable* to perform other operations.

UpdateTable is an asynchronous operation; while it is executing, the table status changes from ACTIVE to UPDATING . While it is UPDATING , you cannot issue another *UpdateTable* request. When the table returns to the ACTIVE state, the *UpdateTable* operation is complete.

Request Syntax

```
table = table.update(  
    AttributeDefinitions=[  
        {  
            'AttributeName': 'string',  
            'AttributeType': 'S' | 'N' | 'B'  
        },  
    ],  
    ProvisionedThroughput={  
        'ReadCapacityUnits': 123,  
        'WriteCapacityUnits': 123  
    },  
    GlobalSecondaryIndexUpdates=[  
        {  
            'Update': {
```

```

        'IndexName': 'string',
        'ProvisionedThroughput': {
            'ReadCapacityUnits': 123,
            'WriteCapacityUnits': 123
        }
    },
    'Create': {
        'IndexName': 'string',
        'KeySchema': [
            {
                'AttributeName': 'string',
                'KeyType': 'HASH' | 'RANGE'
            },
        ],
        'Projection': {
            'ProjectionType': 'ALL' | 'KEYS_ONLY' | 'INCLUDE',
            'NonKeyAttributes': [
                'string',
            ]
        },
        'ProvisionedThroughput': {
            'ReadCapacityUnits': 123,
            'WriteCapacityUnits': 123
        }
    },
    'Delete': {
        'IndexName': 'string'
    }
},
],
StreamSpecification={
    'StreamEnabled': True|False,
    'StreamViewType': 'NEW_IMAGE' | 'OLD_IMAGE' | 'NEW_AND_OLD_IMAGES' | 'KEYS_ONLY'
}
)

```

Parameters

- **AttributeDefinitions** (*list*) – An array of attributes that describe the key schema for the table and indexes. If you are adding a new global secondary index to the table, *AttributeDefinitions* must include the key element(s) of the new index.

– (*dict*) –

Represents an attribute for describing the key schema for the table and indexes.

* **AttributeName** (*string*) – [REQUIRED]

A name for the attribute.

* **AttributeType** (*string*) – [REQUIRED]

The data type for the attribute.

- **ProvisionedThroughput** (*dict*) – Represents the provisioned throughput settings for a specified table or index. The settings can be modified using the *UpdateTable* operation.

For current minimum and maximum provisioned throughput values, see [Limits](#) in the *Amazon DynamoDB Developer Guide*.

– **ReadCapacityUnits** (*integer*) – [REQUIRED]

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a *ThrottlingException* . For more information, see [Specifying Read and Write Requirements](#) in the *Amazon DynamoDB Developer Guide* .

– **WriteCapacityUnits** (*integer*) – [REQUIRED]

The maximum number of writes consumed per second before DynamoDB returns a *ThrottlingException* . For more information, see [Specifying Read and Write Requirements](#) in the *Amazon DynamoDB Developer Guide* .

• **GlobalSecondaryIndexUpdates** (*list*) – An array of one or more global secondary indexes for the table. For each index in the array, you can request one action:

- *Create* - add a new global secondary index to the table.
- *Update* - modify the provisioned throughput settings of an existing global secondary index.
- *Delete* - remove a global secondary index from the table.

For more information, see [Managing Global Secondary Indexes](#) in the *Amazon DynamoDB Developer Guide* .

– (*dict*) –

Represents one of the following:

- * A new global secondary index to be added to an existing table.
- * New provisioned throughput parameters for an existing global secondary index.
- * An existing global secondary index to be removed from an existing table.
- * **Update** (*dict*) –

The name of an existing global secondary index, along with new provisioned throughput settings to be applied to that index.

· **IndexName** (*string*) – [REQUIRED]

The name of the global secondary index to be updated.

· **ProvisionedThroughput** (*dict*) – [REQUIRED]

Represents the provisioned throughput settings for a specified table or index. The settings can be modified using the *UpdateTable* operation.

For current minimum and maximum provisioned throughput values, see [Limits](#) in the *Amazon DynamoDB Developer Guide* .

· **ReadCapacityUnits** (*integer*) – [REQUIRED]

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a *ThrottlingException* . For more information, see [Specifying Read and Write Requirements](#) in the *Amazon DynamoDB Developer Guide* .

· **WriteCapacityUnits** (*integer*) – [REQUIRED]

The maximum number of writes consumed per second before DynamoDB returns a *ThrottlingException* . For more information, see [Specifying Read and Write Requirements](#) in the *Amazon DynamoDB Developer*

Guide .

* **Create** (*dict*) –

The parameters required for creating a global secondary index on an existing table:

- `IndexName`
- `KeySchema`
- `AttributeDefinitions`
- `Projection`
- `ProvisionedThroughput`
- **IndexName** (*string*) – [REQUIRED]

The name of the global secondary index to be created.

- **KeySchema** (*list*) – [REQUIRED]

The key schema for the global secondary index.

- (*dict*) –

Represents a *single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement* . A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.

- **AttributeName** (*string*) – [REQUIRED]

The name of a key attribute.

- **KeyType** (*string*) – [REQUIRED]

The attribute data, consisting of the data type and the attribute value itself.

- **Projection** (*dict*) – [REQUIRED]

Represents attributes that are copied (projected) from the table into an index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

- **ProjectionType** (*string*) –

The set of attributes that are projected into the index:

- `KEYS_ONLY` - Only the index and primary keys are projected into the index.
- `INCLUDE` - Only the specified table attributes are projected into the index. The list of projected attributes are in *NonKeyAttributes* .
- `ALL` - All of the table attributes are projected into the index.

- **NonKeyAttributes** (*list*) –

Represents the non-key attribute names which will be projected into the index.

For local secondary indexes, the total count of *Non-KeyAttributes* summed across all of the local secondary indexes, must not exceed 20. If you project the same

attribute into two different indexes, this counts as two distinct attributes when determining the total.

- *(string)* –
- **ProvisionedThroughput** (*dict*) – [REQUIRED]

Represents the provisioned throughput settings for a specified table or index. The settings can be modified using the *UpdateTable* operation.

For current minimum and maximum provisioned throughput values, see [Limits](#) in the *Amazon DynamoDB Developer Guide*.

- **ReadCapacityUnits** (*integer*) – [REQUIRED]

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a *ThrottlingException*. For more information, see [Specifying Read and Write Requirements](#) in the *Amazon DynamoDB Developer Guide*.

- **WriteCapacityUnits** (*integer*) – [REQUIRED]

The maximum number of writes consumed per second before DynamoDB returns a *ThrottlingException*. For more information, see [Specifying Read and Write Requirements](#) in the *Amazon DynamoDB Developer Guide*.

- * **Delete** (*dict*) –

The name of an existing global secondary index to be removed.

- **IndexName** (*string*) – [REQUIRED]

The name of the global secondary index to be deleted.

- **StreamSpecification** (*dict*) – Represents the DynamoDB Streams configuration for the table.

Note: You will receive a *ResourceInUseException* if you attempt to enable a stream on a table that already has a stream, or if you attempt to disable a stream on a table which does not have a stream.

- **StreamEnabled** (*boolean*) –

Indicates whether DynamoDB Streams is enabled (true) or disabled (false) on the table.

- **StreamViewType** (*string*) –

The DynamoDB Streams settings for the table. These settings consist of:

- * *StreamEnabled* - Indicates whether DynamoDB Streams is enabled (true) or disabled (false) on the table.
- * *StreamViewType* - When an item in the table is modified, *StreamViewType* determines what information is written to the stream for this table. Valid values for *StreamViewType* are: * *KEYS_ONLY* - Only the key attributes of the modified item are written to the stream.
- * *NEW_IMAGE* - The entire item, as it appears after it was modified, is written to the stream.

- * *OLD_IMAGE* - The entire item, as it appeared before it was modified, is written to the stream.
- * *NEW_AND_OLD_IMAGES* - Both the new and the old item images of the item are written to the stream.

Return type dynamodb.Table

Returns Table resource

```
update_item(**kwargs)
```

Edits an existing item's attributes, or adds a new item to the table if it does not already exist. You can put, delete, or add attribute values. You can also perform a conditional update on an existing item (insert a new attribute name-value pair if it doesn't exist, or replace an existing name-value pair if it has certain expected attribute values). If conditions are specified and the item does not exist, then the operation fails and a new item is not created.

You can also return the item's attribute values in the same *UpdateItem* operation using the *ReturnValues* parameter.

Request Syntax

```
response = table.update_item(
    Key={
        'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])|set([123])|se
    },
    AttributeUpdates={
        'string': {
            'Value': 'string'|123|Binary(b'bytes')|True|None|set(['string'])|set([123])
            'Action': 'ADD'|'PUT'|'DELETE'
        }
    },
    Expected={
        'string': {
            'Value': 'string'|123|Binary(b'bytes')|True|None|set(['string'])|set([123])
            'Exists': True|False,
            'ComparisonOperator': 'EQ'|'NE'|'IN'|'LE'|'LT'|'GE'|'GT'|'BETWEEN'|'NOT_NULL'
            'AttributeValueList': [
                'string'|123|Binary(b'bytes')|True|None|set(['string'])|set([123])|set
            ]
        }
    },
    ConditionalOperator='AND'|'OR',
    ReturnValues='NONE'|'ALL_OLD'|'UPDATED_OLD'|'ALL_NEW'|'UPDATED_NEW',
    ReturnConsumedCapacity='INDEXES'|'TOTAL'|'NONE',
    ReturnItemCollectionMetrics='SIZE'|'NONE',
    UpdateExpression='string',
    ConditionExpression=Attr('myattribute').eq('myvalue'),
    ExpressionAttributeNames={
        'string': 'string'
    },
    ExpressionAttributeValues={
        'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])|set([123])|se
    }
)
```

Parameters

- **Key** (*dict*) – **[REQUIRED]**

The primary key of the item to be updated. Each element consists of an attribute name and a value for that attribute.

For the primary key, you must provide all of the attributes. For example, with a hash type primary key, you only need to provide the hash attribute. For a hash-and-range type primary key, you must provide both the hash attribute and the range attribute.

- (*string*) –

- * (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).

- **AttributeUpdates** (*dict*) –

Warning: This is a legacy parameter, for backward compatibility. New applications should use *UpdateExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception. This parameter can be used for modifying top-level attributes; however, it does not support individual list or map elements.

The names of attributes to be modified, the action to perform on each, and the new value for each. If you are updating an attribute that is an index key attribute for any indexes on that table, the attribute type must match the index key type defined in the *AttributesDefinition* of the table description. You can use *UpdateItem* to update any nonkey attributes.

Attribute values cannot be null. String and Binary type attributes must have lengths greater than zero. Set type attributes must not be empty. Requests with empty values will be rejected with a *ValidationException* exception.

Each *AttributeUpdates* element consists of an attribute name to modify, along with the following:

- *Value* - The new value, if applicable, for this attribute.
- *Action* - A value that specifies how to perform the update. This action is only valid for an existing attribute whose data type is Number or is a set; do not use ADD for other data types. If an item with the specified primary key is found in the table, the following values perform the following actions: * PUT - Adds the specified attribute to the item. If the attribute already exists, it is replaced by the new value.
- DELETE - Removes the attribute and its value, if no value is specified for DELETE. The data type of the specified value must match the existing value's data type. If a set of values is specified, then those values are subtracted from the old set. For example, if the attribute value was the set [a, b, c] and the DELETE action specifies [a, c], then the final attribute value is [b]. Specifying an empty set is an error.
- ADD - Adds the specified value to the item, if the attribute does not already exist. If the attribute does exist, then the behavior of ADD depends on the data type of the attribute: * If the existing attribute is a number, and if *Value* is also a number, then *Value* is mathematically added to the existing attribute. If *Value* is a negative number, then it is subtracted from the existing attribute. .. note:: If you use ADD to increment or decrement a number value for an item that doesn't exist before the update, DynamoDB uses 0 as the initial value. Similarly, if you use ADD for an existing item to increment or decrement an attribute value that doesn't exist before the update, DynamoDB uses 0 as the initial value. For example, suppose that the item you

want to update doesn't have an attribute named *itemcount* , but you decide to ADD the number 3 to this attribute anyway. DynamoDB will create the *itemcount* attribute, set its initial value to 0 , and finally add 3 to it. The result will be a new *itemcount* attribute, with a value of 3 .

- If the existing data type is a set, and if *Value* is also a set, then *Value* is appended to the existing set. For example, if the attribute value is the set [1 , 2] , and the ADD action specified [3] , then the final attribute value is [1 , 2 , 3] . An error occurs if an ADD action is specified for a set attribute and the attribute type specified does not match the existing set type. Both sets must have the same primitive data type. For example, if the existing data type is a set of strings, *Value* must also be a set of strings.

If no item with the specified key is found in the table, the following values perform the following actions:

- PUT - Causes DynamoDB to create a new item with the specified primary key, and then adds the attribute.
- DELETE - Nothing happens, because attributes cannot be deleted from a nonexistent item. The operation succeeds, but DynamoDB does not create a new item.
- ADD - Causes DynamoDB to create an item with the supplied primary key and number (or set of numbers) for the attribute value. The only data types allowed are Number and Number Set.

If you provide any attributes that are part of an index key, then the data types for those attributes must match those of the schema in the table's attribute definition.

- (*string*) -
- * (*dict*) -

For the *UpdateItem* operation, represents the attributes to be modified, the action to perform on each, and the new value for each.

Note: You cannot use *UpdateItem* to update any primary key attributes. Instead, you will need to delete the item, and then use *PutItem* to create a new item with new attributes.

Attribute values cannot be null; string and binary type attributes must have lengths greater than zero; and set type attributes must not be empty. Requests with empty values will be rejected with a *ValidationException* exception.

- **Value** (*valid DynamoDB type*) - - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).
- **Action** (*string*) -

Specifies how to perform the update. Valid values are PUT (default), DELETE , and ADD . The behavior depends on whether the specified primary key already exists in the table.

If an item with the specified *Key* is found in the table:

- PUT - Adds the specified attribute to the item. If the attribute already exists, it is replaced by the new value.
- DELETE - If no value is specified, the attribute and

its value are removed from the item. The data type of the specified value must match the existing value's data type. If a *set* of values is specified, then those values are subtracted from the old set. For example, if the attribute value was the set `[a, b, c]` and the *DELETE* action specified `[a, c]`, then the final attribute value would be `[b]`. Specifying an empty set is an error.

- **ADD** - If the attribute does not already exist, then the attribute and its values are added to the item. If the attribute does exist, then the behavior of **ADD** depends on the data type of the attribute: * If the existing attribute is a number, and if *Value* is also a number, then the *Value* is mathematically added to the existing attribute. If *Value* is a negative number, then it is subtracted from the existing attribute. .. note:: If you use **ADD** to increment or decrement a number value for an item that doesn't exist before the update, DynamoDB uses 0 as the initial value. In addition, if you use **ADD** to update an existing item, and intend to increment or decrement an attribute value which does not yet exist, DynamoDB uses 0 as the initial value. For example, suppose that the item you want to update does not yet have an attribute named *itemcount*, but you decide to **ADD** the number 3 to this attribute anyway, even though it currently does not exist. DynamoDB will create the *itemcount* attribute, set its initial value to 0, and finally add 3 to it. The result will be a new *itemcount* attribute in the item, with a value of 3.
- If the existing data type is a set, and if the *Value* is also a set, then the *Value* is added to the existing set. (This is a *set* operation, not mathematical addition.) For example, if the attribute value was the set `[1, 2]`, and the **ADD** action specified `[3]`, then the final attribute value would be `[1, 2, 3]`. An error occurs if an **Add** action is specified for a set attribute and the attribute type specified does not match the existing set type. Both sets must have the same primitive data type. For example, if the existing data type is a set of strings, the *Value* must also be a set of strings. The same holds true for number sets and binary sets.

This action is only valid for an existing attribute whose data type is number or is a set. Do not use **ADD** for any other data types.

If no item with the specified *Key* is found:

- **PUT** - DynamoDB creates a new item with the specified primary key, and then adds the attribute.
- **DELETE** - Nothing happens; there is no attribute to delete.
- **ADD** - DynamoDB creates an item with the supplied primary key and number (or set of numbers) for the attribute value. The only data types allowed are number and number set; no other data types can be specified.

- **Expected** (*dict*) –

Warning: This is a legacy parameter, for backward compatibility. New applications should use *ConditionExpression* instead. Do not combine legacy parameters and expression parameters in a single API call; otherwise, DynamoDB will return a *ValidationException* exception.

A map of attribute/condition pairs. *Expected* provides a conditional block for the *UpdateItem* operation.

Each element of *Expected* consists of an attribute name, a comparison operator, and one or more values. DynamoDB compares the attribute with the value(s) you supplied, using the comparison operator. For each *Expected* element, the result of the evaluation is either true or false.

If you specify more than one element in the *Expected* map, then by default all of the conditions must evaluate to true. In other words, the conditions are ANDed together. (You can use the *ConditionalOperator* parameter to OR the conditions instead. If you do this, then at least one of the conditions must evaluate to true, rather than all of them.)

If the *Expected* map evaluates to true, then the conditional operation succeeds; otherwise, it fails.

Expected contains the following:

- *AttributeValueList* - One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *ComparisonOperator* being used. For type Number, value comparisons are numeric. String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A, and a is greater than B. For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters. For type Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.
- *ComparisonOperator* - A comparator for evaluating attributes in the *AttributeValueList*. When performing the comparison, DynamoDB uses strongly consistent reads. The following comparison operators are available: EQ | NE | LE | LT | GE | GT | NOT_NULL | NULL | CONTAINS | NOT_CONTAINS | BEGINS_WITH | IN | BETWEEN. The following are descriptions of each comparison operator. * EQ : Equal. EQ is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" }. Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] }.
- NE : Not equal. NE is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" }. Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] }.

- LE : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- LT : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- GE : Greater than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- GT : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" } . Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] } .
- NOT_NULL : The attribute exists. NOT_NULL is supported for all datatypes, including lists and maps. .. note::This operator tests for the existence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using NOT_NULL , the result is a Boolean *true* . This result is because the attribute “a ” exists; its data type is not relevant to the NOT_NULL comparison operator.
- NULL : The attribute does not exist. NULL is supported for all datatypes, including lists and maps. .. note::This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using NULL , the result is a Boolean *false* . This is because the attribute “a ” exists; its data type is not relevant to the NULL comparison operator.
- CONTAINS : Checks for a subsequence, or value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS ”, “NS ”, or “BS ”), then the operator evaluates to true if it finds an exact match with any member of the set. CONTAINS is supported for lists: When evaluating “a CONTAINS b”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
- NOT_CONTAINS : Checks for absence of a subsequence, or absence of a value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set

type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set ("SS ", "NS ", or "BS "), then the operator evaluates to true if it *does not* find an exact match with any member of the set. NOT_CONTAINS is supported for lists: When evaluating "a NOT CONTAINS b", "a " can be a list; however, "b " cannot be a set, a map, or a list.

- **BEGINS_WITH** : Checks for a prefix. *AttributeValueList* can contain only one *AttributeValue* of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).
- **IN** : Checks for matching elements within two sets. *AttributeValueList* can contain one or more *AttributeValue* elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
- **BETWEEN** : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, {"S": "6"} does not compare to {"N": "6"} . Also, {"N": "6"} does not compare to {"NS": ["6", "2", "1"]}

For usage examples of *AttributeValueList* and *ComparisonOperator* , see [Legacy Conditional Parameters](#) in the *Amazon DynamoDB Developer Guide* .

For backward compatibility with previous DynamoDB releases, the following parameters can be used instead of *AttributeValueList* and *ComparisonOperator* :

- *Value* - A value for DynamoDB to compare with an attribute.
- *Exists* - A Boolean value that causes DynamoDB to evaluate the value before attempting the conditional operation: * If *Exists* is `true` , DynamoDB will check to see if that attribute value already exists in the table. If it is found, then the condition evaluates to true; otherwise the condition evaluate to false.
- If *Exists* is `false` , DynamoDB assumes that the attribute value does *not* exist in the table. If in fact the value does not exist, then the assumption is valid and the condition evaluates to true. If the value is found, despite the assumption that it does not exist, the condition evaluates to false.

Note that the default value for *Exists* is `true` .

The *Value* and *Exists* parameters are incompatible with *AttributeValueList* and *ComparisonOperator* . Note that if you use both sets of parameters at once, DynamoDB will return a *ValidationException* exception.

Note: This parameter does not support attributes of type List or Map.

- (string) –
- * (dict) –

Represents a condition to be compared with an attribute value. This condition can be used with *DeleteItem* , *PutItem* or *UpdateItem* operations; if the comparison evaluates to true, the operation succeeds; if not, the operation fails. You can use *ExpectedAttributeValue* in one of two different ways:

- Use *AttributeValueList* to specify one or more values to compare against an attribute. Use *ComparisonOperator* to specify how you want to perform the comparison. If the comparison evaluates to true, then the conditional operation succeeds.
- Use *Value* to specify a value that DynamoDB will compare against an attribute. If the values match, then *ExpectedAttributeValue* evaluates to true and the conditional operation succeeds. Optionally, you can also set *Exists* to false, indicating that you *do not* expect to find the attribute value in the table. In this case, the conditional operation succeeds only if the comparison evaluates to false.

Value and *Exists* are incompatible with *AttributeValueList* and *ComparisonOperator* . Note that if you use both sets of parameters at once, DynamoDB will return a *ValidationException* exception.

- **Value** (valid *DynamoDB* type) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).
- **Exists** (boolean) –

Causes DynamoDB to evaluate the value before attempting a conditional operation:

- If *Exists* is `true` , DynamoDB will check to see if that attribute value already exists in the table. If it is found, then the operation succeeds. If it is not found, the operation fails with a *ConditionalCheckFailedException* .
- If *Exists* is `false` , DynamoDB assumes that the attribute value does not exist in the table. If in fact the value does not exist, then the assumption is valid and the operation succeeds. If the value is found, despite the assumption that it does not exist, the operation fails with a *ConditionalCheckFailedException* .

The default setting for *Exists* is `true` . If you supply a *Value* all by itself, DynamoDB assumes the attribute exists: You don't have to set *Exists* to `true` , because it is implied.

DynamoDB returns a *ValidationException* if:

- *Exists* is `true` but there is no *Value* to check. (You expect a value to exist, but don't specify what that value is.)
- *Exists* is `false` but you also provide a *Value* . (You cannot expect an attribute to have a value, while also expecting it not to exist.)

- **ComparisonOperator** (*string*) –

A comparator for evaluating attributes in the *AttributeValueList*. For example, equals, greater than, less than, etc.

The following comparison operators are available:

```
EQ | NE | LE | LT | GE | GT |
NOT_NULL | NULL | CONTAINS |
NOT_CONTAINS | BEGINS_WITH | IN |
BETWEEN
```

The following are descriptions of each comparison operator.

- **EQ** : Equal. **EQ** is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" }. Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] }.
- **NE** : Not equal. **NE** is supported for all datatypes, including lists and maps. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an *AttributeValue* of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" }. Also, { "N": "6" } does not equal { "NS": ["6", "2", "1"] }.
- **LE** : Less than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" }. Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] }.
- **LT** : Less than. *AttributeValueList* can contain only one *AttributeValue* of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" }. Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] }.
- **GE** : Greater than or equal. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S": "6" } does not equal { "N": "6" }. Also, { "N": "6" } does not compare to { "NS": ["6", "2", "1"] }.

- **GT** : Greater than. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not equal { "N" : "6" } . Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] } .
- **NOT_NULL** : The attribute exists. **NOT_NULL** is supported for all datatypes, including lists and maps. .. note::This operator tests for the existence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using **NOT_NULL** , the result is a Boolean *true* . This result is because the attribute “a ” exists; its data type is not relevant to the **NOT_NULL** comparison operator.
- **NULL** : The attribute does not exist. **NULL** is supported for all datatypes, including lists and maps. .. note::This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute “a ” is null, and you evaluate it using **NULL** , the result is a Boolean *false* . This is because the attribute “a ” exists; its data type is not relevant to the **NULL** comparison operator.
- **CONTAINS** : Checks for a subsequence, or value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS ”, “NS ”, or “BS ”), then the operator evaluates to true if it finds an exact match with any member of the set. **CONTAINS** is supported for lists: When evaluating “a **CONTAINS** b ”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
- **NOT_CONTAINS** : Checks for absence of a subsequence, or absence of a value in a set. *AttributeValueList* can contain only one *AttributeValue* element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set (“SS ”, “NS ”, or “BS ”), then the operator evaluates to true if it *does not* find an exact match with any member of the set. **NOT_CONTAINS** is supported for lists: When evaluating “a **NOT CONTAINS** b ”, “a ” can be a list; however, “b ” cannot be a set, a map, or a list.
- **BEGINS_WITH** : Checks for a prefix. *AttributeValueList* can contain only one *AttributeValue* of type

String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).

- **IN** : Checks for matching elements within two sets. *AttributeValueList* can contain one or more *AttributeValue* elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
- **BETWEEN** : Greater than or equal to the first value, and less than or equal to the second value. *AttributeValueList* must contain two *AttributeValue* elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an *AttributeValue* element of a different type than the one provided in the request, the value does not match. For example, { "S" : "6" } does not compare to { "N" : "6" } . Also, { "N" : "6" } does not compare to { "NS" : ["6", "2", "1"] }
- **AttributeValueList** (*list*) –

One or more values to evaluate against the supplied attribute. The number of values in the list depends on the *ComparisonOperator* being used.

For type Number, value comparisons are numeric.

String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A , and a is greater than B . For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters

For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.

For information on specifying data types in JSON, see [JSON Data Format](#) in the *Amazon DynamoDB Developer Guide* .

- (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).

• **ConditionalOperator** (*string*) –

Warning: This is a legacy parameter, for backward compatibility. New applications should use *ConditionExpression* instead. To combine legacy parameters and expression parameters in a call; otherwise, DynamoDB will return a *ValidationException*.

A logical operator to apply to the conditions in the *Expected* map:

- **AND** - If all of the conditions evaluate to true, then the entire map evaluates to true.

- OR - If at least one of the conditions evaluate to true, then the entire map evaluates to true.

If you omit *ConditionalOperator* , then AND is the default.

The operation will succeed only if the entire map evaluates to true.

Note: This parameter does not support attributes of type List or Map.

- **ReturnValues** (*string*) – Use *ReturnValues* if you want to get the item attributes as they appeared either before or after they were updated. For *UpdateItem* , the valid values are:
 - NONE - If *ReturnValues* is not specified, or if its value is NONE , then nothing is returned. (This setting is the default for *ReturnValues* .)
 - ALL_OLD - If *UpdateItem* overwrote an attribute name-value pair, then the content of the old item is returned.
 - UPDATED_OLD - The old versions of only the updated attributes are returned.
 - ALL_NEW - All of the attributes of the new version of the item are returned.
 - UPDATED_NEW - The new versions of only the updated attributes are returned.
- **ReturnConsumedCapacity** (*string*) – Determines the level of detail about provisioned throughput consumption that is returned in the response:
 - INDEXES - The response includes the aggregate *ConsumedCapacity* for the operation, together with *ConsumedCapacity* for each table and secondary index that was accessed. Note that some operations, such as *GetItem* and *BatchGetItem* , do not access any indexes at all. In these cases, specifying *INDEXES* will only return *ConsumedCapacity* information for table(s).
 - TOTAL - The response includes only the aggregate *ConsumedCapacity* for the operation.
 - NONE - No *ConsumedCapacity* details are included in the response.
- **ReturnItemCollectionMetrics** (*string*) – Determines whether item collection metrics are returned. If set to *SIZE* , the response includes statistics about item collections, if any, that were modified during the operation are returned in the response. If set to *NONE* (the default), no statistics are returned.
- **UpdateExpression** (*string*) – An expression that defines one or more attributes to be updated, the action to be performed on them, and new value(s) for them.

The following action values are available for *UpdateExpression* .

- SET - Adds one or more attributes and values to an item. If any of these attribute already exist, they are replaced by the new values. You can also use SET to add or subtract from an attribute that is of type Number. For example: SET myNum = myNum + :val SET supports the following functions: * if_not_exists (path, operand) - if the item does not contain an attribute at the specified path, then if_not_exists evaluates to operand; otherwise, it evaluates to path. You can use this function to avoid overwriting an attribute that may already be present in the item.
- list_append (operand, operand) - evaluates to a list with a new element added to it. You can append the new element to the start or the end of the list by reversing the order of the operands.

These function names are case-sensitive.

- REMOVE - Removes one or more attributes from an item.
- ADD - Adds the specified value to the item, if the attribute does not already exist. If the attribute does exist, then the behavior of ADD depends on the data type of the attribute:
 - * If the existing attribute is a number, and if *Value* is also a number, then *Value* is mathematically added to the existing attribute. If *Value* is a negative number, then it is subtracted from the existing attribute. .. note:: If you use ADD to increment or decrement a number value for an item that doesn't exist before the update, DynamoDB uses 0 as the initial value. Similarly, if you use ADD for an existing item to increment or decrement an attribute value that doesn't exist before the update, DynamoDB uses 0 as the initial value. For example, suppose that the item you want to update doesn't have an attribute named *itemcount*, but you decide to ADD the number 3 to this attribute anyway. DynamoDB will create the *itemcount* attribute, set its initial value to 0, and finally add 3 to it. The result will be a new *itemcount* attribute in the item, with a value of 3.
 - If the existing data type is a set and if *Value* is also a set, then *Value* is added to the existing set. For example, if the attribute value is the set [1, 2], and the ADD action specified [3], then the final attribute value is [1, 2, 3]. An error occurs if an ADD action is specified for a set attribute and the attribute type specified does not match the existing set type. Both sets must have the same primitive data type. For example, if the existing data type is a set of strings, the *Value* must also be a set of strings.

Warning: The ADD action only supports Number and set data types. In addition, ADD can only be used on top-level attributes, not nested attributes.

- DELETE - Deletes an element from a set. If a set of values is specified, then those values are subtracted from the old set. For example, if the attribute value was the set [a, b, c] and the DELETE action specifies [a, c], then the final attribute value is [b]. Specifying an empty set is an error. .. warning::The DELETE action only supports set data types. In addition, DELETE can only be used on top-level attributes, not nested attributes.

You can have many actions in a single expression, such as the following: SET a=:value1, b=:value2 DELETE :value3, :value4, :value5

For more information on update expressions, see [Modifying Items and Attributes](#) in the *Amazon DynamoDB Developer Guide*.

Note: *UpdateExpression* replaces the legacy *AttributeUpdates* parameter.

- **ConditionExpression** (condition from `boto3.dynamodb.conditions.Attr` method) – The condition(s) an attribute(s) must meet. Valid conditions are listed in the *DynamoDB Reference Guide*.
- **ExpressionAttributeNames** (*dict*) – One or more substitution tokens for attribute names in an expression. The following are some use cases for using *ExpressionAttributeNames*:
 - To access an attribute whose name conflicts with a DynamoDB reserved word.
 - To create a placeholder for repeating occurrences of an attribute

name in an expression.

- To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the # character in an expression to dereference an attribute name. For example, consider the following attribute name:

- Percentile

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see [Reserved Words](#) in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for *ExpressionAttributeNames* :

- {"#P": "Percentile"}

You could then use this substitution in an expression, as in this example:

- #P = :val

Note: Tokens that begin with the : character are *expression attribute values* , which are placeholders for the actual value at runtime.

For more information on expression attribute names, see [Accessing Item Attributes](#) in the *Amazon DynamoDB Developer Guide* .

- (string) -
- * (string) -

- **ExpressionAttributeValues** (*dict*) - One or more values that can be substituted in an expression.

Use the : (colon) character in an expression to dereference an attribute value. For example, suppose that you wanted to check whether the value of the *ProductStatus* attribute was one of the following:

Available | Backordered | Discontinued

You would first need to specify *ExpressionAttributeValues* as follows:

```
{ ":avail":{"S":"Available"},
  ":back":{"S":"Backordered"},
  ":disc":{"S":"Discontinued"} }
```

You could then use these values in an expression, such as this:

```
ProductStatus IN (:avail, :back, :disc)
```

For more information on expression attribute values, see [Specifying Conditions](#) in the *Amazon DynamoDB Developer Guide* .

- (string) -
- * (valid *DynamoDB* type) - - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).

Return type dict

Returns

Response Syntax

```
{
  'Attributes': {
    'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])|set
  },
  'ConsumedCapacity': {
    'TableName': 'string',
```

```

        'CapacityUnits': 123.0,
        'Table': {
            'CapacityUnits': 123.0
        },
        'LocalSecondaryIndexes': {
            'string': {
                'CapacityUnits': 123.0
            }
        },
        'GlobalSecondaryIndexes': {
            'string': {
                'CapacityUnits': 123.0
            }
        }
    },
    'ItemCollectionMetrics': {
        'ItemCollectionKey': {
            'string': 'string'|123|Binary(b'bytes')|True|None|set(['string'])
        },
        'SizeEstimateRangeGB': [
            123.0,
        ]
    }
}

```

Response Structure

- (*dict*) –

Represents the output of an *UpdateItem* operation.

- **Attributes** (*dict*) –

A map of attribute values as they appeared before the *UpdateItem* operation. This map only appears if *ReturnValues* was specified as something other than *NONE* in the request. Each element represents one attribute.

- * (*string*) –

- (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the *DynamoDB Reference Guide*.

- **ConsumedCapacity** (*dict*) –

The capacity units consumed by an operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. *ConsumedCapacity* is only returned if the request asked for it. For more information, see *Provisioned Throughput* in the *Amazon DynamoDB Developer Guide*.

- * **TableName** (*string*) –

The name of the table that was affected by the operation.

- * **CapacityUnits** (*float*) –

The total number of capacity units consumed by the operation.

- * **Table** (*dict*) –

The amount of throughput consumed on the table affected by the operation.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

* **LocalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each local index affected by the operation.

- (*string*) –
- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

* **GlobalSecondaryIndexes** (*dict*) –

The amount of throughput consumed on each global index affected by the operation.

- (*string*) –
- (*dict*) –

Represents the amount of provisioned throughput capacity consumed on a table or an index.

- **CapacityUnits** (*float*) –

The total number of capacity units consumed on a table or an index.

– **ItemCollectionMetrics** (*dict*) –

Information about item collections, if any, that were affected by the operation. *ItemCollectionMetrics* is only returned if the request asked for it. If the table does not have any local secondary indexes, this information is not returned in the response.

* **ItemCollectionKey** (*dict*) –

The hash key value of the item collection. This value is the same as the hash key of the item.

- (*string*) –
- (*valid DynamoDB type*) – - The value of the attribute. The valid value types are listed in the [DynamoDB Reference Guide](#).

* **SizeEstimateRangeGB** (*list*) –

An estimate of item collection size, in gigabytes. This value is a two-element array containing a lower bound and an upper bound for the estimate. The estimate includes the size of all the items in the table, plus the size of all attributes projected into all of the local secondary indexes on that table. Use this estimate to measure whether a local secondary index is approaching its size limit.

The estimate is subject to change over time; therefore, do not rely on the precision or accuracy of the estimate.

- (*float*) –

DynamoDBStreams

Table of Contents

- DynamoDBStreams
 - Client

Client**class** `DynamoDBStreams.Client`

A low-level client representing Amazon DynamoDB Streams:

```
import boto3
```

```
client = boto3.client('dynamodbstreams')
```

These are the available methods:

- `can_paginate()`
- `describe_stream()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_records()`
- `get_shard_iterator()`
- `get_waiter()`
- `list_streams()`

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

describe_stream (***kwargs*)

Returns information about a stream, including the current status of the stream, its Amazon Resource Name (ARN), the composition of its shards, and its corresponding DynamoDB table.

Note: You can call *DescribeStream* at a maximum rate of 10 times per second.

Each shard in the stream has a `SequenceNumberRange` associated with it. If the `SequenceNumberRange` has a `StartingSequenceNumber` but no `EndingSequenceNumber`, then the shard is still open (able to receive more stream records). If both `StartingSequenceNumber` and `EndingSequenceNumber` are present, the that shared is closed and can no longer receive more data.

Request Syntax

```
response = client.describe_stream(
    StreamArn='string',
    Limit=123,
    ExclusiveStartShardId='string'
)
```

Parameters

- **StreamArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) for the stream.

- **Limit** (*integer*) – The maximum number of shard objects to return. The upper limit is 100.
- **ExclusiveStartShardId** (*string*) – The shard ID of the first item that this operation will evaluate. Use the value that was returned for `LastEvaluatedShardId` in the previous operation.

Return type dict

Returns

Response Syntax

```
{
  'StreamDescription': {
    'StreamArn': 'string',
    'StreamLabel': 'string',
    'StreamStatus': 'ENABLING'|'ENABLED'|'DISABLING'|'DISABLED',
    'StreamViewType': 'NEW_IMAGE'|'OLD_IMAGE'|'NEW_AND_OLD_IMAGES'|'KEYS_ONLY',
    'CreationRequestDateTime': datetime(2015, 1, 1),
    'TableName': 'string',
    'KeySchema': [
      {
        'AttributeName': 'string',
        'KeyType': 'HASH'|'RANGE'
      },
    ],
    'Shards': [
      {
        'ShardId': 'string',
        'SequenceNumberRange': {
          'StartingSequenceNumber': 'string',
          'EndingSequenceNumber': 'string'
        },
        'ParentShardId': 'string'
      },
    ],
    'LastEvaluatedShardId': 'string'
  }
}
```

Response Structure

- (*dict*) –

Represents the output of a *DescribeStream* operation.

- **StreamDescription** (*dict*) –

A complete description of the stream, including its creation date and time, the DynamoDB table associated with the stream, the shard IDs within the stream, and the beginning and ending sequence numbers of stream records within the shards.

- * **StreamArn** (*string*) –

The Amazon Resource Name (ARN) for the stream.

- * **StreamLabel** (*string*) –

A timestamp, in ISO 8601 format, for this stream.

Note that *LatestStreamLabel* is not a unique identifier for the stream, because it is possible that a stream from another table might have the

same timestamp. However, the combination of the following three elements is guaranteed to be unique:

- the AWS customer ID.
- the table name
- the *StreamLabel*

* **StreamStatus** (*string*) –

Indicates the current status of the stream:

- **ENABLING** - Streams is currently being enabled on the DynamoDB table.
- **ENABLING** - the stream is enabled.
- **DISABLING** - Streams is currently being disabled on the DynamoDB table.
- **DISABLED** - the stream is disabled.

* **StreamViewType** (*string*) –

Indicates the format of the records within this stream:

- **KEYS_ONLY** - only the key attributes of items that were modified in the DynamoDB table.
- **NEW_IMAGE** - entire item from the table, as it appeared after they were modified.
- **OLD_IMAGE** - entire item from the table, as it appeared before they were modified.
- **NEW_AND_OLD_IMAGES** - both the new and the old images of the items from the table.

* **CreationRequestDateTime** (*datetime*) –

The date and time when the request to create this stream was issued.

* **TableName** (*string*) –

The DynamoDB table with which the stream is associated.

* **KeySchema** (*list*) –

The key attribute(s) of the stream's DynamoDB table.

- (*dict*) –

Represents *a single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A *KeySchemaElement* represents exactly one attribute of the primary key. For example, a hash type primary key would be represented by one *KeySchemaElement*. A hash-and-range type primary key would require one *KeySchemaElement* for the hash attribute, and another *KeySchemaElement* for the range attribute.

- **AttributeName** (*string*) –

The name of a key attribute.

- **KeyType** (*string*) –

The attribute data, consisting of the data type and the attribute value itself.

* **Shards** (*list*) –

The shards that comprise the stream.

- (*dict*) –

A uniquely identified group of stream records within a stream.

- **ShardId** (*string*) –
The system-generated identifier for this shard.
- **SequenceNumberRange** (*dict*) –
The range of possible sequence numbers for the shard.
- **StartingSequenceNumber** (*string*) –
The first sequence number.
- **EndingSequenceNumber** (*string*) –
The last sequence number.
- **ParentShardId** (*string*) –
The shard ID of the current shard's parent.

* **LastEvaluatedShardId** (*string*) –

The shard ID of the item where the operation stopped, inclusive of the previous result set. Use this value to start a new operation, excluding this value in the new request.

If `LastEvaluatedShardId` is empty, then the “last page” of results has been processed and there is currently no more data to be retrieved.

If `LastEvaluatedShardId` is not empty, it does not necessarily mean that there is more data in the result set. The only way to know when you have reached the end of the result set is when `LastEvaluatedShardId` is empty.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_records (***kwargs*)

Retrieves the stream records from a given shard.

Specify a shard iterator using the `ShardIterator` parameter. The shard iterator specifies the position in the shard from which you want to start reading stream records sequentially. If there are no stream records available in the portion of the shard that the iterator points to, `GetRecords` returns an empty list. Note that it might take multiple calls to get to a portion of the shard that contains stream records.

Note: GetRecordscan retrieve a maximum of 1 MB of data or 2000 stream records, whichever comes first.

Request Syntax

```
response = client.get_records(
    ShardIterator='string',
    Limit=123
)
```

Parameters

- **ShardIterator** (*string*) – **[REQUIRED]**

A shard iterator that was retrieved from a previous GetShardIterator operation. This iterator can be used to access the stream records in this shard.

- **Limit** (*integer*) – The maximum number of records to return from the shard. The upper limit is 1000.

Return type dict

Returns

Response Syntax

```
{
    'Records': [
        {
            'eventID': 'string',
            'eventName': 'INSERT'|'MODIFY'|'REMOVE',
            'eventVersion': 'string',
            'eventSource': 'string',
            'awsRegion': 'string',
            'dynamodb': {
                'Keys': {
                    'string': {
                        'S': 'string',
                        'N': 'string',
                        'B': b'bytes',
                        'SS': [
                            'string',
                        ],
                        'NS': [
                            'string',
                        ],
                        'BS': [
                            b'bytes',
                        ],
                        'M': {
                            'string': {'... recursive ...'}
                        },
                        'L': [
                            {'... recursive ...'},
                        ],
                        'NULL': True|False,
                        'BOOL': True|False
                    }
                },
            },
            'NewImage': {
                'string': {
```

```
        'S': 'string',
        'N': 'string',
        'B': b'bytes',
        'SS': [
            'string',
        ],
        'NS': [
            'string',
        ],
        'BS': [
            b'bytes',
        ],
        'M': {
            'string': {'... recursive ...'}
        },
        'L': [
            {'... recursive ...'},
        ],
        'NULL': True|False,
        'BOOL': True|False
    }
},
'OldImage': {
    'string': {
        'S': 'string',
        'N': 'string',
        'B': b'bytes',
        'SS': [
            'string',
        ],
        'NS': [
            'string',
        ],
        'BS': [
            b'bytes',
        ],
        'M': {
            'string': {'... recursive ...'}
        },
        'L': [
            {'... recursive ...'},
        ],
        'NULL': True|False,
        'BOOL': True|False
    }
},
'SequenceNumber': 'string',
'SizeBytes': 123,
'StreamViewType': 'NEW_IMAGE' | 'OLD_IMAGE' | 'NEW_AND_OLD_IMAGES' | 'KEYS_ONLY',
},
},
],
'NextShardIterator': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a *GetRecords* operation.

– **Records** (*list*) –

The stream records from the shard, which were retrieved using the shard iterator.

* (*dict*) –

A description of a unique event within a stream.

· **eventID** (*string*) –

A globally unique identifier for the event that was recorded in this stream record.

· **eventName** (*string*) –

The type of data modification that was performed on the DynamoDB table:

- INSERT - a new item was added to the table.
- MODIFY - one or more of the item's attributes were updated.
- REMOVE - the item was deleted from the table

· **eventVersion** (*string*) –

The version number of the stream record format. Currently, this is *1.0*.

· **eventSource** (*string*) –

The AWS service from which the stream record originated. For DynamoDB Streams, this is *aws:dynamodb*.

· **awsRegion** (*string*) –

The region in which the *GetRecords* request was received.

· **dynamodb** (*dict*) –

The main body of the stream record, containing all of the DynamoDB-specific fields.

· **Keys** (*dict*) –

The primary key attribute(s) for the DynamoDB item that was modified.

· (*string*) –

· (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

· **S** (*string*) –

A String data type.

· **N** (*string*) –

A Number data type.

· **B** (*bytes*) –

A Binary data type.

· **SS** (*list*) –

A String Set data type.

· (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –
- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –
- **M** (*dict*) –

A Map data type.

- (*string*) –
- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List data type.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **NewImage** (*dict*) –

The item in the DynamoDB table as it appeared after it was modified.

- (*string*) –
- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –
- **NS** (*list*) –

A Number Set data type.

- (*string*) –
- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –
- **M** (*dict*) –

A Map data type.

- (*string*) –
- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List data type.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **OldImage** (*dict*) –

The item in the DynamoDB table as it appeared before it was modified.

- (*string*) –
- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book

item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **S** (*string*) –

A String data type.

- **N** (*string*) –

A Number data type.

- **B** (*bytes*) –

A Binary data type.

- **SS** (*list*) –

A String Set data type.

- (*string*) –

- **NS** (*list*) –

A Number Set data type.

- (*string*) –

- **BS** (*list*) –

A Binary Set data type.

- (*bytes*) –

- **M** (*dict*) –

A Map data type.

- (*string*) –

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **L** (*list*) –

A List data type.

- (*dict*) –

Represents the data for an attribute. You can set one, and only one, of the elements.

Each attribute in an item is a name-value pair. An attribute can be single-valued or multi-valued set. For example, a book item can have title and authors attributes. Each book has one title but can have many authors. The multi-valued attribute is a set; duplicate values are not allowed.

- **NULL** (*boolean*) –

A Null data type.

- **BOOL** (*boolean*) –

A Boolean data type.

- **SequenceNumber** (*string*) –

The sequence number of the stream record.

- **SizeBytes** (*integer*) –

The size of the stream record, in bytes.

- **StreamViewType** (*string*) –

The type of data from the modified DynamoDB item that was captured in this stream record:

- **KEYS_ONLY** - only the key attributes of the modified item.
- **NEW_IMAGE** - the entire item, as it appears after it was modified.
- **OLD_IMAGE** - the entire item, as it appeared before it was modified.
- **NEW_AND_OLD_IMAGES** — both the new and the old item images of the item.

- **NextShardIterator** (*string*) –

The next position in the shard from which to start sequentially reading stream records. If set to `null`, the shard has been closed and the requested iterator will not return any more data.

get_shard_iterator (***kwargs*)

Returns a shard iterator. A shard iterator provides information about how to retrieve the stream records from within a shard. Use the shard iterator in a subsequent `GetRecords` request to read the stream records from the shard.

Note: A shard iterator expires 15 minutes after it is returned to the requester.

Request Syntax

```
response = client.get_shard_iterator(
    StreamArn='string',
    ShardId='string',
    ShardIteratorType='TRIM_HORIZON' | 'LATEST' | 'AT_SEQUENCE_NUMBER' | 'AFTER_SEQUENCE_NUMBER',
    SequenceNumber='string'
)
```

Parameters

- **StreamArn** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) for the stream.

- **ShardId** (*string*) – **[REQUIRED]**

The identifier of the shard. The iterator will be returned for this shard ID.

- **ShardIteratorType** (*string*) – **[REQUIRED]**

Determines how the shard iterator is used to start reading stream records from the shard:

- **AT_SEQUENCE_NUMBER** - Start reading exactly from the position denoted by a specific sequence number.
- **AFTER_SEQUENCE_NUMBER** - Start reading right after the position denoted by a specific sequence number.
- **TRIM_HORIZON** - Start reading at the last (untrimmed) stream record, which is the oldest record in the shard. In DynamoDB Streams, there is a 24 hour limit on data retention. Stream records whose age exceeds this limit are subject to removal (trimming) from the stream.
- **LATEST** - Start reading just after the most recent stream record in the shard, so that you always read the most recent data in the shard.

- **SequenceNumber** (*string*) – The sequence number of a stream record in the shard from which to start reading.

Return type dict

Returns

Response Syntax

```
{
    'ShardIterator': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a *GetShardIterator* operation.

- **ShardIterator** (*string*) –

The position in the shard from which to start reading stream records sequentially. A shard iterator specifies this position using the sequence number of a stream record in a shard.

get_waiter (*waiter_name*)

list_streams (***kwargs*)

Returns an array of stream ARNs associated with the current account and endpoint. If the *TableName* parameter is present, then *ListStreams* will return only the streams ARNs for that table.

Note: You can call *ListStreams* at a maximum rate of 5 times per second.

Request Syntax

```
response = client.list_streams(
    TableName='string',
    Limit=123,
    ExclusiveStartStreamArn='string'
)
```

Parameters

- **TableName** (*string*) – If this parameter is provided, then only the streams associated with this table name are returned.
- **Limit** (*integer*) – The maximum number of streams to return. The upper limit is 100.
- **ExclusiveStartStreamArn** (*string*) – The ARN (Amazon Resource Name) of the first item that this operation will evaluate. Use the value that was returned for *LastEvaluatedStreamArn* in the previous operation.

Return type dict

Returns

Response Syntax

```
{
    'Streams': [
        {
            'StreamArn': 'string',
            'TableName': 'string',
            'StreamLabel': 'string'
        },
    ],
}
```

```
    'LastEvaluatedStreamArn': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a *ListStreams* operation.

- **Streams** (*list*) –

A list of stream descriptors associated with the current account and end-point.

- * (*dict*) –

Represents all of the data describing a particular stream.

- **StreamArn** (*string*) –

The Amazon Resource Name (ARN) for the stream.

- **TableName** (*string*) –

The DynamoDB table with which the stream is associated.

- **StreamLabel** (*string*) –

A timestamp, in ISO 8601 format, for this stream.

Note that *LatestStreamLabel* is not a unique identifier for the stream, because it is possible that a stream from another table might have the same timestamp. However, the combination of the following three elements is guaranteed to be unique:

- the AWS customer ID.
- the table name
- the *StreamLabel*

- **LastEvaluatedStreamArn** (*string*) –

The stream ARN of the item where the operation stopped, inclusive of the previous result set. Use this value to start a new operation, excluding this value in the new request.

If *LastEvaluatedStreamArn* is empty, then the “last page” of results has been processed and there is no more data to be retrieved.

If *LastEvaluatedStreamArn* is not empty, it does not necessarily mean that there is more data in the result set. The only way to know when you have reached the end of the result set is when *LastEvaluatedStreamArn* is empty.

EC2

Table of Contents

- **EC2**
 - Client
 - Paginators
 - Waiters
 - Service Resource
 - DhcpOptions
 - Image
 - Instance
 - InternetGateway
 - KeyPair
 - NetworkAcl
 - NetworkInterface
 - PlacementGroup
 - RouteTable
 - RouteTableAssociation
 - SecurityGroup
 - Snapshot
 - Subnet
 - Tag
 - Volume
 - Vpc
 - VpcPeeringConnection

Client**class EC2.Client**

A low-level client representing Amazon Elastic Compute Cloud (EC2):

```
import boto3

client = boto3.client('ec2')
```

These are the available methods:

- `accept_vpc_peering_connection()`
- `allocate_address()`
- `assign_private_ip_addresses()`
- `associate_address()`
- `associate_dhcp_options()`
- `associate_route_table()`
- `attach_classic_link_vpc()`
- `attach_internet_gateway()`
- `attach_network_interface()`
- `attach_volume()`
- `attach_vpn_gateway()`
- `authorize_security_group_egress()`
- `authorize_security_group_ingress()`
- `bundle_instance()`
- `can_paginate()`
- `cancel_bundle_task()`
- `cancel_conversion_task()`
- `cancel_export_task()`

- cancel_import_task()
- cancel_reserved_instances_listing()
- cancel_spot_fleet_requests()
- cancel_spot_instance_requests()
- confirm_product_instance()
- copy_image()
- copy_snapshot()
- create_customer_gateway()
- create_dhcp_options()
- create_flow_logs()
- create_image()
- create_instance_export_task()
- create_internet_gateway()
- create_key_pair()
- create_network_acl()
- create_network_acl_entry()
- create_network_interface()
- create_placement_group()
- create_reserved_instances_listing()
- create_route()
- create_route_table()
- create_security_group()
- create_snapshot()
- create_spot_datafeed_subscription()
- create_subnet()
- create_tags()
- create_volume()
- create_vpc()
- create_vpc_endpoint()
- create_vpc_peering_connection()
- create_vpn_connection()
- create_vpn_connection_route()
- create_vpn_gateway()
- delete_customer_gateway()
- delete_dhcp_options()
- delete_flow_logs()
- delete_internet_gateway()
- delete_key_pair()
- delete_network_acl()
- delete_network_acl_entry()
- delete_network_interface()
- delete_placement_group()
- delete_route()
- delete_route_table()
- delete_security_group()
- delete_snapshot()
- delete_spot_datafeed_subscription()
- delete_subnet()
- delete_tags()
- delete_volume()
- delete_vpc()
- delete_vpc_endpoints()
- delete_vpc_peering_connection()
- delete_vpn_connection()

- `delete_vpn_connection_route()`
- `delete_vpn_gateway()`
- `deregister_image()`
- `describe_account_attributes()`
- `describe_addresses()`
- `describe_availability_zones()`
- `describe_bundle_tasks()`
- `describe_classic_link_instances()`
- `describe_conversion_tasks()`
- `describe_customer_gateways()`
- `describe_dhcp_options()`
- `describe_export_tasks()`
- `describe_flow_logs()`
- `describe_image_attribute()`
- `describe_images()`
- `describe_import_image_tasks()`
- `describe_import_snapshot_tasks()`
- `describe_instance_attribute()`
- `describe_instance_status()`
- `describe_instances()`
- `describe_internet_gateways()`
- `describe_key_pairs()`
- `describe_moving_addresses()`
- `describe_network_acls()`
- `describe_network_interface_attribute()`
- `describe_network_interfaces()`
- `describe_placement_groups()`
- `describe_prefix_lists()`
- `describe_regions()`
- `describe_reserved_instances()`
- `describe_reserved_instances_listings()`
- `describe_reserved_instances_modifications()`
- `describe_reserved_instances_offerings()`
- `describe_route_tables()`
- `describe_security_groups()`
- `describe_snapshot_attribute()`
- `describe_snapshots()`
- `describe_spot_datafeed_subscription()`
- `describe_spot_fleet_instances()`
- `describe_spot_fleet_request_history()`
- `describe_spot_fleet_requests()`
- `describe_spot_instance_requests()`
- `describe_spot_price_history()`
- `describe_subnets()`
- `describe_tags()`
- `describe_volume_attribute()`
- `describe_volume_status()`
- `describe_volumes()`
- `describe_vpc_attribute()`
- `describe_vpc_classic_link()`
- `describe_vpc_endpoint_services()`
- `describe_vpc_endpoints()`
- `describe_vpc_peering_connections()`
- `describe_vpcs()`

- describe_vpn_connections()
- describe_vpn_gateways()
- detach_classic_link_vpc()
- detach_internet_gateway()
- detach_network_interface()
- detach_volume()
- detach_vpn_gateway()
- disable_vgw_route_propagation()
- disable_vpc_classic_link()
- disassociate_address()
- disassociate_route_table()
- enable_vgw_route_propagation()
- enable_volume_io()
- enable_vpc_classic_link()
- generate_presigned_url()
- get_console_output()
- get_paginator()
- get_password_data()
- get_waiter()
- import_image()
- import_instance()
- import_key_pair()
- import_snapshot()
- import_volume()
- modify_image_attribute()
- modify_instance_attribute()
- modify_network_interface_attribute()
- modify_reserved_instances()
- modify_snapshot_attribute()
- modify_subnet_attribute()
- modify_volume_attribute()
- modify_vpc_attribute()
- modify_vpc_endpoint()
- monitor_instances()
- move_address_to_vpc()
- purchase_reserved_instances_offering()
- reboot_instances()
- register_image()
- reject_vpc_peering_connection()
- release_address()
- replace_network_acl_association()
- replace_network_acl_entry()
- replace_route()
- replace_route_table_association()
- report_instance_status()
- request_spot_fleet()
- request_spot_instances()
- reset_image_attribute()
- reset_instance_attribute()
- reset_network_interface_attribute()
- reset_snapshot_attribute()
- restore_address_to_classic()
- revoke_security_group_egress()
- revoke_security_group_ingress()

- `run_instances()`
- `start_instances()`
- `stop_instances()`
- `terminate_instances()`
- `unassign_private_ip_addresses()`
- `unmonitor_instances()`

accept_vpc_peering_connection (***kwargs*)

Accept a VPC peering connection request. To accept a request, the VPC peering connection must be in the `pending-acceptance` state, and you must be the owner of the peer VPC. Use the `DescribeVpcPeeringConnections` request to view your outstanding VPC peering connection requests.

Request Syntax

```
response = client.accept_vpc_peering_connection(
    DryRun=True|False,
    VpcPeeringConnectionId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcPeeringConnectionId** (*string*) – The ID of the VPC peering connection.

Return type dict

Returns**Response Syntax**

```
{
    'VpcPeeringConnection': {
        'AcceptorVpcInfo': {
            'CidrBlock': 'string',
            'OwnerId': 'string',
            'VpcId': 'string'
        },
        'ExpirationTime': datetime(2015, 1, 1),
        'RequesterVpcInfo': {
            'CidrBlock': 'string',
            'OwnerId': 'string',
            'VpcId': 'string'
        },
        'Status': {
            'Code': 'initiating-request'|'pending-acceptance'|'active'|'deleted'|'
            'Message': 'string'
        },
        'Tags': [
            {
                'Key': 'string',
                'Value': 'string'
            },
        ],
        'VpcPeeringConnectionId': 'string'
    }
}
```

Response Structure

- (*dict*) –
 - **VpcPeeringConnection** (*dict*) –

Information about the VPC peering connection.

 - * **AccepterVpcInfo** (*dict*) –

The information of the peer VPC.

 - **CidrBlock** (*string*) –

The CIDR block for the VPC.
 - **OwnerId** (*string*) –

The AWS account ID of the VPC owner.
 - **VpcId** (*string*) –

The ID of the VPC.
 - * **ExpirationTime** (*datetime*) –

The time that an unaccepted VPC peering connection will expire.
 - * **RequesterVpcInfo** (*dict*) –

The information of the requester VPC.

 - **CidrBlock** (*string*) –

The CIDR block for the VPC.
 - **OwnerId** (*string*) –

The AWS account ID of the VPC owner.
 - **VpcId** (*string*) –

The ID of the VPC.
 - * **Status** (*dict*) –

The status of the VPC peering connection.

 - **Code** (*string*) –

The status of the VPC peering connection.
 - **Message** (*string*) –

A message that provides more information about the status, if applicable.
 - * **Tags** (*list*) –

Any tags assigned to the resource.

 - (*dict*) –

Describes a tag.
 - **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`
 - **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.
 - * **VpcPeeringConnectionId** (*string*) –

The ID of the VPC peering connection.

allocate_address (**kwargs)

Acquires an Elastic IP address.

An Elastic IP address is for use either in the EC2-Classic platform or in a VPC. For more information, see [Elastic IP Addresses](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.allocate_address(
    DryRun=True|False,
    Domain='vpc'|'standard'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Domain** (*string*) – Set to `vpc` to allocate the address for use with instances in a VPC.

Default: The address is for use with instances in EC2-Classic.

Return type dict

Returns**Response Syntax**

```
{
    'PublicIp': 'string',
    'Domain': 'vpc'|'standard',
    'AllocationId': 'string'
}
```

Response Structure

- (*dict*) –
 - **PublicIp** (*string*) –
The Elastic IP address.
 - **Domain** (*string*) –
Indicates whether this Elastic IP address is for use with instances in EC2-Classic (`standard`) or instances in a VPC (`vpc`).
 - **AllocationId** (*string*) –
[EC2-VPC] The ID that AWS assigns to represent the allocation of the Elastic IP address for use with instances in a VPC.

assign_private_ip_addresses (**kwargs)

Assigns one or more secondary private IP addresses to the specified network interface. You can specify one or more specific secondary IP addresses, or you can specify the number of secondary IP addresses to be automatically assigned within the subnet's CIDR block range. The number of secondary IP addresses that you can assign to an instance varies by instance type. For information about instance types, see [Instance Types](#) in the *Amazon Elastic Compute Cloud User Guide*. For more information about Elastic IP addresses, see [Elastic IP Addresses](#) in the *Amazon Elastic Compute Cloud User Guide*.

AssignPrivateIpAddresses is available only in EC2-VPC.

Request Syntax

```
response = client.assign_private_ip_addresses(
    NetworkInterfaceId='string',
    PrivateIpAddresses=[
        'string',
    ],
    SecondaryPrivateIpAddressCount=123,
    AllowReassignment=True|False
)
```

Parameters

- **NetworkInterfaceId** (*string*) – [REQUIRED]

The ID of the network interface.

- **PrivateIpAddresses** (*list*) – One or more IP addresses to be assigned as a secondary private IP address to the network interface. You can't specify this parameter when also specifying a number of secondary IP addresses.

If you don't specify an IP address, Amazon EC2 automatically selects an IP address within the subnet range.

– (*string*) –

- **SecondaryPrivateIpAddressCount** (*integer*) – The number of secondary IP addresses to assign to the network interface. You can't specify this parameter when also specifying private IP addresses.
- **AllowReassignment** (*boolean*) – Indicates whether to allow an IP address that is already assigned to another network interface or instance to be reassigned to the specified network interface.

Returns None

associate_address (***kwargs*)

Associates an Elastic IP address with an instance or a network interface.

An Elastic IP address is for use in either the EC2-Classic platform or in a VPC. For more information, see [Elastic IP Addresses](#) in the *Amazon Elastic Compute Cloud User Guide*.

[EC2-Classic, VPC in an EC2-VPC-only account] If the Elastic IP address is already associated with a different instance, it is disassociated from that instance and associated with the specified instance.

[VPC in an EC2-Classic account] If you don't specify a private IP address, the Elastic IP address is associated with the primary IP address. If the Elastic IP address is already associated with a different instance or a network interface, you get an error unless you allow reassociation.

This is an idempotent operation. If you perform the operation more than once, Amazon EC2 doesn't return an error.

Request Syntax

```
response = client.associate_address(
    DryRun=True|False,
    InstanceId='string',
    PublicIp='string',
    AllocationId='string',
    NetworkInterfaceId='string',
    PrivateIpAddress='string',
    AllowReassociation=True|False
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If

you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **InstanceId** (*string*) – The ID of the instance. This is required for EC2-Classic. For EC2-VPC, you can specify either the instance ID or the network interface ID, but not both. The operation fails if you specify an instance ID unless exactly one network interface is attached.
- **PublicIp** (*string*) – The Elastic IP address. This is required for EC2-Classic.
- **AllocationId** (*string*) – [EC2-VPC] The allocation ID. This is required for EC2-VPC.
- **NetworkInterfaceId** (*string*) – [EC2-VPC] The ID of the network interface. If the instance has more than one network interface, you must specify a network interface ID.
- **PrivateIpAddress** (*string*) – [EC2-VPC] The primary or secondary private IP address to associate with the Elastic IP address. If no private IP address is specified, the Elastic IP address is associated with the primary private IP address.
- **AllowReassociation** (*boolean*) – [EC2-VPC] Allows an Elastic IP address that is already associated with an instance or network interface to be re-associated with the specified instance or network interface. Otherwise, the operation fails.

Default: `false`

Return type dict

Returns

Response Syntax

```
{
    'AssociationId': 'string'
}
```

Response Structure

- (*dict*) –
 - **AssociationId** (*string*) –
[EC2-VPC] The ID that represents the association of the Elastic IP address with an instance.

associate_dhcp_options (***kwargs*)

Associates a set of DHCP options (that you've previously created) with the specified VPC, or associates no DHCP options with the VPC.

After you associate the options with the VPC, any existing instances and all new instances that you launch in that VPC use the options. You don't need to restart or relaunch the instances. They automatically pick up the changes within a few hours, depending on how frequently the instance renews its DHCP lease. You can explicitly renew the lease using the operating system on the instance.

For more information, see [DHCP Options Sets](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.associate_dhcp_options(
    DryRun=True|False,
    DhcpOptionsId='string',
    VpcId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If

you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **DhcpOptionsId** (*string*) – [REQUIRED]

The ID of the DHCP options set, or default to associate no DHCP options with the VPC.

- **VpcId** (*string*) – [REQUIRED]

The ID of the VPC.

Returns None

associate_route_table (***kwargs*)

Associates a subnet with a route table. The subnet and route table must be in the same VPC. This association causes traffic originating from the subnet to be routed according to the routes in the route table. The action returns an association ID, which you need in order to disassociate the route table from the subnet later. A route table can be associated with multiple subnets.

For more information about route tables, see [Route Tables](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.associate_route_table(
    DryRun=True|False,
    SubnetId='string',
    RouteTableId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **SubnetId** (*string*) – [REQUIRED]

The ID of the subnet.

- **RouteTableId** (*string*) – [REQUIRED]

The ID of the route table.

Return type dict

Returns

Response Syntax

```
{
    'AssociationId': 'string'
}
```

Response Structure

- (*dict*) –
 - **AssociationId** (*string*) –

The route table association ID (needed to disassociate the route table).

attach_classic_link_vpc (***kwargs*)

Links an EC2-Classical instance to a ClassicLink-enabled VPC through one or more of the VPC's security groups. You cannot link an EC2-Classical instance to more than one VPC at a time. You can only link an instance that's in the `running` state. An instance is automatically unlinked from a VPC when it's stopped - you can link it to the VPC again when you restart it.

After you've linked an instance, you cannot change the VPC security groups that are associated with it. To change the security groups, you must first unlink the instance, and then link it again.

Linking your instance to a VPC is sometimes referred to as *attaching* your instance.

Request Syntax

```
response = client.attach_classic_link_vpc(
    DryRun=True|False,
    InstanceId='string',
    VpcId='string',
    Groups=[
        'string',
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceId** (*string*) – [REQUIRED]
The ID of an EC2-Classic instance to link to the ClassicLink-enabled VPC.
- **VpcId** (*string*) – [REQUIRED]
The ID of a ClassicLink-enabled VPC.
- **Groups** (*list*) – [REQUIRED]
The ID of one or more of the VPC's security groups. You cannot specify security groups from a different VPC.
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'Return': True|False
}
```

Response Structure

- (*dict*) –
 - **Return** (*boolean*) –
Returns `true` if the request succeeds; otherwise, it returns an error.

attach_internet_gateway (***kwargs*)

Attaches an Internet gateway to a VPC, enabling connectivity between the Internet and the VPC. For more information about your VPC and Internet gateway, see the [Amazon Virtual Private Cloud User Guide](#).

Request Syntax

```
response = client.attach_internet_gateway(
    DryRun=True|False,
    InternetGatewayId='string',
    VpcId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InternetGatewayId** (*string*) – [REQUIRED]
The ID of the Internet gateway.
- **VpcId** (*string*) – [REQUIRED]
The ID of the VPC.

Returns None

attach_network_interface (***kwargs*)

Attaches a network interface to an instance.

Request Syntax

```
response = client.attach_network_interface(
    DryRun=True|False,
    NetworkInterfaceId='string',
    InstanceId='string',
    DeviceIndex=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **NetworkInterfaceId** (*string*) – [REQUIRED]
The ID of the network interface.
- **InstanceId** (*string*) – [REQUIRED]
The ID of the instance.
- **DeviceIndex** (*integer*) – [REQUIRED]
The index of the device for the network interface attachment.

Return type dict

Returns

Response Syntax

```
{
    'AttachmentId': 'string'
}
```

Response Structure

- (*dict*) –
 - **AttachmentId** (*string*) –
The ID of the network interface attachment.

attach_volume (***kwargs*)

Attaches an EBS volume to a running or stopped instance and exposes it to the instance with the specified device name.

Encrypted EBS volumes may only be attached to instances that support Amazon EBS encryption. For more information, see [Amazon EBS Encryption](#) in the *Amazon Elastic Compute Cloud User Guide*.

For a list of supported device names, see [Attaching an EBS Volume to an Instance](#) . Any device names that aren't reserved for instance store volumes can be used for EBS volumes. For more information, see [Amazon EC2 Instance Store](#) in the *Amazon Elastic Compute Cloud User Guide* .

If a volume has an AWS Marketplace product code:

- The volume can be attached only to a stopped instance.
- AWS Marketplace product codes are copied from the volume to the instance.
- You must be subscribed to the product.
- The instance type and operating system of the instance must support the product. For example, you can't detach a volume from a Windows instance and attach it to a Linux instance.

For an overview of the AWS Marketplace, see [Introducing AWS Marketplace](#) .

For more information about EBS volumes, see [Attaching Amazon EBS Volumes](#) in the *Amazon Elastic Compute Cloud User Guide* .

Request Syntax

```
response = client.attach_volume(  
    DryRun=True|False,  
    VolumeId='string',  
    InstanceId='string',  
    Device='string'  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation` . Otherwise, it is `UnauthorizedOperation` .
- **VolumeId** (*string*) – **[REQUIRED]**
The ID of the EBS volume. The volume and instance must be within the same Availability Zone.
- **InstanceId** (*string*) – **[REQUIRED]**
The ID of the instance.
- **Device** (*string*) – **[REQUIRED]**
The device name to expose to the instance (for example, `/dev/sdh` or `xvdh`).

Return type dict

Returns

Response Syntax

```
{  
    'VolumeId': 'string',  
    'InstanceId': 'string',  
    'Device': 'string',  
    'State': 'attaching'|'attached'|'detaching'|'detached',  
    'AttachTime': datetime(2015, 1, 1),  
    'DeleteOnTermination': True|False  
}
```

Response Structure

- (*dict*) –
Information about the volume attachment.
 - **VolumeId** (*string*) –
The ID of the volume.

- **InstanceId** (*string*) –
The ID of the instance.
- **Device** (*string*) –
The device name.
- **State** (*string*) –
The attachment state of the volume.
- **AttachTime** (*datetime*) –
The time stamp when the attachment initiated.
- **DeleteOnTermination** (*boolean*) –
Indicates whether the EBS volume is deleted on instance termination.

attach_vpn_gateway (***kwargs*)

Attaches a virtual private gateway to a VPC. For more information, see [Adding a Hardware Virtual Private Gateway to Your VPC](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.attach_vpn_gateway(
    DryRun=True|False,
    VpnGatewayId='string',
    VpcId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpnGatewayId** (*string*) – **[REQUIRED]**
The ID of the virtual private gateway.
- **VpcId** (*string*) – **[REQUIRED]**
The ID of the VPC.

Return type dict

Returns

Response Syntax

```
{
  'VpcAttachment': {
    'VpcId': 'string',
    'State': 'attaching'|'attached'|'detaching'|'detached'
  }
}
```

Response Structure

- (*dict*) –
 - **VpcAttachment** (*dict*) –
Information about the attachment.
 - * **VpcId** (*string*) –
The ID of the VPC.

* **State** (*string*) –

The current state of the attachment.

authorize_security_group_egress (***kwargs*)

Adds one or more egress rules to a security group for use with a VPC. Specifically, this action permits instances to send traffic to one or more destination CIDR IP address ranges, or to one or more destination security groups for the same VPC.

Warning: You can have up to 50 rules per security group (covering both ingress and egress rules).

A security group is for use with instances either in the EC2-Classical platform or in a specific VPC. This action doesn't apply to security groups for use in EC2-Classical. For more information, see [Security Groups for Your VPC](#) in the *Amazon Virtual Private Cloud User Guide*.

Each rule consists of the protocol (for example, TCP), plus either a CIDR range or a source group. For the TCP and UDP protocols, you must also specify the destination port or port range. For the ICMP protocol, you must also specify the ICMP type and code. You can use -1 for the type or code to mean all types or all codes.

Rule changes are propagated to affected instances as quickly as possible. However, a small delay might occur.

Request Syntax

```
response = client.authorize_security_group_egress(
    DryRun=True|False,
    GroupId='string',
    SourceSecurityGroupName='string',
    SourceSecurityGroupOwnerId='string',
    IpProtocol='string',
    FromPort=123,
    ToPort=123,
    CidrIp='string',
    IpPermissions=[
        {
            'IpProtocol': 'string',
            'FromPort': 123,
            'ToPort': 123,
            'UserIdGroupPairs': [
                {
                    'UserId': 'string',
                    'GroupName': 'string',
                    'GroupId': 'string'
                },
            ],
            'IpRanges': [
                {
                    'CidrIp': 'string'
                },
            ],
            'PrefixListIds': [
                {
                    'PrefixListId': 'string'
                },
            ],
        },
    ],
)
```

```
]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **GroupId** (*string*) – **[REQUIRED]**
The ID of the security group.
- **SourceSecurityGroupName** (*string*) – The name of a destination security group. To authorize outbound access to a destination security group, we recommend that you use a set of IP permissions instead.
- **SourceSecurityGroupOwnerId** (*string*) – The AWS account number for a destination security group. To authorize outbound access to a destination security group, we recommend that you use a set of IP permissions instead.
- **IpProtocol** (*string*) – The IP protocol name (`tcp`, `udp`, `icmp`) or number (see [Protocol Numbers](#)). Use `-1` to specify all.
- **FromPort** (*integer*) – The start of port range for the TCP and UDP protocols, or an ICMP type number. For the ICMP type number, use `-1` to specify all ICMP types.
- **ToPort** (*integer*) – The end of port range for the TCP and UDP protocols, or an ICMP code number. For the ICMP code number, use `-1` to specify all ICMP codes for the ICMP type.
- **CidrIp** (*string*) – The CIDR IP address range. You can't specify this parameter when specifying a source security group.
- **IpPermissions** (*list*) – A set of IP permissions. You can't specify a destination security group and a CIDR IP address range.
 - (*dict*) –

Describes a security group rule.

- * **IpProtocol** (*string*) –

The protocol.

When you call `DescribeSecurityGroups`, the protocol value returned is the number. Exception: For TCP, UDP, and ICMP, the value returned is the name (for example, `tcp`, `udp`, or `icmp`). For a list of protocol numbers, see [Protocol Numbers](#). (VPC only) When you call `AuthorizeSecurityGroupIngress`, you can use `-1` to specify all.

- * **FromPort** (*integer*) –

The start of port range for the TCP and UDP protocols, or an ICMP type number. A value of `-1` indicates all ICMP types.

- * **ToPort** (*integer*) –

The end of port range for the TCP and UDP protocols, or an ICMP code. A value of `-1` indicates all ICMP codes for the specified ICMP type.

- * **UserIdGroupPairs** (*list*) –

One or more security group and AWS account ID pairs.

- (*dict*) –

Describes a security group and AWS account ID pair.

- **UserId** (*string*) –

The ID of an AWS account. EC2-Classic only.

- **GroupName** (*string*) –

The name of the security group. In a request, use this parameter for a security group in EC2-Classic or a default VPC only. For a security group in a nondefault VPC, use `GroupId`.

- **GroupId** (*string*) –

The ID of the security group.

- * **IpRanges** (*list*) –

One or more IP ranges.

- (*dict*) –

Describes an IP range.

- **CidrIp** (*string*) –

The CIDR range. You can either specify a CIDR range or a source security group, not both.

- * **PrefixListIds** (*list*) –

(Valid for `AuthorizeSecurityGroupEgress`, `RevokeSecurityGroupEgress` and `DescribeSecurityGroups` only) One or more prefix list IDs for an AWS service. In an `AuthorizeSecurityGroupEgress` request, this is the AWS service that you want to access through a VPC endpoint from instances associated with the security group.

- (*dict*) –

The ID of the prefix.

- **PrefixListId** (*string*) –

The ID of the prefix.

Returns None

authorize_security_group_ingress (***kwargs*)

Adds one or more ingress rules to a security group.

Warning: EC2-Classic: You can have up to 100 rules per group.

EC2-VPC: You can have up to 50 rules per group (covering both ingress and egress rules).

Rule changes are propagated to instances within the security group as quickly as possible. However, a small delay might occur.

[EC2-Classic] This action gives one or more CIDR IP address ranges permission to access a security group in your account, or gives one or more security groups (called the *source groups*) permission to access a security group for your account. A source group can be for your own AWS account, or another.

[EC2-VPC] This action gives one or more CIDR IP address ranges permission to access a security group in your VPC, or gives one or more other security groups (called the *source groups*) permission to access a security group for your VPC. The security groups must all be for the same VPC.

Request Syntax

```
response = client.authorize_security_group_ingress(
    DryRun=True|False,
    GroupName='string',
    GroupId='string',
    SourceSecurityGroupName='string',
    SourceSecurityGroupOwnerId='string',
```

```

IpProtocol='string',
FromPort=123,
ToPort=123,
CidrIp='string',
IpPermissions=[
    {
        'IpProtocol': 'string',
        'FromPort': 123,
        'ToPort': 123,
        'UserIdGroupPairs': [
            {
                'UserId': 'string',
                'GroupName': 'string',
                'GroupId': 'string'
            },
        ],
        'IpRanges': [
            {
                'CidrIp': 'string'
            },
        ],
        'PrefixListIds': [
            {
                'PrefixListId': 'string'
            },
        ],
    }
],
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **GroupName** (*string*) – [EC2-Classic, default VPC] The name of the security group.
- **GroupId** (*string*) – The ID of the security group. Required for a nondefault VPC.
- **SourceSecurityGroupName** (*string*) – [EC2-Classic, default VPC] The name of the source security group. You can't specify this parameter in combination with the following parameters: the CIDR IP address range, the start of the port range, the IP protocol, and the end of the port range. For EC2-VPC, the source security group must be in the same VPC.
- **SourceSecurityGroupOwnerId** (*string*) – [EC2-Classic, default VPC] The AWS account number for the source security group. For EC2-VPC, the source security group must be in the same VPC. You can't specify this parameter in combination with the following parameters: the CIDR IP address range, the IP protocol, the start of the port range, and the end of the port range. Creates rules that grant full ICMP, UDP, and TCP access. To create a rule with a specific IP protocol and port range, use a set of IP permissions instead.
- **IpProtocol** (*string*) – The IP protocol name (`tcp`, `udp`, `icmp`) or number (see [Protocol Numbers](#)). (VPC only) Use `-1` to specify all.
- **FromPort** (*integer*) – The start of port range for the TCP and UDP protocols, or an ICMP type number. For the ICMP type number, use `-1` to specify all ICMP types.

- **ToPort** (*integer*) – The end of port range for the TCP and UDP protocols, or an ICMP code number. For the ICMP code number, use `-1` to specify all ICMP codes for the ICMP type.
- **CidrIp** (*string*) – The CIDR IP address range. You can't specify this parameter when specifying a source security group.
- **IpPermissions** (*list*) – A set of IP permissions. Can be used to specify multiple rules in a single command.

– (*dict*) –

Describes a security group rule.

* **IpProtocol** (*string*) –

The protocol.

When you call `DescribeSecurityGroups`, the protocol value returned is the number. Exception: For TCP, UDP, and ICMP, the value returned is the name (for example, `tcp`, `udp`, or `icmp`). For a list of protocol numbers, see [Protocol Numbers](#). (VPC only) When you call `AuthorizeSecurityGroupIngress`, you can use `-1` to specify all.

* **FromPort** (*integer*) –

The start of port range for the TCP and UDP protocols, or an ICMP type number. A value of `-1` indicates all ICMP types.

* **ToPort** (*integer*) –

The end of port range for the TCP and UDP protocols, or an ICMP code. A value of `-1` indicates all ICMP codes for the specified ICMP type.

* **UserIdGroupPairs** (*list*) –

One or more security group and AWS account ID pairs.

· (*dict*) –

Describes a security group and AWS account ID pair.

· **UserId** (*string*) –

The ID of an AWS account. EC2-Classic only.

· **GroupName** (*string*) –

The name of the security group. In a request, use this parameter for a security group in EC2-Classic or a default VPC only. For a security group in a nondefault VPC, use `GroupId`.

· **GroupId** (*string*) –

The ID of the security group.

* **IpRanges** (*list*) –

One or more IP ranges.

· (*dict*) –

Describes an IP range.

· **CidrIp** (*string*) –

The CIDR range. You can either specify a CIDR range or a source security group, not both.

* **PrefixListIds** (*list*) –

(Valid for `AuthorizeSecurityGroupEgress`, `RevokeSecurityGroupEgress` and `DescribeSecurityGroups` only) One or more prefix list IDs for an AWS service. In an `AuthorizeSecurityGroupEgress` request, this is the AWS service that you want to

access through a VPC endpoint from instances associated with the security group.

- *(dict)* –

The ID of the prefix.

- **PrefixListId** (*string*) –

The ID of the prefix.

Returns None

bundle_instance (***kwargs*)

Bundles an Amazon instance store-backed Windows instance.

During bundling, only the root device volume (C:) is bundled. Data on other instance store volumes is not preserved.

Note: This action is not applicable for Linux/Unix instances or Windows instances that are backed by Amazon EBS.

For more information, see [Creating an Instance Store-Backed Windows AMI](#).

Request Syntax

```
response = client.bundle_instance(
    DryRun=True|False,
    InstanceId='string',
    Storage={
        'S3': {
            'Bucket': 'string',
            'Prefix': 'string',
            'AWSAccessKeyId': 'string',
            'UploadPolicy': b'bytes',
            'UploadPolicySignature': 'string'
        }
    }
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceId** (*string*) – **[REQUIRED]**

The ID of the instance to bundle.

Type: String

Default: None

Required: Yes

- **Storage** (*dict*) – **[REQUIRED]**

The bucket in which to store the AMI. You can specify a bucket that you already own or a new bucket that Amazon EC2 creates on your behalf. If you specify a bucket that belongs to someone else, Amazon EC2 returns an error.

- **S3** (*dict*) –

An Amazon S3 storage location.

- * **Bucket** (*string*) –

The bucket in which to store the AMI. You can specify a bucket that you already own or a new bucket that Amazon EC2 creates on your behalf. If you specify a bucket that belongs to someone else, Amazon EC2 returns an error.

- * **Prefix** (*string*) –

The beginning of the file name of the AMI.

- * **AWSAccessKeyId** (*string*) –

The access key ID of the owner of the bucket. Before you specify a value for your access key ID, review and follow the guidance in [Best Practices for Managing AWS Access Keys](#).

- * **UploadPolicy** (*bytes*) –

A Base64-encoded Amazon S3 upload policy that gives Amazon EC2 permission to upload items into Amazon S3 on your behalf.

- * **UploadPolicySignature** (*string*) –

The signature of the Base64 encoded JSON document.

Return type dict

Returns

Response Syntax

```
{
  'BundleTask': {
    'InstanceId': 'string',
    'BundleId': 'string',
    'State': 'pending'|'waiting-for-shutdown'|'bundling'|'storing'|'cancelling',
    'StartTime': datetime(2015, 1, 1),
    'UpdateTime': datetime(2015, 1, 1),
    'Storage': {
      'S3': {
        'Bucket': 'string',
        'Prefix': 'string',
        'AWSAccessKeyId': 'string',
        'UploadPolicy': b'bytes',
        'UploadPolicySignature': 'string'
      }
    },
    'Progress': 'string',
    'BundleTaskError': {
      'Code': 'string',
      'Message': 'string'
    }
  }
}
```

Response Structure

- (*dict*) –

- **BundleTask** (*dict*) –

Information about the bundle task.

- * **InstanceId** (*string*) –

The ID of the instance associated with this bundle task.

- * **BundleId** (*string*) –

The ID of the bundle task.

* **State** (*string*) –

The state of the task.

* **StartTime** (*datetime*) –

The time this task started.

* **UpdateTime** (*datetime*) –

The time of the most recent update for the task.

* **Storage** (*dict*) –

The Amazon S3 storage locations.

· **S3** (*dict*) –

An Amazon S3 storage location.

· **Bucket** (*string*) –

The bucket in which to store the AMI. You can specify a bucket that you already own or a new bucket that Amazon EC2 creates on your behalf. If you specify a bucket that belongs to someone else, Amazon EC2 returns an error.

· **Prefix** (*string*) –

The beginning of the file name of the AMI.

· **AWSAccessKeyId** (*string*) –

The access key ID of the owner of the bucket. Before you specify a value for your access key ID, review and follow the guidance in [Best Practices for Managing AWS Access Keys](#).

· **UploadPolicy** (*bytes*) –

A Base64-encoded Amazon S3 upload policy that gives Amazon EC2 permission to upload items into Amazon S3 on your behalf.

· **UploadPolicySignature** (*string*) –

The signature of the Base64 encoded JSON document.

* **Progress** (*string*) –

The level of task completion, as a percent (for example, 20%).

* **BundleTaskError** (*dict*) –

If the task fails, a description of the error.

· **Code** (*string*) –

The error code.

· **Message** (*string*) –

The error message.

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

cancel_bundle_task (***kwargs*)

Cancels a bundling operation for an instance store-backed Windows instance.

Request Syntax

```
response = client.cancel_bundle_task(
    DryRun=True|False,
    BundleId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **BundleId** (*string*) – **[REQUIRED]**

The ID of the bundle task.

Return type dict

Returns

Response Syntax

```
{
  'BundleTask': {
    'InstanceId': 'string',
    'BundleId': 'string',
    'State': 'pending'|'waiting-for-shutdown'|'bundling'|'storing'|'cancelling',
    'StartTime': datetime(2015, 1, 1),
    'UpdateTime': datetime(2015, 1, 1),
    'Storage': {
      'S3': {
        'Bucket': 'string',
        'Prefix': 'string',
        'AWSAccessKeyId': 'string',
        'UploadPolicy': b'bytes',
        'UploadPolicySignature': 'string'
      }
    },
    'Progress': 'string',
    'BundleTaskError': {
      'Code': 'string',
      'Message': 'string'
    }
  }
}
```

Response Structure

- (*dict*) –
 - **BundleTask** (*dict*) –

Information about the bundle task.

- * **InstanceId** (*string*) –

The ID of the instance associated with this bundle task.

- * **BundleId** (*string*) –

The ID of the bundle task.

- * **State** (*string*) –

The state of the task.

* **StartTime** (*datetime*) –

The time this task started.

* **UpdateTime** (*datetime*) –

The time of the most recent update for the task.

* **Storage** (*dict*) –

The Amazon S3 storage locations.

· **S3** (*dict*) –

An Amazon S3 storage location.

· **Bucket** (*string*) –

The bucket in which to store the AMI. You can specify a bucket that you already own or a new bucket that Amazon EC2 creates on your behalf. If you specify a bucket that belongs to someone else, Amazon EC2 returns an error.

· **Prefix** (*string*) –

The beginning of the file name of the AMI.

· **AWSAccessKeyId** (*string*) –

The access key ID of the owner of the bucket. Before you specify a value for your access key ID, review and follow the guidance in [Best Practices for Managing AWS Access Keys](#).

· **UploadPolicy** (*bytes*) –

A Base64-encoded Amazon S3 upload policy that gives Amazon EC2 permission to upload items into Amazon S3 on your behalf.

· **UploadPolicySignature** (*string*) –

The signature of the Base64 encoded JSON document.

* **Progress** (*string*) –

The level of task completion, as a percent (for example, 20%).

* **BundleTaskError** (*dict*) –

If the task fails, a description of the error.

· **Code** (*string*) –

The error code.

· **Message** (*string*) –

The error message.

cancel_conversion_task (***kwargs*)

Cancels an active conversion task. The task can be the import of an instance or volume. The action removes all artifacts of the conversion, including a partially uploaded volume or instance. If the conversion is complete or is in the process of transferring the final disk image, the command fails and returns an exception.

For more information, see [Using the Command Line Tools to Import Your Virtual Machine to Amazon EC2](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.cancel_conversion_task(
    DryRun=True|False,
    ConversionTaskId='string',
```

```
ReasonMessage='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **ConversionTaskId** (*string*) – [REQUIRED]
The ID of the conversion task.
- **ReasonMessage** (*string*) – The reason for canceling the conversion task.

Returns None

cancel_export_task (***kwargs*)

Cancels an active export task. The request removes all artifacts of the export, including any partially-created Amazon S3 objects. If the export task is complete or is in the process of transferring the final disk image, the command fails and returns an error.

Request Syntax

```
response = client.cancel_export_task(
    ExportTaskId='string'
)
```

Parameters **ExportTaskId** (*string*) – [REQUIRED]

The ID of the export task. This is the ID returned by `CreateInstanceExportTask`.

Returns None

cancel_import_task (***kwargs*)

Cancels an in-process import virtual machine or import snapshot task.

Request Syntax

```
response = client.cancel_import_task(
    DryRun=True|False,
    ImportTaskId='string',
    CancelReason='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **ImportTaskId** (*string*) – The ID of the import image or import snapshot task to be canceled.
- **CancelReason** (*string*) – The reason for canceling the task.

Return type dict

Returns

Response Syntax

```
{
    'ImportTaskId': 'string',
    'State': 'string',
}
```

```
    'PreviousState': 'string'
}
```

Response Structure

- (dict) –
 - **ImportTaskId** (string) –
The ID of the task being canceled.
 - **State** (string) –
The current state of the task being canceled.
 - **PreviousState** (string) –
The current state of the task being canceled.

cancel_reserved_instances_listing (**kwargs)

Cancels the specified Reserved Instance listing in the Reserved Instance Marketplace.

For more information, see [Reserved Instance Marketplace](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.cancel_reserved_instances_listing(
    ReservedInstancesListingId='string'
)
```

Parameters **ReservedInstancesListingId** (string) – [REQUIRED]

The ID of the Reserved Instance listing.

Return type dict

Returns

Response Syntax

```
{
  'ReservedInstancesListings': [
    {
      'ReservedInstancesListingId': 'string',
      'ReservedInstancesId': 'string',
      'CreateDate': datetime(2015, 1, 1),
      'UpdateDate': datetime(2015, 1, 1),
      'Status': 'active'|'pending'|'cancelled'|'closed',
      'StatusMessage': 'string',
      'InstanceCounts': [
        {
          'State': 'available'|'sold'|'cancelled'|'pending',
          'InstanceCount': 123
        },
      ],
      'PriceSchedules': [
        {
          'Term': 123,
          'Price': 123.0,
          'CurrencyCode': 'USD',
          'Active': True|False
        },
      ],
      'Tags': [
        {
```

```
        'Key': 'string',
        'Value': 'string'
    },
    ],
    'ClientToken': 'string'
},
]
```

Response Structure

- (*dict*) –

- **ReservedInstancesListings** (*list*) –

The Reserved Instance listing.

- * (*dict*) –

Describes a Reserved Instance listing.

- **ReservedInstancesListingId** (*string*) –

The ID of the Reserved Instance listing.

- **ReservedInstancesId** (*string*) –

The ID of the Reserved Instance.

- **CreateDate** (*datetime*) –

The time the listing was created.

- **UpdateDate** (*datetime*) –

The last modified timestamp of the listing.

- **Status** (*string*) –

The status of the Reserved Instance listing.

- **StatusMessage** (*string*) –

The reason for the current status of the Reserved Instance listing. The response can be blank.

- **InstanceCounts** (*list*) –

The number of instances in this state.

- (*dict*) –

Describes a Reserved Instance listing state.

- **State** (*string*) –

The states of the listed Reserved Instances.

- **InstanceCount** (*integer*) –

The number of listed Reserved Instances in the state specified by the `state`.

- **PriceSchedules** (*list*) –

The price of the Reserved Instance listing.

- (*dict*) –

Describes the price for a Reserved Instance.

- **Term** (*integer*) –

The number of months remaining in the reservation. For example, 2 is the second to the last month before the capacity reservation expires.

- **Price** (*float*) –

The fixed price for the term.

- **CurrencyCode** (*string*) –

The currency for transacting the Reserved Instance resale. At this time, the only supported currency is USD .

- **Active** (*boolean*) –

The current price schedule, as determined by the term remaining for the Reserved Instance in the listing.

A specific price schedule is always in effect, but only one price schedule can be active at any time. Take, for example, a Reserved Instance listing that has five months remaining in its term. When you specify price schedules for five months and two months, this means that schedule 1, covering the first three months of the remaining term, will be active during months 5, 4, and 3. Then schedule 2, covering the last two months of the term, will be active for months 2 and 1.

- **Tags** (*list*) –

Any tags assigned to the resource.

- (*dict*) –

Describes a tag.

- **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

- **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

- **ClientToken** (*string*) –

A unique, case-sensitive key supplied by the client to ensure that the request is idempotent. For more information, see [Ensuring Idempotency](#) .

cancel_spot_fleet_requests (***kwargs*)

Cancels the specified Spot fleet requests.

Request Syntax

```
response = client.cancel_spot_fleet_requests(
    DryRun=True|False,
    SpotFleetRequestIds=[
        'string',
    ],
    TerminateInstances=True|False
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation` . Otherwise, it is `UnauthorizedOperation` .

- **SpotFleetRequestIds** (*list*) – [REQUIRED]

The IDs of the Spot fleet requests.

– (*string*) –

- **TerminateInstances** (*boolean*) – [REQUIRED]

Indicates whether to terminate instances for a Spot fleet request if it is canceled successfully.

Return type dict

Returns

Response Syntax

```
{
  'UnsuccessfulFleetRequests': [
    {
      'SpotFleetRequestId': 'string',
      'Error': {
        'Code': 'fleetRequestIdDoesNotExist'|'fleetRequestIdMalformed'|'fail
        'Message': 'string'
      }
    },
  ],
  'SuccessfulFleetRequests': [
    {
      'SpotFleetRequestId': 'string',
      'CurrentSpotFleetRequestState': 'submitted'|'active'|'cancelled'|'fail
      'PreviousSpotFleetRequestState': 'submitted'|'active'|'cancelled'|'fail
    },
  ],
}
```

Response Structure

- (*dict*) –

Contains the output of CancelSpotFleetRequests.

- **UnsuccessfulFleetRequests** (*list*) –

Information about the Spot fleet requests that are not successfully canceled.

* (*dict*) –

Describes a Spot fleet request that was not successfully canceled.

- **SpotFleetRequestId** (*string*) –

The ID of the Spot fleet request.

- **Error** (*dict*) –

The error.

- **Code** (*string*) –

The error code.

- **Message** (*string*) –

The description for the error code.

- **SuccessfulFleetRequests** (*list*) –

Information about the Spot fleet requests that are successfully canceled.

* (*dict*) –

Describes a Spot fleet request that was successfully canceled.

- **SpotFleetRequestId** (*string*) –
The ID of the Spot fleet request.
- **CurrentSpotFleetRequestState** (*string*) –
The current state of the Spot fleet request.
- **PreviousSpotFleetRequestState** (*string*) –
The previous state of the Spot fleet request.

cancel_spot_instance_requests (***kwargs*)

Cancels one or more Spot instance requests. Spot instances are instances that Amazon EC2 starts on your behalf when the bid price that you specify exceeds the current Spot price. Amazon EC2 periodically sets the Spot price based on available Spot instance capacity and current Spot instance requests. For more information, see [Spot Instance Requests](#) in the *Amazon Elastic Compute Cloud User Guide*.

Warning: Canceling a Spot instance request does not terminate running Spot instances associated with the request.

Request Syntax

```
response = client.cancel_spot_instance_requests(
    DryRun=True|False,
    SpotInstanceRequestIds=[
        'string',
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SpotInstanceRequestIds** (*list*) – **[REQUIRED]**
One or more Spot instance request IDs.
– (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'CancelledSpotInstanceRequests': [
        {
            'SpotInstanceRequestId': 'string',
            'State': 'active'|'open'|'closed'|'cancelled'|'completed'
        },
    ]
}
```

Response Structure

- (*dict*) –
Contains the output of `CancelSpotInstanceRequests`.
– **CancelledSpotInstanceRequests** (*list*) –
One or more Spot instance requests.

* (*dict*) –

Describes a request to cancel a Spot instance.

- **SpotInstanceRequestId** (*string*) –

The ID of the Spot instance request.

- **State** (*string*) –

The state of the Spot instance request.

confirm_product_instance (***kwargs*)

Determines whether a product code is associated with an instance. This action can only be used by the owner of the product code. It is useful when a product code owner needs to verify whether another user's instance is eligible for support.

Request Syntax

```
response = client.confirm_product_instance(
    DryRun=True|False,
    ProductCode='string',
    InstanceId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **ProductCode** (*string*) – [REQUIRED]
The product code. This must be a product code that you own.
- **InstanceId** (*string*) – [REQUIRED]
The ID of the instance.

Return type dict

Returns

Response Syntax

```
{
    'OwnerId': 'string',
    'Return': True|False
}
```

Response Structure

- (*dict*) –
 - **OwnerId** (*string*) –
The AWS account ID of the instance owner. This is only present if the product code is attached to the instance.
 - **Return** (*boolean*) –
The return value of the request. Returns `true` if the specified product code is owned by the requester and associated with the specified instance.

copy_image (***kwargs*)

Initiates the copy of an AMI from the specified source region to the current region. You specify the destination region by using its endpoint when making the request. AMIs that use encrypted EBS snapshots cannot be copied with this method.

For more information, see [Copying AMIs](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.copy_image(
    DryRun=True|False,
    SourceRegion='string',
    SourceImageId='string',
    Name='string',
    Description='string',
    ClientToken='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SourceRegion** (*string*) – **[REQUIRED]**
The name of the region that contains the AMI to copy.
- **SourceImageId** (*string*) – **[REQUIRED]**
The ID of the AMI to copy.
- **Name** (*string*) – **[REQUIRED]**
The name of the new AMI in the destination region.
- **Description** (*string*) – A description for the new AMI in the destination region.
- **ClientToken** (*string*) – Unique, case-sensitive identifier you provide to ensure idempotency of the request. For more information, see [How to Ensure Idempotency](#) in the *Amazon Elastic Compute Cloud User Guide*.

Return type dict

Returns

Response Syntax

```
{
    'ImageId': 'string'
}
```

Response Structure

- (*dict*) –
 - **ImageId** (*string*) –
The ID of the new AMI.

copy_snapshot (***kwargs*)

Copies a point-in-time snapshot of an EBS volume and stores it in Amazon S3. You can copy the snapshot within the same region or from one region to another. You can use the snapshot to create EBS volumes or Amazon Machine Images (AMIs). The snapshot is copied to the regional endpoint that you send the HTTP request to.

Copies of encrypted EBS snapshots remain encrypted. Copies of unencrypted snapshots remain unencrypted, unless the `Encrypted` flag is specified during the snapshot copy operation. By default, encrypted snapshot copies use the default AWS Key Management Service (AWS KMS) customer master key (CMK); however, you can specify a non-default CMK with the `KmsKeyId` parameter.

For more information, see [Copying an Amazon EBS Snapshot](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.copy_snapshot(  
    DryRun=True|False,  
    SourceRegion='string',  
    SourceSnapshotId='string',  
    Description='string',  
    Encrypted=True|False,  
    KmsKeyId='string'  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SourceRegion** (*string*) – **[REQUIRED]**
The ID of the region that contains the snapshot to be copied.
- **SourceSnapshotId** (*string*) – **[REQUIRED]**
The ID of the EBS snapshot to copy.
- **Description** (*string*) – A description for the EBS snapshot.
- **DestinationRegion** (*string*) – The destination region to use in the `PresignedUrl` parameter of a snapshot copy operation. This parameter is only valid for specifying the destination region in a `PresignedUrl` parameter, where it is required.

Note: `CopySnapshot` sends the snapshot copy to the regional endpoint that you send the HTTP request to, such as `ec2.us-east-1.amazonaws.com` (in the AWS CLI, this is specified with the `--region` parameter or the default region in your AWS configuration file).

Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required

- **PresignedUrl** (*string*) – The pre-signed URL that facilitates copying an encrypted snapshot. This parameter is only required when copying an encrypted snapshot with the Amazon EC2 Query API; it is available as an optional parameter in all other cases. The `PresignedUrl` should use the snapshot source endpoint, the `CopySnapshot` action, and include the `SourceRegion`, `SourceSnapshotId`, and `DestinationRegion` parameters. The `PresignedUrl` must be signed using AWS Signature Version 4. Because EBS snapshots are stored in Amazon S3, the signing algorithm for this parameter uses the same logic that is described in [Authenticating Requests by Using Query Parameters \(AWS Signature Version 4\)](#) in the *Amazon Simple Storage Service API Reference*. An invalid or improperly signed `PresignedUrl` will cause the copy operation to fail asynchronously, and the snapshot will move to an error state.

Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required

- **Encrypted** (*boolean*) – Specifies whether the destination snapshot should be encrypted. There is no way to create an unencrypted snapshot copy from an encrypted snapshot; however, you can encrypt a copy of an unencrypted snapshot with this flag. The default CMK for EBS is used unless a non-default AWS Key Management Service (AWS KMS) CMK is specified with `KmsKeyId`. For more information, see [Amazon EBS Encryption](#) in the *Amazon Elastic Compute Cloud User Guide*.
- **KmsKeyId** (*string*) – The full ARN of the AWS Key Management Service

(AWS KMS) CMK to use when creating the snapshot copy. This parameter is only required if you want to use a non-default CMK; if this parameter is not specified, the default CMK for EBS is used. The ARN contains the `arn:aws:kms` namespace, followed by the region of the CMK, the AWS account ID of the CMK owner, the `key` namespace, and then the CMK ID. For example, `arn:aws:kms:us-east-1:012345678910:key/abcd1234-a123-456a-a12b-a123b4cd56ef`. The specified CMK must exist in the region that the snapshot is being copied to. If a `KmsKeyId` is specified, the `Encrypted` flag must also be set.

Return type dict

Returns

Response Syntax

```
{
    'SnapshotId': 'string'
}
```

Response Structure

- (dict) –
 - **SnapshotId** (string) –
The ID of the new snapshot.

create_customer_gateway (**kwargs)

Provides information to AWS about your VPN customer gateway device. The customer gateway is the appliance at your end of the VPN connection. (The device on the AWS side of the VPN connection is the virtual private gateway.) You must provide the Internet-routable IP address of the customer gateway's external interface. The IP address must be static and can't be behind a device performing network address translation (NAT).

For devices that use Border Gateway Protocol (BGP), you can also provide the device's BGP Autonomous System Number (ASN). You can use an existing ASN assigned to your network. If you don't have an ASN already, you can use a private ASN (in the 64512 - 65534 range).

Note: Amazon EC2 supports all 2-byte ASN numbers in the range of 1 - 65534, with the exception of 7224, which is reserved in the `us-east-1` region, and 9059, which is reserved in the `eu-west-1` region.

For more information about VPN customer gateways, see [Adding a Hardware Virtual Private Gateway to Your VPC](#) in the *Amazon Virtual Private Cloud User Guide*.

Warning: You cannot create more than one customer gateway with the same VPN type, IP address, and BGP ASN parameter values. If you run an identical request more than one time, the first request creates the customer gateway, and subsequent requests return information about the existing customer gateway. The subsequent requests do not create new customer gateway resources.

Request Syntax

```
response = client.create_customer_gateway(
    DryRun=True|False,
    Type='ipsec.1',
    PublicIp='string',
    BgpAsn=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Type** (*string*) – **[REQUIRED]**
The type of VPN connection that this customer gateway supports (`ipsec.1`).
- **PublicIp** (*string*) – **[REQUIRED]**
The Internet-routable IP address for the customer gateway's outside interface. The address must be static.
- **BgpAsn** (*integer*) – **[REQUIRED]**
For devices that support BGP, the customer gateway's BGP ASN.
Default: 65000

Return type dict

Returns

Response Syntax

```
{
  'CustomerGateway': {
    'CustomerGatewayId': 'string',
    'State': 'string',
    'Type': 'string',
    'IpAddress': 'string',
    'BgpAsn': 'string',
    'Tags': [
      {
        'Key': 'string',
        'Value': 'string'
      },
    ]
  }
}
```

Response Structure

- (*dict*) –
 - **CustomerGateway** (*dict*) –
Information about the customer gateway.
 - * **CustomerGatewayId** (*string*) –
The ID of the customer gateway.
 - * **State** (*string*) –
The current state of the customer gateway (`pending` | `available` | `deleting` | `deleted`).
 - * **Type** (*string*) –
The type of VPN connection the customer gateway supports (`ipsec.1`).
 - * **IpAddress** (*string*) –
The Internet-routable IP address of the customer gateway's outside interface.
 - * **BgpAsn** (*string*) –
The customer gateway's Border Gateway Protocol (BGP) Autonomous System Number (ASN).

* **Tags** (*list*) –

Any tags assigned to the customer gateway.

· (*dict*) –

Describes a tag.

· **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

· **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

create_dhcp_options (***kwargs*)

Creates a set of DHCP options for your VPC. After creating the set, you must associate it with the VPC, causing all existing and new instances that you launch in the VPC to use this set of DHCP options. The following are the individual DHCP options you can specify. For more information about the options, see [RFC 2132](#).

- `domain-name-servers` - The IP addresses of up to four domain name servers, or `AmazonProvidedDNS`. The default DHCP option set specifies `AmazonProvidedDNS`. If specifying more than one domain name server, specify the IP addresses in a single parameter, separated by commas.
- `domain-name` - If you're using `AmazonProvidedDNS` in `us-east-1`, specify `ec2.internal`. If you're using `AmazonProvidedDNS` in another region, specify `region.compute.internal` (for example, `ap-northeast-1.compute.internal`). Otherwise, specify a domain name (for example, `MyCompany.com`). **Important**: Some Linux operating systems accept multiple domain names separated by spaces. However, Windows and other Linux operating systems treat the value as a single domain, which results in unexpected behavior. If your DHCP options set is associated with a VPC that has instances with multiple operating systems, specify only one domain name.
- `ntp-servers` - The IP addresses of up to four Network Time Protocol (NTP) servers.
- `netbios-name-servers` - The IP addresses of up to four NetBIOS name servers.
- `netbios-node-type` - The NetBIOS node type (1, 2, 4, or 8). We recommend that you specify 2 (broadcast and multicast are not currently supported). For more information about these node types, see [RFC 2132](#).

Your VPC automatically starts out with a set of DHCP options that includes only a DNS server that we provide (`AmazonProvidedDNS`). If you create a set of options, and if your VPC has an Internet gateway, make sure to set the `domain-name-servers` option either to `AmazonProvidedDNS` or to a domain name server of your choice. For more information about DHCP options, see [DHCP Options Sets](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.create_dhcp_options(
    DryRun=True|False,
    DhcpConfigurations=[
        {
            'Key': 'string',
            'Values': [
                'string',
            ]
        },
    ],
)
```

```
]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **DhcpConfigurations** (*list*) – **[REQUIRED]**

A DHCP configuration option.

- (*dict*) –
 - * **Key** (*string*) –
 - * **Values** (*list*) –
 - (*string*) –

Return type dict

Returns**Response Syntax**

```
{
  'DhcpOptions': {
    'DhcpOptionsId': 'string',
    'DhcpConfigurations': [
      {
        'Key': 'string',
        'Values': [
          {
            'Value': 'string'
          }
        ],
      }
    ],
  },
  'Tags': [
    {
      'Key': 'string',
      'Value': 'string'
    }
  ]
}
```

Response Structure

- (*dict*) –
 - **DhcpOptions** (*dict*) –

A set of DHCP options.

 - * **DhcpOptionsId** (*string*) –

The ID of the set of DHCP options.
 - * **DhcpConfigurations** (*list*) –

One or more DHCP options in the set.

 - (*dict*) –

Describes a DHCP configuration option.

 - **Key** (*string*) –

The name of a DHCP option.

- **Values** (*list*) –
One or more values for the DHCP option.
- (*dict*) –
The value to use for a resource attribute.
- **Value** (*string*) –
Valid values are case-sensitive and vary by action.
- * **Tags** (*list*) –
Any tags assigned to the DHCP options set.
- (*dict*) –
Describes a tag.
- **Key** (*string*) –
The key of the tag.
- Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`
- **Value** (*string*) –
The value of the tag.
- Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

create_flow_logs (***kwargs*)

Creates one or more flow logs to capture IP traffic for a specific network interface, subnet, or VPC. Flow logs are delivered to a specified log group in Amazon CloudWatch Logs. If you specify a VPC or subnet in the request, a log stream is created in CloudWatch Logs for each network interface in the subnet or VPC. Log streams can include information about accepted and rejected traffic to a network interface. You can view the data in your log streams using Amazon CloudWatch Logs.

In your request, you must also specify an IAM role that has permission to publish logs to CloudWatch Logs.

Request Syntax

```
response = client.create_flow_logs(
    ResourceIds=[
        'string',
    ],
    ResourceType='VPC'|'Subnet'|'NetworkInterface',
    TrafficType='ACCEPT'|'REJECT'|'ALL',
    LogGroupName='string',
    DeliverLogsPermissionArn='string',
    ClientToken='string'
)
```

Parameters

- **ResourceIds** (*list*) – **[REQUIRED]**
One or more subnet, network interface, or VPC IDs.
– (*string*) –
- **ResourceType** (*string*) – **[REQUIRED]**
The type of resource on which to create the flow log.
- **TrafficType** (*string*) – **[REQUIRED]**
The type of traffic to log.

- **LogGroupName** (*string*) – [REQUIRED]
The name of the CloudWatch log group.
- **DeliverLogsPermissionArn** (*string*) – [REQUIRED]
The ARN for the IAM role that’s used to post flow logs to a CloudWatch Logs log group.
- **ClientToken** (*string*) – Unique, case-sensitive identifier you provide to ensure the idempotency of the request. For more information, see [How to Ensure Idempotency](#).

Return type dict

Returns

Response Syntax

```
{
  'FlowLogIds': [
    'string',
  ],
  'ClientToken': 'string',
  'Unsuccessful': [
    {
      'ResourceId': 'string',
      'Error': {
        'Code': 'string',
        'Message': 'string'
      }
    }
  ]
}
```

Response Structure

- (*dict*) –
 - **FlowLogIds** (*list*) –
The IDs of the flow logs.
* (*string*) –
 - **ClientToken** (*string*) –
Unique, case-sensitive identifier you provide to ensure the idempotency of the request.
 - **Unsuccessful** (*list*) –
Information about the flow logs that could not be created successfully.
* (*dict*) –
Information about items that were not successfully processed in a batch call.
 - **ResourceId** (*string*) –
The ID of the resource.
 - **Error** (*dict*) –
Information about the error.
 - **Code** (*string*) –
The error code.
 - **Message** (*string*) –
The error message accompanying the error code.

create_image (**kwargs)

Creates an Amazon EBS-backed AMI from an Amazon EBS-backed instance that is either running or stopped.

If you customized your instance with instance store volumes or EBS volumes in addition to the root device volume, the new AMI contains block device mapping information for those volumes. When you launch an instance from this new AMI, the instance automatically launches with those additional volumes.

For more information, see [Creating Amazon EBS-Backed Linux AMIs](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.create_image(
    DryRun=True|False,
    InstanceId='string',
    Name='string',
    Description='string',
    NoReboot=True|False,
    BlockDeviceMappings=[
        {
            'VirtualName': 'string',
            'DeviceName': 'string',
            'Ebs': {
                'SnapshotId': 'string',
                'VolumeSize': 123,
                'DeleteOnTermination': True|False,
                'VolumeType': 'standard'|'io1'|'gp2',
                'Iops': 123,
                'Encrypted': True|False
            },
            'NoDevice': 'string'
        },
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceId** (*string*) – **[REQUIRED]**
The ID of the instance.
- **Name** (*string*) – **[REQUIRED]**
A name for the new image.

Constraints: 3-128 alphanumeric characters, parentheses (()), square brackets ([]), spaces (), periods (.), slashes (/), dashes (-), single quotes ('), at-signs (@), or underscores(_)
- **Description** (*string*) – A description for the new image.
- **NoReboot** (*boolean*) – By default, this parameter is set to `false`, which means Amazon EC2 attempts to shut down the instance cleanly before image creation and then reboots the instance. When the parameter is set to `true`, Amazon EC2 doesn't shut down the instance before creating the image. When this option is used, file system integrity on the created image can't be guaranteed.
- **BlockDeviceMappings** (*list*) – Information about one or more block device mappings.

– (*dict*) –

Describes a block device mapping.

* **VirtualName** (*string*) –

The virtual device name (ephemeral N). Instance store volumes are numbered starting from 0. An instance type with 2 available instance store volumes can specify mappings for `ephemeral0` and `ephemeral1`. The number of available instance store volumes depends on the instance type. After you connect to the instance, you must mount the volume.

Constraints: For M3 instances, you must specify instance store volumes in the block device mapping for the instance. When you launch an M3 instance, we ignore any instance store volumes specified in the block device mapping for the AMI.

* **DeviceName** (*string*) –

The device name exposed to the instance (for example, `/dev/sdh` or `xvdh`).

* **Ebs** (*dict*) –

Parameters used to automatically set up EBS volumes when the instance is launched.

· **SnapshotId** (*string*) –

The ID of the snapshot.

· **VolumeSize** (*integer*) –

The size of the volume, in GiB.

Constraints: 1–1024 for standard volumes, 1–16384 for `gp2` volumes, and 4–16384 for `io1` volumes. If you specify a snapshot, the volume size must be equal to or larger than the snapshot size.

Default: If you’re creating the volume from a snapshot and don’t specify a volume size, the default is the snapshot size.

· **DeleteOnTermination** (*boolean*) –

Indicates whether the EBS volume is deleted on instance termination.

· **VolumeType** (*string*) –

The volume type. `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, and `standard` for Magnetic volumes.

Default: `standard`

· **Iops** (*integer*) –

The number of I/O operations per second (IOPS) that the volume supports. For Provisioned IOPS (SSD) volumes, this represents the number of IOPS that are provisioned for the volume. For General Purpose (SSD) volumes, this represents the baseline performance of the volume and the rate at which the volume accumulates I/O credits for bursting. For more information on General Purpose (SSD) baseline performance, I/O credits, and bursting, see [Amazon EBS Volume Types](#) in the *Amazon Elastic Compute Cloud User Guide*.

Constraint: Range is 100 to 20000 for Provisioned IOPS (SSD) volumes and 3 to 10000 for General Purpose (SSD) volumes.

Condition: This parameter is required for requests to create `io1` volumes; it is not used in requests to create standard or `gp2` volumes.

- **Encrypted** (*boolean*) –

Indicates whether the EBS volume is encrypted. Encrypted Amazon EBS volumes may only be attached to instances that support Amazon EBS encryption.

- * **NoDevice** (*string*) –

Suppresses the specified device included in the block device mapping of the AMI.

Return type dict

Returns

Response Syntax

```
{
    'ImageId': 'string'
}
```

Response Structure

- (*dict*) –
 - **ImageId** (*string*) –

The ID of the new AMI.

create_instance_export_task (**kwargs)

Exports a running or stopped instance to an S3 bucket.

For information about the supported operating systems, image formats, and known limitations for the types of instances you can export, see [Exporting EC2 Instances](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.create_instance_export_task(
    Description='string',
    InstanceId='string',
    TargetEnvironment='citrix'|'vmware'|'microsoft',
    ExportToS3Task={
        'DiskImageFormat': 'VMDK'|'RAW'|'VHD',
        'ContainerFormat': 'ova',
        'S3Bucket': 'string',
        'S3Prefix': 'string'
    }
)
```

Parameters

- **Description** (*string*) – A description for the conversion task or the resource being exported. The maximum length is 255 bytes.
- **InstanceId** (*string*) – **[REQUIRED]**
The ID of the instance.
- **TargetEnvironment** (*string*) – The target virtualization environment.
- **ExportToS3Task** (*dict*) – The format and location for an instance export task.

- **DiskImageFormat** (*string*) –
The format for the exported image.
- **ContainerFormat** (*string*) –
The container format used to combine disk images with metadata (such as OVF). If absent, only the disk image is exported.
- **S3Bucket** (*string*) –
The S3 bucket for the destination image. The destination bucket must exist and grant WRITE and READ_ACP permissions to the AWS account `vm-import-export@amazon.com`.
- **S3Prefix** (*string*) –
The image is written to a single object in the S3 bucket at the S3 key `s3prefix + exportTaskId + '.' + diskImageFormat`.

Return type dict

Returns

Response Syntax

```
{
  'ExportTask': {
    'ExportTaskId': 'string',
    'Description': 'string',
    'State': 'active'|'cancelling'|'cancelled'|'completed',
    'StatusMessage': 'string',
    'InstanceExportDetails': {
      'InstanceId': 'string',
      'TargetEnvironment': 'citrix'|'vmware'|'microsoft'
    },
    'ExportToS3Task': {
      'DiskImageFormat': 'VMDK'|'RAW'|'VHD',
      'ContainerFormat': 'ova',
      'S3Bucket': 'string',
      'S3Key': 'string'
    }
  }
}
```

Response Structure

- (*dict*) –
 - **ExportTask** (*dict*) –
Information about the instance export task.
 - * **ExportTaskId** (*string*) –
The ID of the export task.
 - * **Description** (*string*) –
A description of the resource being exported.
 - * **State** (*string*) –
The state of the export task.
 - * **StatusMessage** (*string*) –
The status message related to the export task.
 - * **InstanceExportDetails** (*dict*) –
Information about the instance to export.

- **InstanceId** (*string*) –
The ID of the resource being exported.
- **TargetEnvironment** (*string*) –
The target virtualization environment.
- * **ExportToS3Task** (*dict*) –
Information about the export task.
 - **DiskImageFormat** (*string*) –
The format for the exported image.
 - **ContainerFormat** (*string*) –
The container format used to combine disk images with meta-data (such as OVF). If absent, only the disk image is exported.
 - **S3Bucket** (*string*) –
The S3 bucket for the destination image. The destination bucket must exist and grant WRITE and READ_ACP permissions to the AWS account `vm-import-export@amazon.com`.
 - **S3Key** (*string*) –
The encryption key for your S3 bucket.

create_internet_gateway (***kwargs*)

Creates an Internet gateway for use with a VPC. After creating the Internet gateway, you attach it to a VPC using `AttachInternetGateway`.

For more information about your VPC and Internet gateway, see the [Amazon Virtual Private Cloud User Guide](#).

Request Syntax

```
response = client.create_internet_gateway(
    DryRun=True|False
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns

Response Syntax

```
{
  'InternetGateway': {
    'InternetGatewayId': 'string',
    'Attachments': [
      {
        'VpcId': 'string',
        'State': 'attaching'|'attached'|'detaching'|'detached'
      },
    ],
    'Tags': [
      {
        'Key': 'string',
```

```
        'Value': 'string'
    },
]
}
```

Response Structure

- *(dict)* –

- **InternetGateway** (*dict*) –

Information about the Internet gateway.

- * **InternetGatewayId** (*string*) –

The ID of the Internet gateway.

- * **Attachments** (*list*) –

Any VPCs attached to the Internet gateway.

- (*dict*) –

Describes the attachment of a VPC to an Internet gateway.

- **VpcId** (*string*) –

The ID of the VPC.

- **State** (*string*) –

The current state of the attachment.

- * **Tags** (*list*) –

Any tags assigned to the Internet gateway.

- (*dict*) –

Describes a tag.

- **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

- **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

create_key_pair (***kwargs*)

Creates a 2048-bit RSA key pair with the specified name. Amazon EC2 stores the public key and displays the private key for you to save to a file. The private key is returned as an unencrypted PEM encoded PKCS#8 private key. If a key with the specified name already exists, Amazon EC2 returns an error.

You can have up to five thousand key pairs per region.

The key pair returned to you is available only in the region in which you create it. To create a key pair that is available in all regions, use `ImportKeyPair`.

For more information about key pairs, see [Key Pairs](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.create_key_pair(
    DryRun=True|False,
    KeyName='string'
)
```


Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **KeyName** (*string*) – **[REQUIRED]**
A unique name for the key pair.
Constraints: Up to 255 ASCII characters

Return type dict**Returns****Response Syntax**

```
{
  'KeyName': 'string',
  'KeyFingerprint': 'string',
  'KeyMaterial': 'string'
}
```

Response Structure

- (*dict*) –
Information about the key pair.
 - **KeyName** (*string*) –
The name of the key pair.
 - **KeyFingerprint** (*string*) –
The SHA-1 digest of the DER encoded private key.
 - **KeyMaterial** (*string*) –
An unencrypted PEM encoded RSA private key.

create_network_acl (***kwargs*)

Creates a network ACL in a VPC. Network ACLs provide an optional layer of security (in addition to security groups) for the instances in your VPC.

For more information about network ACLs, see [Network ACLs](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.create_network_acl(
    DryRun=True|False,
    VpcId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcId** (*string*) – **[REQUIRED]**
The ID of the VPC.

Return type dict**Returns****Response Syntax**

```
{
  'NetworkAcl': {
    'NetworkAclId': 'string',
    'VpcId': 'string',
    'IsDefault': True|False,
    'Entries': [
      {
        'RuleNumber': 123,
        'Protocol': 'string',
        'RuleAction': 'allow'|'deny',
        'Egress': True|False,
        'CidrBlock': 'string',
        'IcmpTypeCode': {
          'Type': 123,
          'Code': 123
        },
        'PortRange': {
          'From': 123,
          'To': 123
        }
      },
    ],
    'Associations': [
      {
        'NetworkAclAssociationId': 'string',
        'NetworkAclId': 'string',
        'SubnetId': 'string'
      },
    ],
    'Tags': [
      {
        'Key': 'string',
        'Value': 'string'
      },
    ],
  }
}
```

Response Structure

- *(dict)* –
 - **NetworkAcl** (*dict*) –

Information about the network ACL.

 - * **NetworkAclId** (*string*) –

The ID of the network ACL.
 - * **VpcId** (*string*) –

The ID of the VPC for the network ACL.
 - * **IsDefault** (*boolean*) –

Indicates whether this is the default network ACL for the VPC.
 - * **Entries** (*list*) –

One or more entries (rules) in the network ACL.

 - *(dict)* –

Describes an entry in a network ACL.
 - **RuleNumber** (*integer*) –

The rule number for the entry. ACL entries are processed in ascending order by rule number.

- **Protocol** (*string*) –

The protocol. A value of -1 means all protocols.

- **RuleAction** (*string*) –

Indicates whether to allow or deny the traffic that matches the rule.

- **Egress** (*boolean*) –

Indicates whether the rule is an egress rule (applied to traffic leaving the subnet).

- **CidrBlock** (*string*) –

The network range to allow or deny, in CIDR notation.

- **IcmpTypeCode** (*dict*) –

ICMP protocol: The ICMP type and code.

- **Type** (*integer*) –

The ICMP code. A value of -1 means all codes for the specified ICMP type.

- **Code** (*integer*) –

The ICMP type. A value of -1 means all types.

- **PortRange** (*dict*) –

TCP or UDP protocols: The range of ports the rule applies to.

- **From** (*integer*) –

The first port in the range.

- **To** (*integer*) –

The last port in the range.

- * **Associations** (*list*) –

Any associations between the network ACL and one or more subnets

- (*dict*) –

Describes an association between a network ACL and a subnet.

- **NetworkAclAssociationId** (*string*) –

The ID of the association between a network ACL and a subnet.

- **NetworkAclId** (*string*) –

The ID of the network ACL.

- **SubnetId** (*string*) –

The ID of the subnet.

- * **Tags** (*list*) –

Any tags assigned to the network ACL.

- (*dict*) –

Describes a tag.

- **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

- **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

create_network_acl_entry (***kwargs*)

Creates an entry (a rule) in a network ACL with the specified rule number. Each network ACL has a set of numbered ingress rules and a separate set of numbered egress rules. When determining whether a packet should be allowed in or out of a subnet associated with the ACL, we process the entries in the ACL according to the rule numbers, in ascending order. Each network ACL has a set of ingress rules and a separate set of egress rules.

We recommend that you leave room between the rule numbers (for example, 100, 110, 120, ...), and not number them one right after the other (for example, 101, 102, 103, ...). This makes it easier to add a rule between existing ones without having to renumber the rules.

After you add an entry, you can't modify it; you must either replace it, or create an entry and delete the old one.

For more information about network ACLs, see [Network ACLs](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.create_network_acl_entry(
    DryRun=True|False,
    NetworkAclId='string',
    RuleNumber=123,
    Protocol='string',
    RuleAction='allow'|'deny',
    Egress=True|False,
    CidrBlock='string',
    IcmpTypeCode={
        'Type': 123,
        'Code': 123
    },
    PortRange={
        'From': 123,
        'To': 123
    }
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **NetworkAclId** (*string*) – **[REQUIRED]**
The ID of the network ACL.
- **RuleNumber** (*integer*) – **[REQUIRED]**
The rule number for the entry (for example, 100). ACL entries are processed in ascending order by rule number.
Constraints: Positive integer from 1 to 32766

- **Protocol** (*string*) – [REQUIRED]
The protocol. A value of -1 means all protocols.
- **RuleAction** (*string*) – [REQUIRED]
Indicates whether to allow or deny the traffic that matches the rule.
- **Egress** (*boolean*) – [REQUIRED]
Indicates whether this is an egress rule (rule is applied to traffic leaving the subnet).
- **CidrBlock** (*string*) – [REQUIRED]
The network range to allow or deny, in CIDR notation (for example 172.16.0.0/24).
- **IcmpTypeCode** (*dict*) – ICMP protocol: The ICMP type and code. Required if specifying ICMP for the protocol.
 - **Type** (*integer*) –
The ICMP code. A value of -1 means all codes for the specified ICMP type.
 - **Code** (*integer*) –
The ICMP type. A value of -1 means all types.
- **PortRange** (*dict*) – TCP or UDP protocols: The range of ports the rule applies to.
 - **From** (*integer*) –
The first port in the range.
 - **To** (*integer*) –
The last port in the range.

Returns None

create_network_interface (***kwargs*)

Creates a network interface in the specified subnet.

For more information about network interfaces, see [Elastic Network Interfaces](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.create_network_interface(
    SubnetId='string',
    Description='string',
    PrivateIpAddress='string',
    Groups=[
        'string',
    ],
    PrivateIpAddresses=[
        {
            'PrivateIpAddress': 'string',
            'Primary': True|False
        },
    ],
    SecondaryPrivateIpAddressCount=123,
    DryRun=True|False
)
```

Parameters

- **SubnetId** (*string*) – [REQUIRED]

The ID of the subnet to associate with the network interface.

- **Description** (*string*) – A description for the network interface.
- **PrivateIpAddress** (*string*) – The primary private IP address of the network interface. If you don't specify an IP address, Amazon EC2 selects one for you from the subnet range. If you specify an IP address, you cannot indicate any IP addresses specified in `privateIpAddresses` as primary (only one IP address can be designated as primary).
- **Groups** (*list*) – The IDs of one or more security groups.
 - (*string*) –
- **PrivateIpAddresses** (*list*) – One or more private IP addresses.
 - (*dict*) –

Describes a secondary private IP address for a network interface.

* **PrivateIpAddress** (*string*) – [REQUIRED]

The private IP addresses.

* **Primary** (*boolean*) –

Indicates whether the private IP address is the primary private IP address. Only one IP address can be designated as primary.

- **SecondaryPrivateIpAddressCount** (*integer*) – The number of secondary private IP addresses to assign to a network interface. When you specify a number of secondary IP addresses, Amazon EC2 selects these IP addresses within the subnet range. You can't specify this option and specify more than one private IP address using `privateIpAddresses`.

The number of IP addresses you can assign to a network interface varies by instance type. For more information, see [Private IP Addresses Per ENI Per Instance Type](#) in the *Amazon Elastic Compute Cloud User Guide*.

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns

Response Syntax

```
{
  'NetworkInterface': {
    'NetworkInterfaceId': 'string',
    'SubnetId': 'string',
    'VpcId': 'string',
    'AvailabilityZone': 'string',
    'Description': 'string',
    'OwnerId': 'string',
    'RequesterId': 'string',
    'RequesterManaged': True|False,
    'Status': 'available'|'attaching'|'in-use'|'detaching',
    'MacAddress': 'string',
    'PrivateIpAddress': 'string',
    'PrivateDnsName': 'string',
    'SourceDestCheck': True|False,
    'Groups': [
      {
        'GroupName': 'string',
        'GroupId': 'string'
      },
    ],
  },
}
```

```

],
'Attachment': {
    'AttachmentId': 'string',
    'InstanceId': 'string',
    'InstanceOwnerId': 'string',
    'DeviceIndex': 123,
    'Status': 'attaching'|'attached'|'detaching'|'detached',
    'AttachTime': datetime(2015, 1, 1),
    'DeleteOnTermination': True|False
},
'Association': {
    'PublicIp': 'string',
    'PublicDnsName': 'string',
    'IpOwnerId': 'string',
    'AllocationId': 'string',
    'AssociationId': 'string'
},
'TagSet': [
    {
        'Key': 'string',
        'Value': 'string'
    },
],
'PrivateIpAddresses': [
    {
        'PrivateIpAddress': 'string',
        'PrivateDnsName': 'string',
        'Primary': True|False,
        'Association': {
            'PublicIp': 'string',
            'PublicDnsName': 'string',
            'IpOwnerId': 'string',
            'AllocationId': 'string',
            'AssociationId': 'string'
        }
    },
]
}

```

Response Structure

- (*dict*) –
 - **NetworkInterface** (*dict*) –

Information about the network interface.

 - * **NetworkInterfaceId** (*string*) –

The ID of the network interface.
 - * **SubnetId** (*string*) –

The ID of the subnet.
 - * **VpcId** (*string*) –

The ID of the VPC.
 - * **AvailabilityZone** (*string*) –

The Availability Zone.
 - * **Description** (*string*) –

A description.

- * **OwnerId** (*string*) –

The AWS account ID of the owner of the network interface.

- * **RequesterId** (*string*) –

The ID of the entity that launched the instance on your behalf (for example, AWS Management Console or Auto Scaling).

- * **RequesterManaged** (*boolean*) –

Indicates whether the network interface is being managed by AWS.

- * **Status** (*string*) –

The status of the network interface.

- * **MacAddress** (*string*) –

The MAC address.

- * **PrivateIpAddress** (*string*) –

The IP address of the network interface within the subnet.

- * **PrivateDnsName** (*string*) –

The private DNS name.

- * **SourceDestCheck** (*boolean*) –

Indicates whether traffic to or from the instance is validated.

- * **Groups** (*list*) –

Any security groups for the network interface.

- (*dict*) –

Describes a security group.

- **GroupName** (*string*) –

The name of the security group.

- **GroupId** (*string*) –

The ID of the security group.

- * **Attachment** (*dict*) –

The network interface attachment.

- **AttachmentId** (*string*) –

The ID of the network interface attachment.

- **InstanceId** (*string*) –

The ID of the instance.

- **InstanceOwnerId** (*string*) –

The AWS account ID of the owner of the instance.

- **DeviceIndex** (*integer*) –

The device index of the network interface attachment on the instance.

- **Status** (*string*) –

The attachment state.

- **AttachTime** (*datetime*) –

The timestamp indicating when the attachment initiated.

- **DeleteOnTermination** (*boolean*) –

Indicates whether the network interface is deleted when the instance is terminated.

* **Association** (*dict*) –

The association information for an Elastic IP associated with the network interface.

· **PublicIp** (*string*) –

The address of the Elastic IP address bound to the network interface.

· **PublicDnsName** (*string*) –

The public DNS name.

· **IpOwnerId** (*string*) –

The ID of the Elastic IP address owner.

· **AllocationId** (*string*) –

The allocation ID.

· **AssociationId** (*string*) –

The association ID.

* **TagSet** (*list*) –

Any tags assigned to the network interface.

· (*dict*) –

Describes a tag.

· **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

· **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

* **PrivateIpAddresses** (*list*) –

The private IP addresses associated with the network interface.

· (*dict*) –

Describes the private IP address of a network interface.

· **PrivateIpAddress** (*string*) –

The private IP address.

· **PrivateDnsName** (*string*) –

The private DNS name.

· **Primary** (*boolean*) –

Indicates whether this IP address is the primary private IP address of the network interface.

· **Association** (*dict*) –

The association information for an Elastic IP address associated with the network interface.

· **PublicIp** (*string*) –

The address of the Elastic IP address bound to the network interface.

- **PublicDnsName** (*string*) –
The public DNS name.
- **IpOwnerId** (*string*) –
The ID of the Elastic IP address owner.
- **AllocationId** (*string*) –
The allocation ID.
- **AssociationId** (*string*) –
The association ID.

create_placement_group (***kwargs*)

Creates a placement group that you launch cluster instances into. You must give the group a name that's unique within the scope of your account.

For more information about placement groups and cluster instances, see [Cluster Instances](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.create_placement_group(  
    DryRun=True|False,  
    GroupName='string',  
    Strategy='cluster'  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **GroupName** (*string*) – **[REQUIRED]**
A name for the placement group.
Constraints: Up to 255 ASCII characters
- **Strategy** (*string*) – **[REQUIRED]**
The placement strategy.

Returns None

create_reserved_instances_listing (***kwargs*)

Creates a listing for Amazon EC2 Reserved Instances to be sold in the Reserved Instance Marketplace. You can submit one Reserved Instance listing at a time. To get a list of your Reserved Instances, you can use the `DescribeReservedInstances` operation.

The Reserved Instance Marketplace matches sellers who want to resell Reserved Instance capacity that they no longer need with buyers who want to purchase additional capacity. Reserved Instances bought and sold through the Reserved Instance Marketplace work like any other Reserved Instances.

To sell your Reserved Instances, you must first register as a seller in the Reserved Instance Marketplace. After completing the registration process, you can create a Reserved Instance Marketplace listing of some or all of your Reserved Instances, and specify the upfront price to receive for them. Your Reserved Instance listings then become available for purchase. To view the details of your Reserved Instance listing, you can use the `DescribeReservedInstancesListings` operation.

For more information, see [Reserved Instance Marketplace](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```

response = client.create_reserved_instances_listing(
    ReservedInstancesId='string',
    InstanceCount=123,
    PriceSchedules=[
        {
            'Term': 123,
            'Price': 123.0,
            'CurrencyCode': 'USD'
        },
    ],
    ClientToken='string'
)

```

Parameters

- **ReservedInstancesId** (*string*) – [REQUIRED]

The ID of the active Reserved Instance.

- **InstanceCount** (*integer*) – [REQUIRED]

The number of instances that are a part of a Reserved Instance account to be listed in the Reserved Instance Marketplace. This number should be less than or equal to the instance count associated with the Reserved Instance ID specified in this call.

- **PriceSchedules** (*list*) – [REQUIRED]

A list specifying the price of the Reserved Instance for each month remaining in the Reserved Instance term.

– (*dict*) –

Describes the price for a Reserved Instance.

- * **Term** (*integer*) –

The number of months remaining in the reservation. For example, 2 is the second to the last month before the capacity reservation expires.

- * **Price** (*float*) –

The fixed price for the term.

- * **CurrencyCode** (*string*) –

The currency for transacting the Reserved Instance resale. At this time, the only supported currency is USD .

- **ClientToken** (*string*) – [REQUIRED]

Unique, case-sensitive identifier you provide to ensure idempotency of your listings. This helps avoid duplicate listings. For more information, see [Ensuring Idempotency](#) .

Return type dict

Returns

Response Syntax

```

{
    'ReservedInstancesListings': [
        {
            'ReservedInstancesListingId': 'string',
            'ReservedInstancesId': 'string',
            'CreateDate': datetime(2015, 1, 1),
            'UpdateDate': datetime(2015, 1, 1),

```

```
'Status': 'active'|'pending'|'cancelled'|'closed',
'StatusMessage': 'string',
'InstanceCounts': [
    {
        'State': 'available'|'sold'|'cancelled'|'pending',
        'InstanceCount': 123
    },
],
'PriceSchedules': [
    {
        'Term': 123,
        'Price': 123.0,
        'CurrencyCode': 'USD',
        'Active': True|False
    },
],
'Tags': [
    {
        'Key': 'string',
        'Value': 'string'
    },
],
'ClientToken': 'string'
},
]
```

Response Structure

- (*dict*) –

- **ReservedInstancesListings** (*list*) –

Information about the Reserved Instances listing.

- * (*dict*) –

Describes a Reserved Instance listing.

- **ReservedInstancesListingId** (*string*) –

The ID of the Reserved Instance listing.

- **ReservedInstancesId** (*string*) –

The ID of the Reserved Instance.

- **CreateDate** (*datetime*) –

The time the listing was created.

- **UpdateDate** (*datetime*) –

The last modified timestamp of the listing.

- **Status** (*string*) –

The status of the Reserved Instance listing.

- **StatusMessage** (*string*) –

The reason for the current status of the Reserved Instance listing. The response can be blank.

- **InstanceCounts** (*list*) –

The number of instances in this state.

- (*dict*) –

Describes a Reserved Instance listing state.

- **State** (*string*) –
The states of the listed Reserved Instances.
- **InstanceCount** (*integer*) –
The number of listed Reserved Instances in the state specified by the `state`.
- **PriceSchedules** (*list*) –
The price of the Reserved Instance listing.
- (*dict*) –
Describes the price for a Reserved Instance.
- **Term** (*integer*) –
The number of months remaining in the reservation. For example, 2 is the second to the last month before the capacity reservation expires.
- **Price** (*float*) –
The fixed price for the term.
- **CurrencyCode** (*string*) –
The currency for transacting the Reserved Instance resale. At this time, the only supported currency is `USD`.
- **Active** (*boolean*) –
The current price schedule, as determined by the term remaining for the Reserved Instance in the listing.

A specific price schedule is always in effect, but only one price schedule can be active at any time. Take, for example, a Reserved Instance listing that has five months remaining in its term. When you specify price schedules for five months and two months, this means that schedule 1, covering the first three months of the remaining term, will be active during months 5, 4, and 3. Then schedule 2, covering the last two months of the term, will be active for months 2 and 1.
- **Tags** (*list*) –
Any tags assigned to the resource.
- (*dict*) –
Describes a tag.
- **Key** (*string*) –
The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`
- **Value** (*string*) –
The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.
- **ClientToken** (*string*) –
A unique, case-sensitive key supplied by the client to ensure that the request is idempotent. For more information, see [Ensuring Idempotency](#).

create_route (**kwargs)

Creates a route in a route table within a VPC.

You must specify one of the following targets: Internet gateway or virtual private gateway, NAT instance, VPC peering connection, or network interface.

When determining how to route traffic, we use the route with the most specific match. For example, let's say the traffic is destined for 192.0.2.3, and the route table includes the following two routes:

- 192.0.2.0/24 (goes to some target A)
- 192.0.2.0/28 (goes to some target B)

Both routes apply to the traffic destined for 192.0.2.3. However, the second route in the list covers a smaller number of IP addresses and is therefore more specific, so we use that route to determine where to target the traffic.

For more information about route tables, see [Route Tables](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.create_route(
    DryRun=True|False,
    RouteTableId='string',
    DestinationCidrBlock='string',
    GatewayId='string',
    InstanceId='string',
    NetworkInterfaceId='string',
    VpcPeeringConnectionId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **RouteTableId** (*string*) – **[REQUIRED]**
The ID of the route table for the route.
- **DestinationCidrBlock** (*string*) – **[REQUIRED]**
The CIDR address block used for the destination match. Routing decisions are based on the most specific match.
- **GatewayId** (*string*) – The ID of an Internet gateway or virtual private gateway attached to your VPC.
- **InstanceId** (*string*) – The ID of a NAT instance in your VPC. The operation fails if you specify an instance ID unless exactly one network interface is attached.
- **NetworkInterfaceId** (*string*) – The ID of a network interface.
- **VpcPeeringConnectionId** (*string*) – The ID of a VPC peering connection.

Return type dict

Returns**Response Syntax**

```
{
    'Return': True|False
}
```

Response Structure

- (*dict*) –

– **Return** (*boolean*) –

Returns `true` if the request succeeds; otherwise, it returns an error.

create_route_table (***kwargs*)

Creates a route table for the specified VPC. After you create a route table, you can add routes and associate the table with a subnet.

For more information about route tables, see [Route Tables](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.create_route_table(
    DryRun=True|False,
    VpcId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcId** (*string*) – **[REQUIRED]**

The ID of the VPC.

Return type dict

Returns

Response Syntax

```
{
  'RouteTable': {
    'RouteTableId': 'string',
    'VpcId': 'string',
    'Routes': [
      {
        'DestinationCidrBlock': 'string',
        'DestinationPrefixListId': 'string',
        'GatewayId': 'string',
        'InstanceId': 'string',
        'InstanceOwnerId': 'string',
        'NetworkInterfaceId': 'string',
        'VpcPeeringConnectionId': 'string',
        'State': 'active'|'blackhole',
        'Origin': 'CreateRouteTable'|'CreateRoute'|'EnableVgwRoutePropagati
      },
    ],
    'Associations': [
      {
        'RouteTableAssociationId': 'string',
        'RouteTableId': 'string',
        'SubnetId': 'string',
        'Main': True|False
      },
    ],
    'Tags': [
      {
        'Key': 'string',
```

```
        'Value': 'string'
    },
],
'PropagatingVgws': [
    {
        'GatewayId': 'string'
    },
]
}
```

Response Structure

- (*dict*) –

- **RouteTable** (*dict*) –

Information about the route table.

- * **RouteTableId** (*string*) –

The ID of the route table.

- * **VpcId** (*string*) –

The ID of the VPC.

- * **Routes** (*list*) –

The routes in the route table.

- (*dict*) –

Describes a route in a route table.

- **DestinationCidrBlock** (*string*) –

The CIDR block used for the destination match.

- **DestinationPrefixListId** (*string*) –

The prefix of the AWS service.

- **GatewayId** (*string*) –

The ID of a gateway attached to your VPC.

- **InstanceId** (*string*) –

The ID of a NAT instance in your VPC.

- **InstanceOwnerId** (*string*) –

The AWS account ID of the owner of the instance.

- **NetworkInterfaceId** (*string*) –

The ID of the network interface.

- **VpcPeeringConnectionId** (*string*) –

The ID of the VPC peering connection.

- **State** (*string*) –

The state of the route. The `blackhole` state indicates that the route's target isn't available (for example, the specified gateway isn't attached to the VPC, or the specified NAT instance has been terminated).

- **Origin** (*string*) –

Describes how the route was created.

- `CreateRouteTable` indicates that route was automatically created when the route table was created.

- `CreateRoute` indicates that the route was manually added to the route table.
- `EnableVgwRoutePropagation` indicates that the route was propagated by route propagation.

* **Associations** (*list*) –

The associations between the route table and one or more subnets.

- (*dict*) –
Describes an association between a route table and a subnet.
- **RouteTableAssociationId** (*string*) –
The ID of the association between a route table and a subnet.
- **RouteTableId** (*string*) –
The ID of the route table.
- **SubnetId** (*string*) –
The ID of the subnet. A subnet ID is not returned for an implicit association.
- **Main** (*boolean*) –
Indicates whether this is the main route table.

* **Tags** (*list*) –

Any tags assigned to the route table.

- (*dict*) –
Describes a tag.
- **Key** (*string*) –
The key of the tag.
Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`
- **Value** (*string*) –
The value of the tag.
Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

* **PropagatingVgws** (*list*) –

Any virtual private gateway (VGW) propagating routes.

- (*dict*) –
Describes a virtual private gateway propagating route.
- **GatewayId** (*string*) –
The ID of the virtual private gateway (VGW).

create_security_group (**kwargs)

Creates a security group.

A security group is for use with instances either in the EC2-Classic platform or in a specific VPC. For more information, see [Amazon EC2 Security Groups](#) in the *Amazon Elastic Compute Cloud User Guide* and [Security Groups for Your VPC](#) in the *Amazon Virtual Private Cloud User Guide*.

Warning: EC2-Classic: You can have up to 500 security groups.
EC2-VPC: You can create up to 100 security groups per VPC.

When you create a security group, you specify a friendly name of your choice. You can have a security group for use in EC2-Classic with the same name as a security group for use in a VPC. However, you can't have two security groups for use in EC2-Classic with the same name or two security groups for use in a VPC with the same name.

You have a default security group for use in EC2-Classic and a default security group for use in your VPC. If you don't specify a security group when you launch an instance, the instance is launched into the appropriate default security group. A default security group includes a default rule that grants instances unrestricted network access to each other.

You can add or remove rules from your security groups using `AuthorizeSecurityGroupIngress`, `AuthorizeSecurityGroupEgress`, `RevokeSecurityGroupIngress`, and `RevokeSecurityGroupEgress`.

Request Syntax

```
response = client.create_security_group(
    DryRun=True|False,
    GroupName='string',
    Description='string',
    VpcId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **GroupName** (*string*) – **[REQUIRED]**
The name of the security group.
Constraints: Up to 255 characters in length
Constraints for EC2-Classic: ASCII characters
Constraints for EC2-VPC: a-z, A-Z, 0-9, spaces, and `._-:/()#@[]+=;{}!$*`
- **Description** (*string*) – **[REQUIRED]**
A description for the security group. This is informational only.
Constraints: Up to 255 characters in length
Constraints for EC2-Classic: ASCII characters
Constraints for EC2-VPC: a-z, A-Z, 0-9, spaces, and `._-:/()#@[]+=;{}!$*`
- **VpcId** (*string*) – **[EC2-VPC]** The ID of the VPC. Required for EC2-VPC.

Return type dict

Returns

Response Syntax

```
{
    'GroupId': 'string'
}
```

Response Structure

- (*dict*) –
 - **GroupId** (*string*) –
The ID of the security group.

create_snapshot (***kwargs*)

Creates a snapshot of an EBS volume and stores it in Amazon S3. You can use snapshots for backups, to make copies of EBS volumes, and to save data before shutting down an instance.

When a snapshot is created, any AWS Marketplace product codes that are associated with the source volume are propagated to the snapshot.

You can take a snapshot of an attached volume that is in use. However, snapshots only capture data that has been written to your EBS volume at the time the snapshot command is issued; this may exclude any data that has been cached by any applications or the operating system. If you can pause any file systems on the volume long enough to take a snapshot, your snapshot should be complete. However, if you cannot pause all file writes to the volume, you should unmount the volume from within the instance, issue the snapshot command, and then remount the volume to ensure a consistent and complete snapshot. You may remount and use your volume while the snapshot status is `pending`.

To create a snapshot for EBS volumes that serve as root devices, you should stop the instance before taking the snapshot.

Snapshots that are taken from encrypted volumes are automatically encrypted. Volumes that are created from encrypted snapshots are also automatically encrypted. Your encrypted volumes and any associated snapshots always remain protected.

For more information, see [Amazon Elastic Block Store](#) and [Amazon EBS Encryption](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.create_snapshot(
    DryRun=True|False,
    VolumeId='string',
    Description='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VolumeId** (*string*) – **[REQUIRED]**
The ID of the EBS volume.
- **Description** (*string*) – A description for the snapshot.

Return type dict

Returns**Response Syntax**

```
{
    'SnapshotId': 'string',
    'VolumeId': 'string',
    'State': 'pending'|'completed'|'error',
    'StateMessage': 'string',
    'StartTime': datetime(2015, 1, 1),
    'Progress': 'string',
    'OwnerId': 'string',
    'Description': 'string',
    'VolumeSize': 123,
    'OwnerAlias': 'string',
    'Tags': [
```

```
{
    'Key': 'string',
    'Value': 'string'
},
],
'Encrypted': True|False,
'KmsKeyId': 'string',
'DataEncryptionKeyId': 'string'
}
```

Response Structure

- *(dict)* –

Information about the snapshot.

- **SnapshotId** (*string*) –

The ID of the snapshot. Each snapshot receives a unique identifier when it is created.

- **VolumeId** (*string*) –

The ID of the volume that was used to create the snapshot.

- **State** (*string*) –

The snapshot state.

- **StateMessage** (*string*) –

Encrypted Amazon EBS snapshots are copied asynchronously. If a snapshot copy operation fails (for example, if the proper AWS Key Management Service (AWS KMS) permissions are not obtained) this field displays error state details to help you diagnose why the error occurred. This parameter is only returned by the DescribeSnapshots API operation.

- **StartTime** (*datetime*) –

The time stamp when the snapshot was initiated.

- **Progress** (*string*) –

The progress of the snapshot, as a percentage.

- **OwnerId** (*string*) –

The AWS account ID of the EBS snapshot owner.

- **Description** (*string*) –

The description for the snapshot.

- **VolumeSize** (*integer*) –

The size of the volume, in GiB.

- **OwnerAlias** (*string*) –

The AWS account alias (for example, `amazon`, `self`) or AWS account ID that owns the snapshot.

- **Tags** (*list*) –

Any tags assigned to the snapshot.

 * *(dict)* –

 Describes a tag.

- **Key** (*string*) –

 The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

- **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

- **Encrypted** (*boolean*) –

Indicates whether the snapshot is encrypted.

- **KmsKeyId** (*string*) –

The full ARN of the AWS Key Management Service (AWS KMS) customer master key (CMK) that was used to protect the volume encryption key for the parent volume.

- **DataEncryptionKeyId** (*string*) –

The data encryption key identifier for the snapshot. This value is a unique identifier that corresponds to the data encryption key that was used to encrypt the original volume or snapshot copy. Because data encryption keys are inherited by volumes created from snapshots, and vice versa, if snapshots share the same data encryption key identifier, then they belong to the same volume/snapshot lineage. This parameter is only returned by the DescribeSnapshots API operation.

create_spot_datafeed_subscription (***kwargs*)

Creates a data feed for Spot instances, enabling you to view Spot instance usage logs. You can create one data feed per AWS account. For more information, see [Spot Instance Data Feed](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.create_spot_datafeed_subscription(
    DryRun=True|False,
    Bucket='string',
    Prefix='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Bucket** (*string*) – **[REQUIRED]**
The Amazon S3 bucket in which to store the Spot instance data feed.
- **Prefix** (*string*) – A prefix for the data feed file names.

Return type dict

Returns

Response Syntax

```
{
  'SpotDatafeedSubscription': {
    'OwnerId': 'string',
    'Bucket': 'string',
    'Prefix': 'string',
    'State': 'Active'|'Inactive',
    'Fault': {
      'Code': 'string',
```

```
        'Message': 'string'
    }
}
```

Response Structure

- *(dict)* –

Contains the output of `CreateSpotDatafeedSubscription`.

- **SpotDatafeedSubscription** (*dict*) –

The Spot instance data feed subscription.

- * **OwnerId** (*string*) –

The AWS account ID of the account.

- * **Bucket** (*string*) –

The Amazon S3 bucket where the Spot instance data feed is located.

- * **Prefix** (*string*) –

The prefix that is prepended to data feed files.

- * **State** (*string*) –

The state of the Spot instance data feed subscription.

- * **Fault** (*dict*) –

The fault codes for the Spot instance request, if any.

- **Code** (*string*) –

The reason code for the Spot instance state change.

- **Message** (*string*) –

The message for the Spot instance state change.

`create_subnet` (***kwargs*)

Creates a subnet in an existing VPC.

When you create each subnet, you provide the VPC ID and the CIDR block you want for the subnet. After you create a subnet, you can't change its CIDR block. The subnet's CIDR block can be the same as the VPC's CIDR block (assuming you want only a single subnet in the VPC), or a subset of the VPC's CIDR block. If you create more than one subnet in a VPC, the subnets' CIDR blocks must not overlap. The smallest subnet (and VPC) you can create uses a /28 netmask (16 IP addresses), and the largest uses a /16 netmask (65,536 IP addresses).

Warning: AWS reserves both the first four and the last IP address in each subnet's CIDR block. They're not available for use.

If you add more than one subnet to a VPC, they're set up in a star topology with a logical router in the middle.

If you launch an instance in a VPC using an Amazon EBS-backed AMI, the IP address doesn't change if you stop and restart the instance (unlike a similar instance launched outside a VPC, which gets a new IP address when restarted). It's therefore possible to have a subnet with no running instances (they're all stopped), but no remaining IP addresses available.

For more information about subnets, see [Your VPC and Subnets](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.create_subnet(
    DryRun=True|False,
    VpcId='string',
    CidrBlock='string',
    AvailabilityZone='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcId** (*string*) – **[REQUIRED]**
The ID of the VPC.
- **CidrBlock** (*string*) – **[REQUIRED]**
The network range for the subnet, in CIDR notation. For example, `10.0.0.0/24`.
- **AvailabilityZone** (*string*) – The Availability Zone for the subnet.

Default: Amazon EC2 selects one for you (recommended).

Return type dict

Returns

Response Syntax

```
{
  'Subnet': {
    'SubnetId': 'string',
    'State': 'pending'|'available',
    'VpcId': 'string',
    'CidrBlock': 'string',
    'AvailableIpAddressCount': 123,
    'AvailabilityZone': 'string',
    'DefaultForAz': True|False,
    'MapPublicIpOnLaunch': True|False,
    'Tags': [
      {
        'Key': 'string',
        'Value': 'string'
      },
    ],
  }
}
```

Response Structure

- (*dict*) –
 - **Subnet** (*dict*) –
Information about the subnet.
 - * **SubnetId** (*string*) –
The ID of the subnet.
 - * **State** (*string*) –
The current state of the subnet.
 - * **VpcId** (*string*) –

The ID of the VPC the subnet is in.

* **CidrBlock** (*string*) –

The CIDR block assigned to the subnet.

* **AvailableIpAddressCount** (*integer*) –

The number of unused IP addresses in the subnet. Note that the IP addresses for any stopped instances are considered unavailable.

* **AvailabilityZone** (*string*) –

The Availability Zone of the subnet.

* **DefaultForAz** (*boolean*) –

Indicates whether this is the default subnet for the Availability Zone.

* **MapPublicIpOnLaunch** (*boolean*) –

Indicates whether instances launched in this subnet receive a public IP address.

* **Tags** (*list*) –

Any tags assigned to the subnet.

· (*dict*) –

Describes a tag.

· **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

· **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

create_tags (***kwargs*)

Adds or overwrites one or more tags for the specified Amazon EC2 resource or resources. Each resource can have a maximum of 10 tags. Each tag consists of a key and optional value. Tag keys must be unique per resource.

For more information about tags, see [Tagging Your Resources](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.create_tags(
    DryRun=True|False,
    Resources=[
        'string',
    ],
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **Resources** (*list*) – [REQUIRED]

The IDs of one or more resources to tag. For example, `ami-1a2b3c4d`.

– (*string*) –

- **Tags** (*list*) – [REQUIRED]

One or more tags. The `value` parameter is required, but if you don't want the tag to have a value, specify the parameter with no value, and we set the value to an empty string.

– (*dict*) –

Describes a tag.

* **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

* **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

Returns None

create_volume (***kwargs*)

Creates an EBS volume that can be attached to an instance in the same Availability Zone. The volume is created in the regional endpoint that you send the HTTP request to. For more information see [Regions and Endpoints](#).

You can create a new empty volume or restore a volume from an EBS snapshot. Any AWS Marketplace product codes from the snapshot are propagated to the volume.

You can create encrypted volumes with the `Encrypted` parameter. Encrypted volumes may only be attached to instances that support Amazon EBS encryption. Volumes that are created from encrypted snapshots are also automatically encrypted. For more information, see [Amazon EBS Encryption](#) in the *Amazon Elastic Compute Cloud User Guide*.

For more information, see [Creating or Restoring an Amazon EBS Volume](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.create_volume(
    DryRun=True|False,
    Size=123,
    SnapshotId='string',
    AvailabilityZone='string',
    VolumeType='standard'|'io1'|'gp2',
    Iops=123,
    Encrypted=True|False,
    KmsKeyId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Size** (*integer*) – The size of the volume, in GiBs.

Constraints: 1–1024 for standard volumes, 1–16384 for gp2 volumes, and 4–16384 for io1 volumes. If you specify a snapshot, the volume size must be equal to or larger than the snapshot size.

Default: If you're creating the volume from a snapshot and don't specify a volume size, the default is the snapshot size.

- **SnapshotId** (*string*) – The snapshot from which to create the volume.
- **AvailabilityZone** (*string*) – **[REQUIRED]**

The Availability Zone in which to create the volume. Use `DescribeAvailabilityZones` to list the Availability Zones that are currently available to you.

- **VolumeType** (*string*) – The volume type. This can be `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, or `standard` for Magnetic volumes.

Default: `standard`

- **Iops** (*integer*) – Only valid for Provisioned IOPS (SSD) volumes. The number of I/O operations per second (IOPS) to provision for the volume, with a maximum ratio of 30 IOPS/GiB.

Constraint: Range is 100 to 20000 for Provisioned IOPS (SSD) volumes

- **Encrypted** (*boolean*) – Specifies whether the volume should be encrypted. Encrypted Amazon EBS volumes may only be attached to instances that support Amazon EBS encryption. Volumes that are created from encrypted snapshots are automatically encrypted. There is no way to create an encrypted volume from an unencrypted snapshot or vice versa. If your AMI uses encrypted volumes, you can only launch it on supported instance types. For more information, see [Amazon EBS Encryption](#) in the *Amazon Elastic Compute Cloud User Guide*.
- **KmsKeyId** (*string*) – The full ARN of the AWS Key Management Service (AWS KMS) customer master key (CMK) to use when creating the encrypted volume. This parameter is only required if you want to use a non-default CMK; if this parameter is not specified, the default CMK for EBS is used. The ARN contains the `arn:aws:kms` namespace, followed by the region of the CMK, the AWS account ID of the CMK owner, the `key` namespace, and then the CMK ID. For example, `arn:aws:kms:us-east-1:012345678910:key/abcd1234-a123-456a-a12b-a123b4cd56ef`. If a `KmsKeyId` is specified, the `Encrypted` flag must also be set.

Return type dict

Returns

Response Syntax

```
{
    'VolumeId': 'string',
    'Size': 123,
    'SnapshotId': 'string',
    'AvailabilityZone': 'string',
    'State': 'creating'|'available'|'in-use'|'deleting'|'deleted'|'error',
    'CreateTime': datetime(2015, 1, 1),
    'Attachments': [
```

```

    {
        'VolumeId': 'string',
        'InstanceId': 'string',
        'Device': 'string',
        'State': 'attaching'|'attached'|'detaching'|'detached',
        'AttachTime': datetime(2015, 1, 1),
        'DeleteOnTermination': True|False
    },
],
'Tags': [
    {
        'Key': 'string',
        'Value': 'string'
    },
],
'VolumeType': 'standard'|'io1'|'gp2',
'Iops': 123,
'Encrypted': True|False,
'KmsKeyId': 'string'
}

```

Response Structure

- (*dict*) –

Information about the volume.

- **VolumeId** (*string*) –

The ID of the volume.

- **Size** (*integer*) –

The size of the volume, in GiBs.

- **SnapshotId** (*string*) –

The snapshot from which the volume was created, if applicable.

- **AvailabilityZone** (*string*) –

The Availability Zone for the volume.

- **State** (*string*) –

The volume state.

- **CreateTime** (*datetime*) –

The time stamp when volume creation was initiated.

- **Attachments** (*list*) –

Information about the volume attachments.

* (*dict*) –

Describes volume attachment details.

- **VolumeId** (*string*) –

The ID of the volume.

- **InstanceId** (*string*) –

The ID of the instance.

- **Device** (*string*) –

The device name.

- **State** (*string*) –

The attachment state of the volume.

- **AttachTime** (*datetime*) –
The time stamp when the attachment initiated.
- **DeleteOnTermination** (*boolean*) –
Indicates whether the EBS volume is deleted on instance termination.
- **Tags** (*list*) –
Any tags assigned to the volume.
 - * (*dict*) –
Describes a tag.
 - **Key** (*string*) –
The key of the tag.
Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`
 - **Value** (*string*) –
The value of the tag.
Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.
- **VolumeType** (*string*) –
The volume type. This can be `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, or `standard` for Magnetic volumes.
- **Iops** (*integer*) –
The number of I/O operations per second (IOPS) that the volume supports. For Provisioned IOPS (SSD) volumes, this represents the number of IOPS that are provisioned for the volume. For General Purpose (SSD) volumes, this represents the baseline performance of the volume and the rate at which the volume accumulates I/O credits for bursting. For more information on General Purpose (SSD) baseline performance, I/O credits, and bursting, see [Amazon EBS Volume Types](#) in the *Amazon Elastic Compute Cloud User Guide* .

Constraint: Range is 100 to 20000 for Provisioned IOPS (SSD) volumes and 3 to 10000 for General Purpose (SSD) volumes.

Condition: This parameter is required for requests to create `io1` volumes; it is not used in requests to create `standard` or `gp2` volumes.
- **Encrypted** (*boolean*) –
Indicates whether the volume will be encrypted.
- **KmsKeyId** (*string*) –
The full ARN of the AWS Key Management Service (AWS KMS) customer master key (CMK) that was used to protect the volume encryption key for the volume.

create_vpc (***kwargs*)

Creates a VPC with the specified CIDR block.

The smallest VPC you can create uses a /28 netmask (16 IP addresses), and the largest uses a /16 netmask (65,536 IP addresses). To help you decide how big to make your VPC, see [Your VPC and Subnets](#) in the *Amazon Virtual Private Cloud User Guide* .

By default, each instance you launch in the VPC has the default DHCP options, which includes only a default DNS server that we provide (AmazonProvidedDNS). For more information about DHCP options, see [DHCP Options Sets](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.create_vpc(
    DryRun=True|False,
    CidrBlock='string',
    InstanceTenancy='default'|'dedicated'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **CidrBlock** (*string*) – **[REQUIRED]**
The network range for the VPC, in CIDR notation. For example, `10.0.0.0/16`.
- **InstanceTenancy** (*string*) – The supported tenancy options for instances launched into the VPC. A value of `default` means that instances can be launched with any tenancy; a value of `dedicated` means all instances launched into the VPC are launched as dedicated tenancy instances regardless of the tenancy assigned to the instance at launch. Dedicated tenancy instances run on single-tenant hardware.

Default: `default`

Return type dict

Returns

Response Syntax

```
{
  'Vpc': {
    'VpcId': 'string',
    'State': 'pending'|'available',
    'CidrBlock': 'string',
    'DhcpOptionsId': 'string',
    'Tags': [
      {
        'Key': 'string',
        'Value': 'string'
      },
    ],
    'InstanceTenancy': 'default'|'dedicated',
    'IsDefault': True|False
  }
}
```

Response Structure

- (*dict*) –
 - **Vpc** (*dict*) –
Information about the VPC.
 - * **VpcId** (*string*) –
The ID of the VPC.

- * **State** (*string*) –
The current state of the VPC.
- * **CidrBlock** (*string*) –
The CIDR block for the VPC.
- * **DhcpOptionsId** (*string*) –
The ID of the set of DHCP options you’ve associated with the VPC (or `default` if the default options are associated with the VPC).
- * **Tags** (*list*) –
Any tags assigned to the VPC.
 - (*dict*) –
Describes a tag.
 - **Key** (*string*) –
The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`
 - **Value** (*string*) –
The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.
- * **InstanceTenancy** (*string*) –
The allowed tenancy of instances launched into the VPC.
- * **IsDefault** (*boolean*) –
Indicates whether the VPC is the default VPC.

create_vpc_endpoint (***kwargs*)

Creates a VPC endpoint for a specified AWS service. An endpoint enables you to create a private connection between your VPC and another AWS service in your account. You can specify an endpoint policy to attach to the endpoint that will control access to the service from your VPC. You can also specify the VPC route tables that use the endpoint.

Currently, only endpoints to Amazon S3 are supported.

Request Syntax

```
response = client.create_vpc_endpoint(
    DryRun=True|False,
    VpcId='string',
    ServiceName='string',
    PolicyDocument='string',
    RouteTableIds=[
        'string',
    ],
    ClientToken='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **VpcId** (*string*) – [REQUIRED]

The ID of the VPC in which the endpoint will be used.

- **ServiceName** (*string*) – [REQUIRED]

The AWS service name, in the form `com.amazonaws.*region*.*service*`. To get a list of available services, use the `DescribeVpcEndpointServices` request.

- **PolicyDocument** (*string*) – A policy to attach to the endpoint that controls access to the service. The policy must be in valid JSON format. If this parameter is not specified, we attach a default policy that allows full access to the service.

- **RouteTableIds** (*list*) – One or more route table IDs.

– (*string*) –

- **ClientToken** (*string*) – Unique, case-sensitive identifier you provide to ensure the idempotency of the request. For more information, see [How to Ensure Idempotency](#).

Return type dict

Returns

Response Syntax

```
{
  'VpcEndpoint': {
    'VpcEndpointId': 'string',
    'VpcId': 'string',
    'ServiceName': 'string',
    'State': 'Pending'|'Available'|'Deleting'|'Deleted',
    'PolicyDocument': 'string',
    'RouteTableIds': [
      'string',
    ],
    'CreationTimestamp': datetime(2015, 1, 1)
  },
  'ClientToken': 'string'
}
```

Response Structure

- (*dict*) –

- **VpcEndpoint** (*dict*) –

Information about the endpoint.

- * **VpcEndpointId** (*string*) –

The ID of the VPC endpoint.

- * **VpcId** (*string*) –

The ID of the VPC to which the endpoint is associated.

- * **ServiceName** (*string*) –

The name of the AWS service to which the endpoint is associated.

- * **State** (*string*) –

The state of the VPC endpoint.

- * **PolicyDocument** (*string*) –

The policy document associated with the endpoint.

- * **RouteTableIds** (*list*) –

One or more route tables associated with the endpoint.

· (*string*) –

* **CreationTimestamp** (*datetime*) –

The date and time the VPC endpoint was created.

– **ClientToken** (*string*) –

Unique, case-sensitive identifier you provide to ensure the idempotency of the request.

create_vpc_peering_connection (**kwargs)

Requests a VPC peering connection between two VPCs: a requester VPC that you own and a peer VPC with which to create the connection. The peer VPC can belong to another AWS account. The requester VPC and peer VPC cannot have overlapping CIDR blocks.

The owner of the peer VPC must accept the peering request to activate the peering connection. The VPC peering connection request expires after 7 days, after which it cannot be accepted or rejected.

A `CreateVpcPeeringConnection` request between VPCs with overlapping CIDR blocks results in the VPC peering connection having a status of `failed`.

Request Syntax

```
response = client.create_vpc_peering_connection(
    DryRun=True|False,
    VpcId='string',
    PeerVpcId='string',
    PeerOwnerId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcId** (*string*) – The ID of the requester VPC.
- **PeerVpcId** (*string*) – The ID of the VPC with which you are creating the VPC peering connection.
- **PeerOwnerId** (*string*) – The AWS account ID of the owner of the peer VPC.

Default: Your AWS account ID

Return type dict

Returns

Response Syntax

```
{
  'VpcPeeringConnection': {
    'AccepterVpcInfo': {
      'CidrBlock': 'string',
      'OwnerId': 'string',
      'VpcId': 'string'
    },
    'ExpirationTime': datetime(2015, 1, 1),
    'RequesterVpcInfo': {
      'CidrBlock': 'string',
      'OwnerId': 'string',
      'VpcId': 'string'
    },
    'Status': {
      'Code': 'initiating-request'|'pending-acceptance'|'active'|'deleted'|'
```



```

        'Message': 'string'
    },
    'Tags': [
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
    'VpcPeeringConnectionId': 'string'
}

```

Response Structure

- *(dict)* –
 - **VpcPeeringConnection** (*dict*) –

Information about the VPC peering connection.

 - * **AccepterVpcInfo** (*dict*) –

The information of the peer VPC.

 - **CidrBlock** (*string*) –

The CIDR block for the VPC.
 - **OwnerId** (*string*) –

The AWS account ID of the VPC owner.
 - **VpcId** (*string*) –

The ID of the VPC.
 - * **ExpirationTime** (*datetime*) –

The time that an unaccepted VPC peering connection will expire.
 - * **RequesterVpcInfo** (*dict*) –

The information of the requester VPC.

 - **CidrBlock** (*string*) –

The CIDR block for the VPC.
 - **OwnerId** (*string*) –

The AWS account ID of the VPC owner.
 - **VpcId** (*string*) –

The ID of the VPC.
 - * **Status** (*dict*) –

The status of the VPC peering connection.

 - **Code** (*string*) –

The status of the VPC peering connection.
 - **Message** (*string*) –

A message that provides more information about the status, if applicable.
 - * **Tags** (*list*) –

Any tags assigned to the resource.

 - (*dict*) –

Describes a tag.
 - **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

- **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

- * **VpcPeeringConnectionId** (*string*) –

The ID of the VPC peering connection.

create_vpn_connection (***kwargs*)

Creates a VPN connection between an existing virtual private gateway and a VPN customer gateway. The only supported connection type is `ipsec.1`.

The response includes information that you need to give to your network administrator to configure your customer gateway.

Warning: We strongly recommend that you use HTTPS when calling this operation because the response contains sensitive cryptographic information for configuring your customer gateway.

If you decide to shut down your VPN connection for any reason and later create a new VPN connection, you must reconfigure your customer gateway with the new information returned from this call.

For more information about VPN connections, see [Adding a Hardware Virtual Private Gateway to Your VPC](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.create_vpn_connection(
    DryRun=True|False,
    Type='string',
    CustomerGatewayId='string',
    VpnGatewayId='string',
    Options={
        'StaticRoutesOnly': True|False
    }
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Type** (*string*) – **[REQUIRED]**
The type of VPN connection (`ipsec.1`).
- **CustomerGatewayId** (*string*) – **[REQUIRED]**
The ID of the customer gateway.
- **VpnGatewayId** (*string*) – **[REQUIRED]**
The ID of the virtual private gateway.
- **Options** (*dict*) – Indicates whether the VPN connection requires static routes. If you are creating a VPN connection for a device that does not support BGP, you must specify `true`.

Default: `false`

– **StaticRoutesOnly** (*boolean*) –

Indicates whether the VPN connection uses static routes only. Static routes must be used for devices that don't support BGP.

Return type dict

Returns

Response Syntax

```
{
  'VpnConnection': {
    'VpnConnectionId': 'string',
    'State': 'pending'|'available'|'deleting'|'deleted',
    'CustomerGatewayConfiguration': 'string',
    'Type': 'ipsec.1',
    'CustomerGatewayId': 'string',
    'VpnGatewayId': 'string',
    'Tags': [
      {
        'Key': 'string',
        'Value': 'string'
      },
    ],
    'VgwTelemetry': [
      {
        'OutsideIpAddress': 'string',
        'Status': 'UP'|'DOWN',
        'LastStatusChange': datetime(2015, 1, 1),
        'StatusMessage': 'string',
        'AcceptedRouteCount': 123
      },
    ],
    'Options': {
      'StaticRoutesOnly': True|False
    },
    'Routes': [
      {
        'DestinationCidrBlock': 'string',
        'Source': 'Static',
        'State': 'pending'|'available'|'deleting'|'deleted'
      },
    ]
  }
}
```

Response Structure

- (*dict*) –

– **VpnConnection** (*dict*) –

Information about the VPN connection.

- * **VpnConnectionId** (*string*) –

The ID of the VPN connection.

- * **State** (*string*) –

The current state of the VPN connection.

- * **CustomerGatewayConfiguration** (*string*) –

The configuration information for the VPN connection's customer gateway (in the native XML format). This element is always present

in the `CreateVpnConnection` response; however, it's present in the `DescribeVpnConnections` response only if the VPN connection is in the `pending` or `available` state.

* **Type** (*string*) –

The type of VPN connection.

* **CustomerGatewayId** (*string*) –

The ID of the customer gateway at your end of the VPN connection.

* **VpnGatewayId** (*string*) –

The ID of the virtual private gateway at the AWS side of the VPN connection.

* **Tags** (*list*) –

Any tags assigned to the VPN connection.

· (*dict*) –

Describes a tag.

· **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

· **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

* **VgwTelemetry** (*list*) –

Information about the VPN tunnel.

· (*dict*) –

Describes telemetry for a VPN tunnel.

· **OutsideIpAddress** (*string*) –

The Internet-routable IP address of the virtual private gateway's outside interface.

· **Status** (*string*) –

The status of the VPN tunnel.

· **LastStatusChange** (*datetime*) –

The date and time of the last change in status.

· **StatusMessage** (*string*) –

If an error occurs, a description of the error.

· **AcceptedRouteCount** (*integer*) –

The number of accepted routes.

* **Options** (*dict*) –

The VPN connection options.

· **StaticRoutesOnly** (*boolean*) –

Indicates whether the VPN connection uses static routes only. Static routes must be used for devices that don't support BGP.

* **Routes** (*list*) –

The static routes associated with the VPN connection.

- *(dict)* –
Describes a static route for a VPN connection.
- **DestinationCidrBlock** (*string*) –
The CIDR block associated with the local subnet of the customer data center.
- **Source** (*string*) –
Indicates how the routes were provided.
- **State** (*string*) –
The current state of the static route.

create_vpn_connection_route (***kwargs*)

Creates a static route associated with a VPN connection between an existing virtual private gateway and a VPN customer gateway. The static route allows traffic to be routed from the virtual private gateway to the VPN customer gateway.

For more information about VPN connections, see [Adding a Hardware Virtual Private Gateway to Your VPC](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.create_vpn_connection_route(
    VpnConnectionId='string',
    DestinationCidrBlock='string'
)
```

Parameters

- **VpnConnectionId** (*string*) – **[REQUIRED]**
The ID of the VPN connection.
- **DestinationCidrBlock** (*string*) – **[REQUIRED]**
The CIDR block associated with the local subnet of the customer network.

Returns None

create_vpn_gateway (***kwargs*)

Creates a virtual private gateway. A virtual private gateway is the endpoint on the VPC side of your VPN connection. You can create a virtual private gateway before creating the VPC itself.

For more information about virtual private gateways, see [Adding a Hardware Virtual Private Gateway to Your VPC](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.create_vpn_gateway(
    DryRun=True|False,
    Type='ipsec.1',
    AvailabilityZone='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Type** (*string*) – **[REQUIRED]**
The type of VPN connection this virtual private gateway supports.

- **AvailabilityZone** (*string*) – The Availability Zone for the virtual private gateway.

Return type dict

Returns

Response Syntax

```
{
  'VpnGateway': {
    'VpnGatewayId': 'string',
    'State': 'pending'|'available'|'deleting'|'deleted',
    'Type': 'ipsec.1',
    'AvailabilityZone': 'string',
    'VpcAttachments': [
      {
        'VpcId': 'string',
        'State': 'attaching'|'attached'|'detaching'|'detached'
      },
    ],
    'Tags': [
      {
        'Key': 'string',
        'Value': 'string'
      },
    ],
  }
}
```

Response Structure

- (*dict*) –

– **VpnGateway** (*dict*) –

Information about the virtual private gateway.

- * **VpnGatewayId** (*string*) –

The ID of the virtual private gateway.

- * **State** (*string*) –

The current state of the virtual private gateway.

- * **Type** (*string*) –

The type of VPN connection the virtual private gateway supports.

- * **AvailabilityZone** (*string*) –

The Availability Zone where the virtual private gateway was created.

- * **VpcAttachments** (*list*) –

Any VPCs attached to the virtual private gateway.

- (*dict*) –

Describes an attachment between a virtual private gateway and a VPC.

- **VpcId** (*string*) –

The ID of the VPC.

- **State** (*string*) –

The current state of the attachment.

- * **Tags** (*list*) –

Any tags assigned to the virtual private gateway.

- *(dict)* –
Describes a tag.
- **Key** (*string*) –
The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`
- **Value** (*string*) –
The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

delete_customer_gateway (***kwargs*)

Deletes the specified customer gateway. You must delete the VPN connection before you can delete the customer gateway.

Request Syntax

```
response = client.delete_customer_gateway(
    DryRun=True|False,
    CustomerGatewayId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **CustomerGatewayId** (*string*) – **[REQUIRED]**
The ID of the customer gateway.

Returns None

delete_dhcp_options (***kwargs*)

Deletes the specified set of DHCP options. You must disassociate the set of DHCP options before you can delete it. You can disassociate the set of DHCP options by associating either a new set of options or the default set of options with the VPC.

Request Syntax

```
response = client.delete_dhcp_options(
    DryRun=True|False,
    DhcpOptionsId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **DhcpOptionsId** (*string*) – **[REQUIRED]**
The ID of the DHCP options set.

Returns None

delete_flow_logs (**kwargs)
Deletes one or more flow logs.

Request Syntax

```
response = client.delete_flow_logs(
    FlowLogIds=[
        'string',
    ]
)
```

Parameters **FlowLogIds** (list) – [REQUIRED]

One or more flow log IDs.

- (string) –

Return type dict

Returns

Response Syntax

```
{
    'Unsuccessful': [
        {
            'ResourceId': 'string',
            'Error': {
                'Code': 'string',
                'Message': 'string'
            }
        },
    ]
}
```

Response Structure

- (dict) –
 - **Unsuccessful** (list) –

Information about the flow logs that could not be deleted successfully.

* (dict) –

Information about items that were not successfully processed in a batch call.

- **ResourceId** (string) –

The ID of the resource.

- **Error** (dict) –

Information about the error.

- **Code** (string) –

The error code.

- **Message** (string) –

The error message accompanying the error code.

delete_internet_gateway (**kwargs)

Deletes the specified Internet gateway. You must detach the Internet gateway from the VPC before you can delete it.

Request Syntax


```
response = client.delete_internet_gateway(
    DryRun=True|False,
    InternetGatewayId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InternetGatewayId** (*string*) – [REQUIRED]

The ID of the Internet gateway.

Returns None

delete_key_pair (***kwargs*)

Deletes the specified key pair, by removing the public key from Amazon EC2.

Request Syntax

```
response = client.delete_key_pair(
    DryRun=True|False,
    KeyName='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **KeyName** (*string*) – [REQUIRED]

The name of the key pair.

Returns None

delete_network_acl (***kwargs*)

Deletes the specified network ACL. You can't delete the ACL if it's associated with any subnets. You can't delete the default network ACL.

Request Syntax

```
response = client.delete_network_acl(
    DryRun=True|False,
    NetworkAclId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **NetworkAclId** (*string*) – [REQUIRED]

The ID of the network ACL.

Returns None

delete_network_acl_entry (***kwargs*)

Deletes the specified ingress or egress entry (rule) from the specified network ACL.

Request Syntax

```
response = client.delete_network_acl_entry(  
    DryRun=True|False,  
    NetworkAclId='string',  
    RuleNumber=123,  
    Egress=True|False  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **NetworkAclId** (*string*) – **[REQUIRED]**
The ID of the network ACL.
- **RuleNumber** (*integer*) – **[REQUIRED]**
The rule number of the entry to delete.
- **Egress** (*boolean*) – **[REQUIRED]**
Indicates whether the rule is an egress rule.

Returns None

delete_network_interface (***kwargs*)

Deletes the specified network interface. You must detach the network interface before you can delete it.

Request Syntax

```
response = client.delete_network_interface(  
    DryRun=True|False,  
    NetworkInterfaceId='string'  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **NetworkInterfaceId** (*string*) – **[REQUIRED]**
The ID of the network interface.

Returns None

delete_placement_group (***kwargs*)

Deletes the specified placement group. You must terminate all instances in the placement group before you can delete the placement group. For more information about placement groups and cluster instances, see [Cluster Instances](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.delete_placement_group(  
    DryRun=True|False,  
    GroupName='string'  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **GroupName** (*string*) – [REQUIRED]

The name of the placement group.

Returns None

delete_route (***kwargs*)

Deletes the specified route from the specified route table.

Request Syntax

```
response = client.delete_route(
    DryRun=True|False,
    RouteTableId='string',
    DestinationCidrBlock='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **RouteTableId** (*string*) – [REQUIRED]

The ID of the route table.

- **DestinationCidrBlock** (*string*) – [REQUIRED]

The CIDR range for the route. The value you specify must match the CIDR for the route exactly.

Returns None

delete_route_table (***kwargs*)

Deletes the specified route table. You must disassociate the route table from any subnets before you can delete it. You can't delete the main route table.

Request Syntax

```
response = client.delete_route_table(
    DryRun=True|False,
    RouteTableId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **RouteTableId** (*string*) – [REQUIRED]

The ID of the route table.

Returns None

delete_security_group (***kwargs*)

Deletes a security group.

If you attempt to delete a security group that is associated with an instance, or is referenced by another security group, the operation fails with `InvalidGroup.InUse` in EC2-Classic or `DependencyViolation` in EC2-VPC.

Request Syntax

```
response = client.delete_security_group(
    DryRun=True|False,
    GroupName='string',
    GroupId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **GroupName** (*string*) – [EC2-Classic, default VPC] The name of the security group. You can specify either the security group name or the security group ID.
- **GroupId** (*string*) – The ID of the security group. Required for a nondefault VPC.

Returns None

delete_snapshot (***kwargs*)

Deletes the specified snapshot.

When you make periodic snapshots of a volume, the snapshots are incremental, and only the blocks on the device that have changed since your last snapshot are saved in the new snapshot. When you delete a snapshot, only the data not needed for any other snapshot is removed. So regardless of which prior snapshots have been deleted, all active snapshots will have access to all the information needed to restore the volume.

You cannot delete a snapshot of the root device of an EBS volume used by a registered AMI. You must first de-register the AMI before you can delete the snapshot.

For more information, see [Deleting an Amazon EBS Snapshot](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.delete_snapshot(
    DryRun=True|False,
    SnapshotId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SnapshotId** (*string*) – **[REQUIRED]**
The ID of the EBS snapshot.

Returns None

delete_spot_datafeed_subscription (***kwargs*)

Deletes the data feed for Spot instances.

Request Syntax

```
response = client.delete_spot_datafeed_subscription(
    DryRun=True|False
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns `None`

delete_subnet (***kwargs*)

Deletes the specified subnet. You must terminate all running instances in the subnet before you can delete the subnet.

Request Syntax

```
response = client.delete_subnet(
    DryRun=True|False,
    SubnetId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SubnetId** (*string*) – **[REQUIRED]**

The ID of the subnet.

Returns `None`

delete_tags (***kwargs*)

Deletes the specified set of tags from the specified set of resources. This call is designed to follow a `DescribeTags` request.

For more information about tags, see [Tagging Your Resources](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.delete_tags(
    DryRun=True|False,
    Resources=[
        'string',
    ],
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **Resources** (*list*) – [REQUIRED]

The ID of the resource. For example, ami-1a2b3c4d. You can specify more than one resource ID.

- (*string*) –

- **Tags** (*list*) – One or more tags to delete. If you omit the `value` parameter, we delete the tag regardless of its value. If you specify this parameter with an empty string as the value, we delete the key only if its value is an empty string.

- (*dict*) –

Describes a tag.

- * **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

- * **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

Returns None

delete_volume (***kwargs*)

Deletes the specified EBS volume. The volume must be in the `available` state (not attached to an instance).

Note: The volume may remain in the `deleting` state for several minutes.

For more information, see [Deleting an Amazon EBS Volume](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.delete_volume(  
    DryRun=True|False,  
    VolumeId='string'  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VolumeId** (*string*) – [REQUIRED]

The ID of the volume.

Returns None

delete_vpc (***kwargs*)

Deletes the specified VPC. You must detach or delete all gateways and resources that are associated with the VPC before you can delete it. For example, you must terminate all instances running in the VPC, delete all security groups associated with the VPC (except the default one), delete all route tables associated with the VPC (except the default one), and so on.

Request Syntax

```
response = client.delete_vpc(
    DryRun=True|False,
    VpcId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcId** (*string*) – **[REQUIRED]**

The ID of the VPC.

Returns None

delete_vpc_endpoints (***kwargs*)

Deletes one or more specified VPC endpoints. Deleting the endpoint also deletes the endpoint routes in the route tables that were associated with the endpoint.

Request Syntax

```
response = client.delete_vpc_endpoints(
    DryRun=True|False,
    VpcEndpointIds=[
        'string',
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcEndpointIds** (*list*) – **[REQUIRED]**

One or more endpoint IDs.

– (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'Unsuccessful': [
        {
            'ResourceId': 'string',
            'Error': {
                'Code': 'string',
                'Message': 'string'
            }
        },
    ]
}
```

Response Structure

- (*dict*) –

– **Unsuccessful** (*list*) –

Information about the endpoints that were not successfully deleted.

* (*dict*) –

Information about items that were not successfully processed in a batch call.

- **ResourceId** (*string*) –

The ID of the resource.

- **Error** (*dict*) –

Information about the error.

- **Code** (*string*) –

The error code.

- **Message** (*string*) –

The error message accompanying the error code.

delete_vpc_peering_connection (**kwargs)

Deletes a VPC peering connection. Either the owner of the requester VPC or the owner of the peer VPC can delete the VPC peering connection if it's in the `active` state. The owner of the requester VPC can delete a VPC peering connection in the `pending-acceptance` state.

Request Syntax

```
response = client.delete_vpc_peering_connection(
    DryRun=True|False,
    VpcPeeringConnectionId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcPeeringConnectionId** (*string*) – **[REQUIRED]**

The ID of the VPC peering connection.

Return type dict

Returns

Response Syntax

```
{
    'Return': True|False
}
```

Response Structure

- (*dict*) –
 - **Return** (*boolean*) –

Returns `true` if the request succeeds; otherwise, it returns an error.

delete_vpn_connection (**kwargs)

Deletes the specified VPN connection.

If you're deleting the VPC and its associated components, we recommend that you detach the virtual private gateway from the VPC and delete the VPC before deleting the VPN connection. If you believe that the tunnel credentials for your VPN connection have been compromised, you can delete the VPN

connection and create a new one that has new keys, without needing to delete the VPC or virtual private gateway. If you create a new VPN connection, you must reconfigure the customer gateway using the new configuration information returned with the new VPN connection ID.

Request Syntax

```
response = client.delete_vpn_connection(
    DryRun=True|False,
    VpnConnectionId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpnConnectionId** (*string*) – **[REQUIRED]**

The ID of the VPN connection.

Returns None

`delete_vpn_connection_route` (**kwargs)

Deletes the specified static route associated with a VPN connection between an existing virtual private gateway and a VPN customer gateway. The static route allows traffic to be routed from the virtual private gateway to the VPN customer gateway.

Request Syntax

```
response = client.delete_vpn_connection_route(
    VpnConnectionId='string',
    DestinationCidrBlock='string'
)
```

Parameters

- **VpnConnectionId** (*string*) – **[REQUIRED]**
- **DestinationCidrBlock** (*string*) – **[REQUIRED]**

The CIDR block associated with the local subnet of the customer network.

Returns None

`delete_vpn_gateway` (**kwargs)

Deletes the specified virtual private gateway. We recommend that before you delete a virtual private gateway, you detach it from the VPC and delete the VPN connection. Note that you don't need to delete the virtual private gateway if you plan to delete and recreate the VPN connection between your VPC and your network.

Request Syntax

```
response = client.delete_vpn_gateway(
    DryRun=True|False,
    VpnGatewayId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If

you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **VpnGatewayId** (*string*) – [REQUIRED]

The ID of the virtual private gateway.

Returns None

deregister_image (***kwargs*)

Deregisters the specified AMI. After you deregister an AMI, it can't be used to launch new instances.

This command does not delete the AMI.

Request Syntax

```
response = client.deregister_image(  
    DryRun=True|False,  
    ImageId='string'  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **ImageId** (*string*) – [REQUIRED]

The ID of the AMI.

Returns None

describe_account_attributes (***kwargs*)

Describes attributes of your AWS account. The following are the supported account attributes:

- **supported-platforms** : Indicates whether your account can launch instances into EC2-Classic and EC2-VPC, or only into EC2-VPC.
- **default-vpc** : The ID of the default VPC for your account, or `none`.
- **max-instances** : The maximum number of On-Demand instances that you can run.
- **vpc-max-security-groups-per-interface** : The maximum number of security groups that you can assign to a network interface.
- **max-elastic-ips** : The maximum number of Elastic IP addresses that you can allocate for use with EC2-Classic.
- **vpc-max-elastic-ips** : The maximum number of Elastic IP addresses that you can allocate for use with EC2-VPC.

Request Syntax

```
response = client.describe_account_attributes(  
    DryRun=True|False,  
    AttributeNames=[  
        'supported-platforms'|'default-vpc',  
    ]  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **AttributeNames** (*list*) – One or more account attribute names.
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'AccountAttributes': [
    {
      'AttributeName': 'string',
      'AttributeValues': [
        {
          'AttributeValue': 'string'
        },
      ],
    },
  ]
}
```

Response Structure

- *(dict)* –
 - **AccountAttributes** (*list*) –

Information about one or more account attributes.

 - * *(dict)* –

Describes an account attribute.

 - **AttributeName** (*string*) –

The name of the account attribute.
 - **AttributeValues** (*list*) –

One or more values for the account attribute.
 - *(dict)* –

Describes a value of an account attribute.
 - **AttributeValue** (*string*) –

The value of the attribute.

describe_addresses (**kwargs)

Describes one or more of your Elastic IP addresses.

An Elastic IP address is for use in either the EC2-Classic platform or in a VPC. For more information, see [Elastic IP Addresses](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.describe_addresses(
    DryRun=True|False,
    PublicIps=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ],
        },
    ],
    AllocationIds=[
        'string',
    ],
```

```
]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **PublicIps** (*list*) – [EC2-Classic] One or more Elastic IP addresses.

Default: Describes all your Elastic IP addresses.

- (*string*) –
- **Filters** (*list*) – One or more filters. Filter names and values are case-sensitive.
 - `allocation-id` - [EC2-VPC] The allocation ID for the address.
 - `association-id` - [EC2-VPC] The association ID for the address.
 - `domain` - Indicates whether the address is for use in EC2-Classic (`standard`) or in a VPC (`vpc`).
 - `instance-id` - The ID of the instance the address is associated with, if any.
 - `network-interface-id` - [EC2-VPC] The ID of the network interface that the address is associated with, if any.
 - `network-interface-owner-id` - The AWS account ID of the owner.
 - `private-ip-address` - [EC2-VPC] The private IP address associated with the Elastic IP address.
 - `public-ip` - The Elastic IP address.
- (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

- **AllocationIds** (*list*) – [EC2-VPC] One or more allocation IDs.

Default: Describes all your Elastic IP addresses.

– (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'Addresses': [
    {
      'InstanceId': 'string',
      'PublicIp': 'string',
      'AllocationId': 'string',
      'AssociationId': 'string',
      'Domain': 'vpc'|'standard',
      'NetworkInterfaceId': 'string',
      'NetworkInterfaceOwnerId': 'string',
```

```

        'PrivateIpAddress': 'string'
    },
]
}

```

Response Structure

- *(dict)* –

- **Addresses** (*list*) –

Information about one or more Elastic IP addresses.

- * *(dict)* –

Describes an Elastic IP address.

- **InstanceId** (*string*) –

The ID of the instance that the address is associated with (if any).

- **PublicIp** (*string*) –

The Elastic IP address.

- **AllocationId** (*string*) –

The ID representing the allocation of the address for use with EC2-VPC.

- **AssociationId** (*string*) –

The ID representing the association of the address with an instance in a VPC.

- **Domain** (*string*) –

Indicates whether this Elastic IP address is for use with instances in EC2-Classic (*standard*) or instances in a VPC (*vpc*).

- **NetworkInterfaceId** (*string*) –

The ID of the network interface.

- **NetworkInterfaceOwnerId** (*string*) –

The ID of the AWS account that owns the network interface.

- **PrivateIpAddress** (*string*) –

The private IP address associated with the Elastic IP address.

describe_availability_zones (***kwargs*)

Describes one or more of the Availability Zones that are available to you. The results include zones only for the region you're currently using. If there is an event impacting an Availability Zone, you can use this request to view the state and any provided message for that Availability Zone.

For more information, see [Regions and Availability Zones](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```

response = client.describe_availability_zones(
    DryRun=True|False,
    ZoneNames=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',

```

```
        'Values': [
            'string',
        ]
    },
]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **ZoneNames** (*list*) – The names of one or more Availability Zones.
 - (*string*) –
- **Filters** (*list*) – One or more filters.
 - message - Information about the Availability Zone.
 - region-name - The name of the region for the Availability Zone (for example, `us-east-1`).
 - state - The state of the Availability Zone (`available` | `impaired` | `unavailable`).
 - zone-name - The name of the Availability Zone (for example, `us-east-1a`).
 - (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

- * **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

- * **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'AvailabilityZones': [
    {
      'ZoneName': 'string',
      'State': 'available',
      'RegionName': 'string',
      'Messages': [
        {
          'Message': 'string'
        },
      ],
    },
  ],
}
```

Response Structure

- (*dict*) –

– **AvailabilityZones** (*list*) –

Information about one or more Availability Zones.

* (*dict*) –

Describes an Availability Zone.

· **ZoneName** (*string*) –

The name of the Availability Zone.

· **State** (*string*) –

The state of the Availability Zone (available|impaired|unavailable).

· **RegionName** (*string*) –

The name of the region.

· **Messages** (*list*) –

Any messages about the Availability Zone.

· (*dict*) –

Describes a message about an Availability Zone.

· **Message** (*string*) –

The message about the Availability Zone.

describe_bundle_tasks (***kwargs*)

Describes one or more of your bundling tasks.

Note: Completed bundle tasks are listed for only a limited time. If your bundle task is no longer in the list, you can still register an AMI from it. Just use `RegisterImage` with the Amazon S3 bucket name and image manifest name you provided to the bundle task.

Request Syntax

```
response = client.describe_bundle_tasks(
    DryRun=True|False,
    BundleIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **BundleIds** (*list*) – One or more bundle task IDs.
Default: Describes all your bundle tasks.
– (*string*) –
- **Filters** (*list*) – One or more filters.

- `bundle-id` - The ID of the bundle task.
- `error-code` - If the task failed, the error code returned.
- `error-message` - If the task failed, the error message returned.
- `instance-id` - The ID of the instance.
- `progress` - The level of task completion, as a percentage (for example, 20%).
- `s3-bucket` - The Amazon S3 bucket to store the AMI.
- `s3-prefix` - The beginning of the AMI name.
- `start-time` - The time the task started (for example, 2013-09-15T17:15:20.000Z).
- `state` - The state of the task (pending|waiting-for-shutdown|bundling|storing|cancelling|complete|failed).
- `update-time` - The time of the most recent update for the task.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

Return type dict

Returns

Response Syntax

```
{
  'BundleTasks': [
    {
      'InstanceId': 'string',
      'BundleId': 'string',
      'State': 'pending'|'waiting-for-shutdown'|'bundling'|'storing'|'cancel',
      'StartTime': datetime(2015, 1, 1),
      'UpdateTime': datetime(2015, 1, 1),
      'Storage': {
        'S3': {
          'Bucket': 'string',
          'Prefix': 'string',
          'AWSAccessKeyId': 'string',
          'UploadPolicy': b'bytes',
          'UploadPolicySignature': 'string'
        }
      },
      'Progress': 'string',
      'BundleTaskError': {
        'Code': 'string',
        'Message': 'string'
      }
    },
  ],
}
```

Response Structure

- (*dict*) -

– **BundleTasks** (*list*) –

Information about one or more bundle tasks.

* (*dict*) –

Describes a bundle task.

· **InstanceId** (*string*) –

The ID of the instance associated with this bundle task.

· **BundleId** (*string*) –

The ID of the bundle task.

· **State** (*string*) –

The state of the task.

· **StartTime** (*datetime*) –

The time this task started.

· **UpdateTime** (*datetime*) –

The time of the most recent update for the task.

· **Storage** (*dict*) –

The Amazon S3 storage locations.

· **S3** (*dict*) –

An Amazon S3 storage location.

· **Bucket** (*string*) –

The bucket in which to store the AMI. You can specify a bucket that you already own or a new bucket that Amazon EC2 creates on your behalf. If you specify a bucket that belongs to someone else, Amazon EC2 returns an error.

· **Prefix** (*string*) –

The beginning of the file name of the AMI.

· **AWSAccessKeyId** (*string*) –

The access key ID of the owner of the bucket. Before you specify a value for your access key ID, review and follow the guidance in [Best Practices for Managing AWS Access Keys](#).

· **UploadPolicy** (*bytes*) –

A Base64-encoded Amazon S3 upload policy that gives Amazon EC2 permission to upload items into Amazon S3 on your behalf.

· **UploadPolicySignature** (*string*) –

The signature of the Base64 encoded JSON document.

· **Progress** (*string*) –

The level of task completion, as a percent (for example, 20%).

· **BundleTaskError** (*dict*) –

If the task fails, a description of the error.

· **Code** (*string*) –

The error code.

· **Message** (*string*) –

The error message.

describe_classic_link_instances (***kwargs*)

Describes one or more of your linked EC2-Classic instances. This request only returns information about EC2-Classic instances linked to a VPC through ClassicLink; you cannot use this request to return information about other instances.

Request Syntax

```
response = client.describe_classic_link_instances(
    DryRun=True|False,
    InstanceIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceIds** (*list*) – One or more instance IDs. Must be instances linked to a VPC through ClassicLink.
 - (*string*) –
- **Filters** (*list*) – One or more filters.
 - `group-id` - The ID of a VPC security group that's associated with the instance.
 - `instance-id` - The ID of the instance.
 - `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
 - `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag's value is), and the tag value `X` (regardless of what the tag's key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
 - `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
 - `vpc-id` - The ID of the VPC that the instance is linked to.
 - (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

 - * **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.
 - * **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· *(string)* –

- **NextToken** (*string*) – The token to retrieve the next page of results.
- **MaxResults** (*integer*) – The maximum number of results to return for the request in a single page. The remaining results of the initial request can be seen by sending another request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. You cannot specify this parameter and the `instance IDs` parameter in the same request.

Constraint: If the value is greater than 1000, we return only 1000 items.

Return type dict

Returns

Response Syntax

```
{
  'Instances': [
    {
      'InstanceId': 'string',
      'VpcId': 'string',
      'Groups': [
        {
          'GroupName': 'string',
          'GroupId': 'string'
        },
      ],
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        },
      ],
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

– **Instances** (*list*) –

Information about one or more linked EC2-Classic instances.

* (*dict*) –

Describes a linked EC2-Classic instance.

· **InstanceId** (*string*) –

The ID of the instance.

· **VpcId** (*string*) –

The ID of the VPC.

· **Groups** (*list*) –

A list of security groups.

· (*dict*) –

Describes a security group.

· **GroupName** (*string*) –

- The name of the security group.
- **GroupId** (*string*) –
 - The ID of the security group.
- **Tags** (*list*) –
 - Any tags assigned to the instance.
- (*dict*) –
 - Describes a tag.
- **Key** (*string*) –
 - The key of the tag.
- Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`
- **Value** (*string*) –
 - The value of the tag.
- Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.
- **NextToken** (*string*) –
 - The token to use to retrieve the next page of results. This value is `null` when there are no more results to return.

describe_conversion_tasks (***kwargs*)

Describes one or more of your conversion tasks. For more information, see [Using the Command Line Tools to Import Your Virtual Machine to Amazon EC2](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.describe_conversion_tasks(
    DryRun=True|False,
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    ConversionTaskIds=[
        'string',
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Filters** (*list*) – One or more filters.
 - (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

• **ConversionTaskIds** (*list*) – One or more conversion task IDs.

– (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'ConversionTasks': [
    {
      'ConversionTaskId': 'string',
      'ExpirationTime': 'string',
      'ImportInstance': {
        'Volumes': [
          {
            'BytesConverted': 123,
            'AvailabilityZone': 'string',
            'Image': {
              'Format': 'VMDK'|'RAW'|'VHD',
              'Size': 123,
              'ImportManifestUrl': 'string',
              'Checksum': 'string'
            },
            'Volume': {
              'Size': 123,
              'Id': 'string'
            },
            'Status': 'string',
            'StatusMessage': 'string',
            'Description': 'string'
          },
        ],
        'InstanceId': 'string',
        'Platform': 'Windows',
        'Description': 'string'
      },
      'ImportVolume': {
        'BytesConverted': 123,
        'AvailabilityZone': 'string',
        'Description': 'string',
        'Image': {
          'Format': 'VMDK'|'RAW'|'VHD',
          'Size': 123,
          'ImportManifestUrl': 'string',
          'Checksum': 'string'
        },
        'Volume': {
          'Size': 123,
          'Id': 'string'
        }
      },
      'State': 'active'|'cancelling'|'cancelled'|'completed',
    },
  ],
}
```

```
        'StatusMessage': 'string',
        'Tags': [
            {
                'Key': 'string',
                'Value': 'string'
            },
        ],
    ],
}
```

Response Structure

- (*dict*) –
 - **ConversionTasks** (*list*) –

Information about the conversion tasks.

- * (*dict*) –

Describes a conversion task.

- **ConversionTaskId** (*string*) –

The ID of the conversion task.

- **ExpirationTime** (*string*) –

The time when the task expires. If the upload isn't complete before the expiration time, we automatically cancel the task.

- **ImportInstance** (*dict*) –

If the task is for importing an instance, this contains information about the import instance task.

- **Volumes** (*list*) –

One or more volumes.

- (*dict*) –

Describes an import volume task.

- **BytesConverted** (*integer*) –

The number of bytes converted so far.

- **AvailabilityZone** (*string*) –

The Availability Zone where the resulting instance will reside.

- **Image** (*dict*) –

The image.

- **Format** (*string*) –

The disk image format.

- **Size** (*integer*) –

The size of the disk image, in GiB.

- **ImportManifestUrl** (*string*) –

A presigned URL for the import manifest stored in Amazon S3. For information about creating a presigned URL for an Amazon S3 object, read the “Query String Request Authentication Alternative” section of the [Authenticating REST Requests](#) topic in the *Amazon Simple Storage Service Developer Guide*.

- **Checksum** (*string*) –

- The checksum computed for the disk image.
- **Volume** (*dict*) –
 - The volume.
- **Size** (*integer*) –
 - The size of the volume, in GiB.
- **Id** (*string*) –
 - The volume identifier.
- **Status** (*string*) –
 - The status of the import of this particular disk image.
- **StatusMessage** (*string*) –
 - The status information or errors related to the disk image.
- **Description** (*string*) –
 - A description of the task.
- **InstanceId** (*string*) –
 - The ID of the instance.
- **Platform** (*string*) –
 - The instance operating system.
- **Description** (*string*) –
 - A description of the task.
- **ImportVolume** (*dict*) –
 - If the task is for importing a volume, this contains information about the import volume task.
- **BytesConverted** (*integer*) –
 - The number of bytes converted so far.
- **AvailabilityZone** (*string*) –
 - The Availability Zone where the resulting volume will reside.
- **Description** (*string*) –
 - The description you provided when starting the import volume task.
- **Image** (*dict*) –
 - The image.
- **Format** (*string*) –
 - The disk image format.
- **Size** (*integer*) –
 - The size of the disk image, in GiB.
- **ImportManifestUrl** (*string*) –
 - A presigned URL for the import manifest stored in Amazon S3. For information about creating a presigned URL for an Amazon S3 object, read the “Query String Request Authentication Alternative” section of the [Authenticating REST Requests](#) topic in the *Amazon Simple Storage Service Developer Guide*.
- **Checksum** (*string*) –
 - The checksum computed for the disk image.

- **Volume** (*dict*) –
The volume.
- **Size** (*integer*) –
The size of the volume, in GiB.
- **Id** (*string*) –
The volume identifier.
- **State** (*string*) –
The state of the conversion task.
- **StatusMessage** (*string*) –
The status message related to the conversion task.
- **Tags** (*list*) –
Any tags assigned to the task.
- (*dict*) –
Describes a tag.
- **Key** (*string*) –
The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`
- **Value** (*string*) –
The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

describe_customer_gateways (***kwargs*)

Describes one or more of your VPN customer gateways.

For more information about VPN customer gateways, see [Adding a Hardware Virtual Private Gateway to Your VPC](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.describe_customer_gateways(
    DryRun=True|False,
    CustomerGatewayIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **CustomerGatewayIds** (*list*) – One or more customer gateway IDs.

Default: Describes all your customer gateways.

– (*string*) –

- **Filters** (*list*) – One or more filters.

- `bgp-asn` - The customer gateway’s Border Gateway Protocol (BGP) Autonomous System Number (ASN).
- `customer-gateway-id` - The ID of the customer gateway.
- `ip-address` - The IP address of the customer gateway’s Internet-routable external interface.
- `state` - The state of the customer gateway (`pending` | `available` | `deleting` | `deleted`).
- `type` - The type of customer gateway. Currently, the only supported type is `ipsec.1`.
- `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag’s value is), and the tag value `X` (regardless of what the tag’s key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.

– (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

Return type `dict`

Returns

Response Syntax

```
{
  'CustomerGateways': [
    {
      'CustomerGatewayId': 'string',
      'State': 'string',
      'Type': 'string',
      'IpAddress': 'string',
      'BgpAsn': 'string',
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        },
      ],
    },
  ],
}
```

```
]
}
```

Response Structure

- *(dict)* –
 - **CustomerGateways** (*list*) –

Information about one or more customer gateways.

* *(dict)* –

Describes a customer gateway.

- **CustomerGatewayId** (*string*) –

The ID of the customer gateway.

- **State** (*string*) –

The current state of the customer gateway (pending | available | deleting | deleted).

- **Type** (*string*) –

The type of VPN connection the customer gateway supports (ipsec.1).

- **IpAddress** (*string*) –

The Internet-routable IP address of the customer gateway's outside interface.

- **BgpAsn** (*string*) –

The customer gateway's Border Gateway Protocol (BGP) Autonomous System Number (ASN).

- **Tags** (*list*) –

Any tags assigned to the customer gateway.

- *(dict)* –

Describes a tag.

- **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

- **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

describe_dhcp_options (**kwargs)

Describes one or more of your DHCP options sets.

For more information about DHCP options sets, see [DHCP Options Sets](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.describe_dhcp_options(
    DryRun=True|False,
    DhcpOptionsIds=[
        'string',
    ],
```

```

Filters=[
    {
        'Name': 'string',
        'Values': [
            'string',
        ]
    },
]
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **DhcpOptionsIds** (*list*) – The IDs of one or more DHCP options sets.

Default: Describes all your DHCP options sets.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `dhcp-options-id` - The ID of a set of DHCP options.
 - `key` - The key for one of the options (for example, `domain-name`).
 - `value` - The value for one of the options.
 - `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
 - `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag’s value is), and the tag value `X` (regardless of what the tag’s key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
 - `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.

– (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

Return type dict

Returns

Response Syntax

```

{
    'DhcpOptions': [
        {
            'DhcpOptionsId': 'string',
            'DhcpConfigurations': [
                {
                    'Key': 'string',

```

```
        'Values': [
            {
                'Value': 'string'
            },
        ],
    },
],
'Tags': [
    {
        'Key': 'string',
        'Value': 'string'
    },
],
},
]
```

Response Structure

- (*dict*) –

- **DhcpOptions** (*list*) –

Information about one or more DHCP options sets.

- * (*dict*) –

Describes a set of DHCP options.

- **DhcpOptionsId** (*string*) –

The ID of the set of DHCP options.

- **DhcpConfigurations** (*list*) –

One or more DHCP options in the set.

- (*dict*) –

Describes a DHCP configuration option.

- **Key** (*string*) –

The name of a DHCP option.

- **Values** (*list*) –

One or more values for the DHCP option.

- (*dict*) –

The value to use for a resource attribute.

- **Value** (*string*) –

Valid values are case-sensitive and vary by action.

- **Tags** (*list*) –

Any tags assigned to the DHCP options set.

- (*dict*) –

Describes a tag.

- **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

- **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

describe_export_tasks (**kwargs)

Describes one or more of your export tasks.

Request Syntax

```
response = client.describe_export_tasks(
    ExportTaskIds=[
        'string',
    ]
)
```

Parameters **ExportTaskIds** (*list*) – One or more export task IDs.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'ExportTasks': [
        {
            'ExportTaskId': 'string',
            'Description': 'string',
            'State': 'active'|'cancelling'|'cancelled'|'completed',
            'StatusMessage': 'string',
            'InstanceExportDetails': {
                'InstanceId': 'string',
                'TargetEnvironment': 'citrix'|'vmware'|'microsoft'
            },
            'ExportToS3Task': {
                'DiskImageFormat': 'VMDK'|'RAW'|'VHD',
                'ContainerFormat': 'ova',
                'S3Bucket': 'string',
                'S3Key': 'string'
            }
        },
    ]
}
```

Response Structure

- (*dict*) –

– **ExportTasks** (*list*) –

Information about the export tasks.

- * (*dict*) –

Describes an instance export task.

- **ExportTaskId** (*string*) –

The ID of the export task.

- **Description** (*string*) –

A description of the resource being exported.

- **State** (*string*) –

The state of the export task.

- **StatusMessage** (*string*) –
The status message related to the export task.
- **InstanceExportDetails** (*dict*) –
Information about the instance to export.
- **InstanceId** (*string*) –
The ID of the resource being exported.
- **TargetEnvironment** (*string*) –
The target virtualization environment.
- **ExportToS3Task** (*dict*) –
Information about the export task.
- **DiskImageFormat** (*string*) –
The format for the exported image.
- **ContainerFormat** (*string*) –
The container format used to combine disk images with meta-data (such as OVF). If absent, only the disk image is exported.
- **S3Bucket** (*string*) –
The S3 bucket for the destination image. The destination bucket must exist and grant WRITE and READ_ACP permissions to the AWS account `vm-import-export@amazon.com`.
- **S3Key** (*string*) –
The encryption key for your S3 bucket.

describe_flow_logs (***kwargs*)

Describes one or more flow logs. To view the information in your flow logs (the log streams for the network interfaces), you must use the CloudWatch Logs console or the CloudWatch Logs API.

Request Syntax

```
response = client.describe_flow_logs(  
    FlowLogIds=[  
        'string',  
    ],  
    Filter=[  
        {  
            'Name': 'string',  
            'Values': [  
                'string',  
            ],  
        },  
    ],  
    NextToken='string',  
    MaxResults=123  
)
```

Parameters

- **FlowLogIds** (*list*) – One or more flow log IDs.
 - (*string*) –
- **Filter** (*list*) – One or more filters.
 - `deliver-log-status` - The status of the logs delivery (SUCCESS | FAILED).
 - `flow-log-id` - The ID of the flow log.

- `log-group-name` - The name of the log group.
- `resource-id` - The ID of the VPC, subnet, or network interface.
- `traffic-type` - The type of traffic (ACCEPT | REJECT | ALL)
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

- * **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

- * **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

- (*string*) -

- **NextToken** (*string*) - The token to retrieve the next page of results.
- **MaxResults** (*integer*) - The maximum number of results to return for the request in a single page. The remaining results can be seen by sending another request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. You cannot specify this parameter and the flow log IDs parameter in the same request.

Return type dict

Returns

Response Syntax

```
{
  'FlowLogs': [
    {
      'CreationTime': datetime(2015, 1, 1),
      'FlowLogId': 'string',
      'FlowLogStatus': 'string',
      'ResourceId': 'string',
      'TrafficType': 'ACCEPT'|'REJECT'|'ALL',
      'LogGroupName': 'string',
      'DeliverLogsStatus': 'string',
      'DeliverLogsErrorMessage': 'string',
      'DeliverLogsPermissionArn': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- *(dict)* -
 - **FlowLogs** (*list*) -

Information about the flow logs.

- * *(dict)* -

Describes a flow log.

- **CreationTime** (*datetime*) -

The date and time the flow log was created.

- **FlowLogId** (*string*) -

The flow log ID.

- **FlowLogStatus** (*string*) –
The status of the flow log (ACTIVE).
- **ResourceId** (*string*) –
The ID of the resource on which the flow log was created.
- **TrafficType** (*string*) –
The type of traffic captured for the flow log.
- **LogGroupName** (*string*) –
The name of the flow log group.
- **DeliverLogsStatus** (*string*) –
The status of the logs delivery (SUCCESS | FAILED).
- **DeliverLogErrorMessage** (*string*) –
Information about the error that occurred. Rate limited indicates that CloudWatch logs throttling has been applied for one or more network interfaces. Access error indicates that the IAM role associated with the flow log does not have sufficient permissions to publish to CloudWatch Logs. Unknown error indicates an internal error.
- **DeliverLogsPermissionArn** (*string*) –
The ARN of the IAM role that posts logs to CloudWatch Logs.
- **NextToken** (*string*) –
The token to use to retrieve the next page of results. This value is null when there are no more results to return.

describe_image_attribute (***kwargs*)

Describes the specified attribute of the specified AMI. You can specify only one attribute at a time.

Request Syntax

```
response = client.describe_image_attribute(  
    DryRun=True|False,  
    ImageId='string',  
    Attribute='description'|'kernel'|'ramdisk'|'launchPermission'|'productCodes'|'blockDevi  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **ImageId** (*string*) – [REQUIRED]
The ID of the AMI.
- **Attribute** (*string*) – [REQUIRED]
The AMI attribute.

Note : Depending on your account privileges, the `blockDeviceMapping` attribute may return a `Client.AuthFailure` error. If this happens, use `DescribeImages` to get information about the block device mapping for the AMI.

Return type dict

Returns

Response Syntax

```
{
  'ImageId': 'string',
  'LaunchPermissions': [
    {
      'UserId': 'string',
      'Group': 'all'
    },
  ],
  'ProductCodes': [
    {
      'ProductCodeId': 'string',
      'ProductCodeType': 'devpay'|'marketplace'
    },
  ],
  'KernelId': {
    'Value': 'string'
  },
  'RamdiskId': {
    'Value': 'string'
  },
  'Description': {
    'Value': 'string'
  },
  'SriovNetSupport': {
    'Value': 'string'
  },
  'BlockDeviceMappings': [
    {
      'VirtualName': 'string',
      'DeviceName': 'string',
      'Ebs': {
        'SnapshotId': 'string',
        'VolumeSize': 123,
        'DeleteOnTermination': True|False,
        'VolumeType': 'standard'|"io1"|"gp2',
        'Iops': 123,
        'Encrypted': True|False
      },
      'NoDevice': 'string'
    },
  ],
}
```

Response Structure

- (*dict*) –

Information about the image attribute.

- **ImageId** (*string*) –

The ID of the AMI.

- **LaunchPermissions** (*list*) –

One or more launch permissions.

- * (*dict*) –

Describes a launch permission.

- **UserId** (*string*) –
The AWS account ID.
- **Group** (*string*) –
The name of the group.
- **ProductCodes** (*list*) –
One or more product codes.
 - * (*dict*) –
Describes a product code.
 - **ProductCodeId** (*string*) –
The product code.
 - **ProductCodeType** (*string*) –
The type of product code.
- **KernelId** (*dict*) –
The kernel ID.
 - * **Value** (*string*) –
Valid values are case-sensitive and vary by action.
- **RamdiskId** (*dict*) –
The RAM disk ID.
 - * **Value** (*string*) –
Valid values are case-sensitive and vary by action.
- **Description** (*dict*) –
A description for the AMI.
 - * **Value** (*string*) –
Valid values are case-sensitive and vary by action.
- **SriovNetSupport** (*dict*) –
The value to use for a resource attribute.
 - * **Value** (*string*) –
Valid values are case-sensitive and vary by action.
- **BlockDeviceMappings** (*list*) –
One or more block device mapping entries.
 - * (*dict*) –
Describes a block device mapping.
 - **VirtualName** (*string*) –
The virtual device name (ephemeral N). Instance store volumes are numbered starting from 0. An instance type with 2 available instance store volumes can specify mappings for `ephemeral0` and `ephemeral1`. The number of available instance store volumes depends on the instance type. After you connect to the instance, you must mount the volume.

Constraints: For M3 instances, you must specify instance store volumes in the block device mapping for the instance. When you launch an M3 instance, we ignore any instance store volumes specified in the block device mapping for the AMI.

- **DeviceName** (*string*) –
The device name exposed to the instance (for example, `/dev/sdh` or `xvdh`).
- **Ebs** (*dict*) –
Parameters used to automatically set up EBS volumes when the instance is launched.
- **SnapshotId** (*string*) –
The ID of the snapshot.
- **VolumeSize** (*integer*) –
The size of the volume, in GiB.

Constraints: 1-1024 for standard volumes, 1-16384 for gp2 volumes, and 4-16384 for io1 volumes. If you specify a snapshot, the volume size must be equal to or larger than the snapshot size.

Default: If you're creating the volume from a snapshot and don't specify a volume size, the default is the snapshot size.
- **DeleteOnTermination** (*boolean*) –
Indicates whether the EBS volume is deleted on instance termination.
- **VolumeType** (*string*) –
The volume type. `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, and `standard` for Magnetic volumes.

Default: `standard`
- **Iops** (*integer*) –
The number of I/O operations per second (IOPS) that the volume supports. For Provisioned IOPS (SSD) volumes, this represents the number of IOPS that are provisioned for the volume. For General Purpose (SSD) volumes, this represents the baseline performance of the volume and the rate at which the volume accumulates I/O credits for bursting. For more information on General Purpose (SSD) baseline performance, I/O credits, and bursting, see [Amazon EBS Volume Types](#) in the *Amazon Elastic Compute Cloud User Guide*.

Constraint: Range is 100 to 20000 for Provisioned IOPS (SSD) volumes and 3 to 10000 for General Purpose (SSD) volumes.

Condition: This parameter is required for requests to create `io1` volumes; it is not used in requests to create `standard` or `gp2` volumes.
- **Encrypted** (*boolean*) –
Indicates whether the EBS volume is encrypted. Encrypted Amazon EBS volumes may only be attached to instances that support Amazon EBS encryption.
- **NoDevice** (*string*) –
Suppresses the specified device included in the block device mapping of the AMI.

describe_images (**kwargs)

Describes one or more of the images (AMIs, AKIs, and ARIs) available to you. Images available to you include public images, private images that you own, and private images owned by other AWS accounts but for which you have explicit launch permissions.

Note: Deregistered images are included in the returned results for an unspecified interval after deregistration.

Request Syntax

```
response = client.describe_images(  
    DryRun=True|False,  
    ImageIds=[  
        'string',  
    ],  
    Owners=[  
        'string',  
    ],  
    ExecutableUsers=[  
        'string',  
    ],  
    Filters=[  
        {  
            'Name': 'string',  
            'Values': [  
                'string',  
            ]  
        },  
    ],  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **ImageIds** (*list*) – One or more image IDs.

Default: Describes all images available to you.
 - (*string*) –
- **Owners** (*list*) – Filters the images by the owner. Specify an AWS account ID, `amazon` (owner is Amazon), `aws-marketplace` (owner is AWS Marketplace), `self` (owner is the sender of the request). Omitting this option returns all images for which you have launch permissions, regardless of ownership.
 - (*string*) –
- **ExecutableUsers** (*list*) – Scopes the images by users with explicit launch permissions. Specify an AWS account ID, `self` (the sender of the request), or `all` (public AMIs).
 - (*string*) –
- **Filters** (*list*) – One or more filters.
 - `architecture` - The image architecture (`i386` | `x86_64`).
 - `block-device-mapping.delete-on-termination` - A Boolean value that indicates whether the Amazon EBS volume is deleted on instance termination.
 - `block-device-mapping.device-name` - The device name for the EBS volume (for example, `/dev/sdh`).

- `block-device-mapping.snapshot-id` - The ID of the snapshot used for the EBS volume.
- `block-device-mapping.volume-size` - The volume size of the EBS volume, in GiB.
- `block-device-mapping.volume-type` - The volume type of the EBS volume (`gp2` | `standard` | `io1`).
- `description` - The description of the image (provided during image creation).
- `hypervisor` - The hypervisor type (`ovm` | `xen`).
- `image-id` - The ID of the image.
- `image-type` - The image type (`machine` | `kernel` | `ramdisk`).
- `is-public` - A Boolean that indicates whether the image is public.
- `kernel-id` - The kernel ID.
- `manifest-location` - The location of the image manifest.
- `name` - The name of the AMI (provided during image creation).
- `owner-alias` - The AWS account alias (for example, `amazon`).
- `owner-id` - The AWS account ID of the image owner.
- `platform` - The platform. To only list Windows-based AMIs, use `windows`.
- `product-code` - The product code.
- `product-code.type` - The type of the product code (`devpay` | `marketplace`).
- `ramdisk-id` - The RAM disk ID.
- `root-device-name` - The name of the root device volume (for example, `/dev/sda1`).
- `root-device-type` - The type of the root device volume (`ebs` | `instance-store`).
- `state` - The state of the image (`available` | `pending` | `failed`).
- `state-reason-code` - The reason code for the state change.
- `state-reason-message` - The message for the state change.
- `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the tag-value filter. For example, if you use both the filter “tag-key=Purpose” and the filter “tag-value=X”, you get any resources assigned both the tag key Purpose (regardless of what the tag’s value is), and the tag value X (regardless of what the tag’s key is). If you want to list only resources where Purpose is X, see the `tag :key =*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the tag-key filter.
- `virtualization-type` - The virtualization type (`paravirtual` | `hvm`).
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

Return type dict

Returns

Response Syntax

```
{
  'Images': [
    {
      'ImageId': 'string',
      'ImageLocation': 'string',
      'State': 'pending'|'available'|'invalid'|'deregistered'|'transient'|'f
      'OwnerId': 'string',
      'CreationDate': 'string',
      'Public': True|False,
      'ProductCodes': [
        {
          'ProductCodeId': 'string',
          'ProductCodeType': 'devpay'|'marketplace'
        },
      ],
      'Architecture': 'i386'|'x86_64',
      'ImageType': 'machine'|'kernel'|'ramdisk',
      'KernelId': 'string',
      'RamdiskId': 'string',
      'Platform': 'Windows',
      'SriovNetSupport': 'string',
      'StateReason': {
        'Code': 'string',
        'Message': 'string'
      },
      'ImageOwnerAlias': 'string',
      'Name': 'string',
      'Description': 'string',
      'RootDeviceType': 'ebs'|'instance-store',
      'RootDeviceName': 'string',
      'BlockDeviceMappings': [
        {
          'VirtualName': 'string',
          'DeviceName': 'string',
          'Ebs': {
            'SnapshotId': 'string',
            'VolumeSize': 123,
            'DeleteOnTermination': True|False,
            'VolumeType': 'standard'|'io1'|'gp2',
            'Iops': 123,
            'Encrypted': True|False
          },
          'NoDevice': 'string'
        },
      ],
      'VirtualizationType': 'hvm'|'paravirtual',
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        },
      ],
      'Hypervisor': 'ovm'|'xen'
    },
  ],
}
```

```
]
}
```

Response Structure

- *(dict)* –

- **Images** (*list*) –

Information about one or more images.

- * *(dict)* –

Describes an image.

- **ImageId** (*string*) –

The ID of the AMI.

- **ImageLocation** (*string*) –

The location of the AMI.

- **State** (*string*) –

The current state of the AMI. If the state is `available`, the image is successfully registered and can be used to launch an instance.

- **OwnerId** (*string*) –

The AWS account ID of the image owner.

- **CreationDate** (*string*) –

The date and time the image was created.

- **Public** (*boolean*) –

Indicates whether the image has public launch permissions. The value is `true` if this image has public launch permissions or `false` if it has only implicit and explicit launch permissions.

- **ProductCodes** (*list*) –

Any product codes associated with the AMI.

- *(dict)* –

Describes a product code.

- **ProductCodeId** (*string*) –

The product code.

- **ProductCodeType** (*string*) –

The type of product code.

- **Architecture** (*string*) –

The architecture of the image.

- **ImageType** (*string*) –

The type of image.

- **KernelId** (*string*) –

The kernel associated with the image, if any. Only applicable for machine images.

- **RamdiskId** (*string*) –

The RAM disk associated with the image, if any. Only applicable for machine images.

- **Platform** (*string*) –
The value is `Windows` for Windows AMIs; otherwise blank.
- **SriovNetSupport** (*string*) –
Specifies whether enhanced networking is enabled.
- **StateReason** (*dict*) –
The reason for the state change.
- **Code** (*string*) –
The reason code for the state change.
- **Message** (*string*) –
The message for the state change.
- `Server.SpotInstanceTermination` : A Spot Instance was terminated due to an increase in the market price.
- `Server.InternalError` : An internal error occurred during instance launch, resulting in termination.
- `Server.InsufficientInstanceCapacity` : There was insufficient instance capacity to satisfy the launch request.
- `Client.InternalError` : A client error caused the instance to terminate on launch.
- `Client.InstanceInitiatedShutdown` : The instance was shut down using the shutdown `-h` command from the instance.
- `Client.UserInitiatedShutdown` : The instance was shut down using the Amazon EC2 API.
- `Client.VolumeLimitExceeded` : The volume limit was exceeded.
- `Client.InvalidSnapshot.NotFound` : The specified snapshot was not found.
- **ImageOwnerAlias** (*string*) –
The AWS account alias (for example, `amazon`, `self`) or the AWS account ID of the AMI owner.
- **Name** (*string*) –
The name of the AMI that was provided during image creation.
- **Description** (*string*) –
The description of the AMI that was provided during image creation.
- **RootDeviceType** (*string*) –
The type of root device used by the AMI. The AMI can use an EBS volume or an instance store volume.
- **RootDeviceName** (*string*) –
The device name of the root device (for example, `/dev/sda1` or `/dev/xvda`).
- **BlockDeviceMappings** (*list*) –
Any block device mapping entries.
- (*dict*) –
Describes a block device mapping.

- **VirtualName** (*string*) –

The virtual device name (`ephemeralN`). Instance store volumes are numbered starting from 0. An instance type with 2 available instance store volumes can specify mappings for `ephemeral0` and `ephemeral1`. The number of available instance store volumes depends on the instance type. After you connect to the instance, you must mount the volume.

Constraints: For M3 instances, you must specify instance store volumes in the block device mapping for the instance. When you launch an M3 instance, we ignore any instance store volumes specified in the block device mapping for the AMI.

- **DeviceName** (*string*) –

The device name exposed to the instance (for example, `/dev/sdh` or `xvdh`).

- **Ebs** (*dict*) –

Parameters used to automatically set up EBS volumes when the instance is launched.

- **SnapshotId** (*string*) –

The ID of the snapshot.

- **VolumeSize** (*integer*) –

The size of the volume, in GiB.

Constraints: 1-1024 for standard volumes, 1-16384 for `gp2` volumes, and 4-16384 for `io1` volumes. If you specify a snapshot, the volume size must be equal to or larger than the snapshot size.

Default: If you're creating the volume from a snapshot and don't specify a volume size, the default is the snapshot size.

- **DeleteOnTermination** (*boolean*) –

Indicates whether the EBS volume is deleted on instance termination.

- **VolumeType** (*string*) –

The volume type. `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, and `standard` for Magnetic volumes.

Default: `standard`

- **Iops** (*integer*) –

The number of I/O operations per second (IOPS) that the volume supports. For Provisioned IOPS (SSD) volumes, this represents the number of IOPS that are provisioned for the volume. For General Purpose (SSD) volumes, this represents the baseline performance of the volume and the rate at which the volume accumulates I/O credits for bursting. For more information on General Purpose (SSD) baseline performance, I/O credits, and bursting, see [Amazon EBS Volume Types](#) in the *Amazon Elastic Compute Cloud User Guide*.

Constraint: Range is 100 to 20000 for Provisioned IOPS

(SSD) volumes and 3 to 10000 for General Purpose (SSD) volumes.

Condition: This parameter is required for requests to create `io1` volumes; it is not used in requests to create `standard` or `gp2` volumes.

- **Encrypted** (*boolean*) –

Indicates whether the EBS volume is encrypted. Encrypted Amazon EBS volumes may only be attached to instances that support Amazon EBS encryption.

- **NoDevice** (*string*) –

Suppresses the specified device included in the block device mapping of the AMI.

- **VirtualizationType** (*string*) –

The type of virtualization of the AMI.

- **Tags** (*list*) –

Any tags assigned to the image.

- (*dict*) –

Describes a tag.

- **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

- **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

- **Hypervisor** (*string*) –

The hypervisor type of the image.

describe_import_image_tasks (***kwargs*)

Displays details about an import virtual machine or import snapshot tasks that are already created.

Request Syntax

```
response = client.describe_import_image_tasks(
    DryRun=True|False,
    ImportTaskIds=[
        'string',
    ],
    NextToken='string',
    MaxResults=123,
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **ImportTaskIds** (*list*) – A list of import image task IDs.
 - (*string*) –
- **NextToken** (*string*) – A token that indicates the next page of results.
- **MaxResults** (*integer*) – The maximum number of results to return in a single request.
- **Filters** (*list*) – One or more filters.
 - (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

Return type dict

Returns**Response Syntax**

```
{
  'ImportImageTasks': [
    {
      'ImportTaskId': 'string',
      'Architecture': 'string',
      'LicenseType': 'string',
      'Platform': 'string',
      'Hypervisor': 'string',
      'Description': 'string',
      'SnapshotDetails': [
        {
          'DiskImageSize': 123.0,
          'Description': 'string',
          'Format': 'string',
          'Url': 'string',
          'UserBucket': {
            'S3Bucket': 'string',
            'S3Key': 'string'
          },
          'DeviceName': 'string',
          'SnapshotId': 'string',
          'Progress': 'string',
          'StatusMessage': 'string',
          'Status': 'string'
        },
      ],
      'ImageId': 'string',
      'Progress': 'string',
      'StatusMessage': 'string',
      'Status': 'string'
    },
  ],
}
```

```
    },  
    ],  
    'NextToken': 'string'  
}
```

Response Structure

- (*dict*) –
 - **ImportImageTasks** (*list*) –

A list of zero or more import image tasks that are currently active or were completed or canceled in the previous 7 days.

- * (*dict*) –

Describes an import image task.

- **ImportTaskId** (*string*) –

The ID of the import image task.

- **Architecture** (*string*) –

The architecture of the virtual machine.

Valid values: `i386` | `x86_64`

- **LicenseType** (*string*) –

The license type of the virtual machine.

- **Platform** (*string*) –

The description string for the import image task.

- **Hypervisor** (*string*) –

The target hypervisor for the import task.

Valid values: `xen`

- **Description** (*string*) –

A description of the import task.

- **SnapshotDetails** (*list*) –

Information about the snapshots.

- (*dict*) –

Describes the snapshot created from the imported disk.

- **DiskImageSize** (*float*) –

The size of the disk in the snapshot, in GiB.

- **Description** (*string*) –

A description for the snapshot.

- **Format** (*string*) –

The format of the disk image from which the snapshot is created.

- **Url** (*string*) –

The URL used to access the disk image.

- **UserBucket** (*dict*) –

Describes the S3 bucket for the disk image.

- **S3Bucket** (*string*) –

The S3 bucket from which the disk image was created.

- **S3Key** (*string*) –
The key from which the disk image was created.
- **DeviceName** (*string*) –
The block device mapping for the snapshot.
- **SnapshotId** (*string*) –
The snapshot ID of the disk being imported.
- **Progress** (*string*) –
The percentage of progress for the task.
- **StatusMessage** (*string*) –
A detailed status message for the snapshot creation.
- **Status** (*string*) –
A brief status of the snapshot creation.
- **ImageId** (*string*) –
The ID of the Amazon Machine Image (AMI) of the imported virtual machine.
- **Progress** (*string*) –
The percentage of progress of the import image task.
- **StatusMessage** (*string*) –
A descriptive status message for the import image task.
- **Status** (*string*) –
A brief status for the import image task.
- **NextToken** (*string*) –
The token to use to get the next page of results. This value is `null` when there are no more results to return.

describe_import_snapshot_tasks (***kwargs*)

Describes your import snapshot tasks.

Request Syntax

```
response = client.describe_import_snapshot_tasks(
    DryRun=True|False,
    ImportTaskIds=[
        'string',
    ],
    NextToken='string',
    MaxResults=123,
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If

you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **ImportTaskIds** (*list*) – A list of import snapshot task IDs.
 - (*string*) –
- **NextToken** (*string*) – A token that indicates the next page of results.
- **MaxResults** (*integer*) – The maximum number of results to return in a single request.
- **Filters** (*list*) – One or more filters.
 - (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'ImportSnapshotTasks': [
    {
      'ImportTaskId': 'string',
      'SnapshotTaskDetail': {
        'DiskImageSize': 123.0,
        'Description': 'string',
        'Format': 'string',
        'Url': 'string',
        'UserBucket': {
          'S3Bucket': 'string',
          'S3Key': 'string'
        },
        'SnapshotId': 'string',
        'Progress': 'string',
        'StatusMessage': 'string',
        'Status': 'string'
      },
      'Description': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **ImportSnapshotTasks** (*list*) –

A list of zero or more import snapshot tasks that are currently active or were completed or canceled in the previous 7 days.

* (*dict*) –

Describes an import snapshot task.

- **ImportTaskId** (*string*) –
The ID of the import snapshot task.
- **SnapshotTaskDetail** (*dict*) –
Describes an import snapshot task.
- **DiskImageSize** (*float*) –
The size of the disk in the snapshot, in GiB.
- **Description** (*string*) –
The description of the snapshot.
- **Format** (*string*) –
The format of the disk image from which the snapshot is created.
- **Url** (*string*) –
The URL of the disk image from which the snapshot is created.
- **UserBucket** (*dict*) –
The S3 bucket for the disk image.
- **S3Bucket** (*string*) –
The S3 bucket from which the disk image was created.
- **S3Key** (*string*) –
The key from which the disk image was created.
- **SnapshotId** (*string*) –
The snapshot ID of the disk being imported.
- **Progress** (*string*) –
The percentage of completion for the import snapshot task.
- **StatusMessage** (*string*) –
A detailed status message for the import snapshot task.
- **Status** (*string*) –
A brief status for the import snapshot task.
- **Description** (*string*) –
A description of the import snapshot task.
- **NextToken** (*string*) –
The token to use to get the next page of results. This value is `null` when there are no more results to return.

describe_instance_attribute (***kwargs*)

Describes the specified attribute of the specified instance. You can specify only one attribute at a time. Valid attribute values are: `instanceType` | `kernel` | `ramdisk` | `userData` | `disableApiTermination` | `instanceInitiatedShutdownBehavior` | `rootDeviceName` | `blockDeviceMapping` | `productCodes` | `sourceDestCheck` | `groupSet` | `efsOptimized` | `sriovNetSupport`

Request Syntax

```
response = client.describe_instance_attribute(
    DryRun=True|False,
    InstanceId='string',
```

```
Attribute='instanceType'|'kernel'|'ramdisk'|'userData'|'disableApiTermination'|'instanc  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceId** (*string*) – **[REQUIRED]**
The ID of the instance.
- **Attribute** (*string*) – **[REQUIRED]**
The instance attribute.

Return type dict

Returns

Response Syntax

```
{  
  'InstanceId': 'string',  
  'InstanceType': {  
    'Value': 'string'  
  },  
  'KernelId': {  
    'Value': 'string'  
  },  
  'RamdiskId': {  
    'Value': 'string'  
  },  
  'UserData': {  
    'Value': 'string'  
  },  
  'DisableApiTermination': {  
    'Value': True|False  
  },  
  'InstanceInitiatedShutdownBehavior': {  
    'Value': 'string'  
  },  
  'RootDeviceName': {  
    'Value': 'string'  
  },  
  'BlockDeviceMappings': [  
    {  
      'DeviceName': 'string',  
      'Ebs': {  
        'VolumeId': 'string',  
        'Status': 'attaching'|'attached'|'detaching'|'detached',  
        'AttachTime': datetime(2015, 1, 1),  
        'DeleteOnTermination': True|False  
      }  
    },  
  ],  
  'ProductCodes': [  
    {  
      'ProductCodeId': 'string',  
      'ProductCodeType': 'devpay'|'marketplace'  
    },  
  ],  
}
```



```

],
'EbsOptimized': {
    'Value': True|False
},
'SriovNetSupport': {
    'Value': 'string'
},
'SourceDestCheck': {
    'Value': True|False
},
'Groups': [
    {
        'GroupName': 'string',
        'GroupId': 'string'
    },
]
}

```

Response Structure

- *(dict)* –

Information about the instance attribute.

- **InstanceId** (*string*) –

The ID of the instance.

- **InstanceType** (*dict*) –

The instance type.

- * **Value** (*string*) –

Valid values are case-sensitive and vary by action.

- **KernelId** (*dict*) –

The kernel ID.

- * **Value** (*string*) –

Valid values are case-sensitive and vary by action.

- **RamdiskId** (*dict*) –

The RAM disk ID.

- * **Value** (*string*) –

Valid values are case-sensitive and vary by action.

- **UserData** (*dict*) –

The Base64-encoded MIME user data.

- * **Value** (*string*) –

Valid values are case-sensitive and vary by action.

- **DisableApiTermination** (*dict*) –

If the value is `true`, you can't terminate the instance through the Amazon EC2 console, CLI, or API; otherwise, you can.

- * **Value** (*boolean*) –

Valid values are `true` or `false`.

- **InstanceInitiatedShutdownBehavior** (*dict*) –

Indicates whether an instance stops or terminates when you initiate shutdown from the instance (using the operating system command for system shutdown).

- * **Value** (*string*) –

Valid values are case-sensitive and vary by action.

- **RootDeviceName** (*dict*) –

The name of the root device (for example, `/dev/sda1` or `/dev/xvda`).

- * **Value** (*string*) –

Valid values are case-sensitive and vary by action.

- **BlockDeviceMappings** (*list*) –

The block device mapping of the instance.

- * (*dict*) –

Describes a block device mapping.

- **DeviceName** (*string*) –

The device name exposed to the instance (for example, `/dev/sdh` or `xvdh`).

- **Ebs** (*dict*) –

Parameters used to automatically set up EBS volumes when the instance is launched.

- **VolumeId** (*string*) –

The ID of the EBS volume.

- **Status** (*string*) –

The attachment state.

- **AttachTime** (*datetime*) –

The time stamp when the attachment initiated.

- **DeleteOnTermination** (*boolean*) –

Indicates whether the volume is deleted on instance termination.

- **ProductCodes** (*list*) –

A list of product codes.

- * (*dict*) –

Describes a product code.

- **ProductCodeId** (*string*) –

The product code.

- **ProductCodeType** (*string*) –

The type of product code.

- **EbsOptimized** (*dict*) –

Indicates whether the instance is optimized for EBS I/O.

- * **Value** (*boolean*) –

Valid values are `true` or `false`.

- **SriovNetSupport** (*dict*) –

The value to use for a resource attribute.

- * **Value** (*string*) –

Valid values are case-sensitive and vary by action.

– **SourceDestCheck** (*dict*) –

Indicates whether source/destination checking is enabled. A value of `true` means checking is enabled, and `false` means checking is disabled. This value must be `false` for a NAT instance to perform NAT.

* **Value** (*boolean*) –

Valid values are `true` or `false`.

– **Groups** (*list*) –

The security groups associated with the instance.

* (*dict*) –

Describes a security group.

· **GroupName** (*string*) –

The name of the security group.

· **GroupId** (*string*) –

The ID of the security group.

describe_instance_status (***kwargs*)

Describes the status of one or more instances.

Instance status includes the following components:

• **Status checks** - Amazon EC2 performs status checks on running EC2 instances to identify hardware and software issues. For more information, see [Status Checks for Your Instances](#) and [Troubleshooting Instances with Failed Status Checks](#) in the *Amazon Elastic Compute Cloud User Guide*.

• **Scheduled events** - Amazon EC2 can schedule events (such as reboot, stop, or terminate) for your instances related to hardware issues, software updates, or system maintenance. For more information, see [Scheduled Events for Your Instances](#) in the *Amazon Elastic Compute Cloud User Guide*.

• **Instance state** - You can manage your instances from the moment you launch them through their termination. For more information, see [Instance Lifecycle](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.describe_instance_status(
    DryRun=True|False,
    InstanceIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    NextToken='string',
    MaxResults=123,
    IncludeAllInstances=True|False
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If

you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **InstanceIds** (*list*) – One or more instance IDs.

Default: Describes all your instances.

Constraints: Maximum 100 explicitly specified instance IDs.

– (*string*) –

- **Filters** (*list*) – One or more filters.

- `availability-zone` - The Availability Zone of the instance.
- `event.code` - The code for the scheduled event (`instance-reboot` | `system-reboot` | `system-maintenance` | `instance-retirement` | `instance-stop`).
- `event.description` - A description of the event.
- `event.not-after` - The latest end time for the scheduled event (for example, `2014-09-15T17:15:20.000Z`).
- `event.not-before` - The earliest start time for the scheduled event (for example, `2014-09-15T17:15:20.000Z`).
- `instance-state-code` - The code for the instance state, as a 16-bit unsigned integer. The high byte is an opaque internal value and should be ignored. The low byte is set based on the state represented. The valid values are 0 (pending), 16 (running), 32 (shutting-down), 48 (terminated), 64 (stopping), and 80 (stopped).
- `instance-state-name` - The state of the instance (`pending` | `running` | `shutting-down` | `terminated` | `stopping` | `stopped`).
- `instance-status.reachability` - Filters on instance status where the name is `reachability` (`passed` | `failed` | `initializing` | `insufficient-data`).
- `instance-status.status` - The status of the instance (`ok` | `impaired` | `initializing` | `insufficient-data` | `not-applicable`).
- `system-status.reachability` - Filters on system status where the name is `reachability` (`passed` | `failed` | `initializing` | `insufficient-data`).
- `system-status.status` - The system status of the instance (`ok` | `impaired` | `initializing` | `insufficient-data` | `not-applicable`).
- (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

- **NextToken** (*string*) – The token to retrieve the next page of results.
- **MaxResults** (*integer*) – The maximum number of results to return for the request in a single page. The remaining results of the initial request can be seen by sending another request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. You cannot specify this parameter and the instance

IDs parameter in the same request.

- **IncludeAllInstances** (*boolean*) – When `true`, includes the health status for all instances. When `false`, includes the health status for running instances only.

Default: `false`

Return type dict

Returns

Response Syntax

```
{
  'InstanceStatuses': [
    {
      'InstanceId': 'string',
      'AvailabilityZone': 'string',
      'Events': [
        {
          'Code': 'instance-reboot'|'system-reboot'|'system-maintenance',
          'Description': 'string',
          'NotBefore': datetime(2015, 1, 1),
          'NotAfter': datetime(2015, 1, 1)
        },
      ],
      'InstanceState': {
        'Code': 123,
        'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stopping',
      },
      'SystemStatus': {
        'Status': 'ok'|'impaired'|'insufficient-data'|'not-applicable'|'in-progress',
        'Details': [
          {
            'Name': 'reachability',
            'Status': 'passed'|'failed'|'insufficient-data'|'initializing',
            'ImpairedSince': datetime(2015, 1, 1)
          },
        ],
      },
    },
    {
      'InstanceId': 'string',
      'AvailabilityZone': 'string',
      'Events': [
        {
          'Code': 'instance-reboot'|'system-reboot'|'system-maintenance',
          'Description': 'string',
          'NotBefore': datetime(2015, 1, 1),
          'NotAfter': datetime(2015, 1, 1)
        },
      ],
      'InstanceState': {
        'Code': 123,
        'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stopping',
      },
      'SystemStatus': {
        'Status': 'ok'|'impaired'|'insufficient-data'|'not-applicable'|'in-progress',
        'Details': [
          {
            'Name': 'reachability',
            'Status': 'passed'|'failed'|'insufficient-data'|'initializing',
            'ImpairedSince': datetime(2015, 1, 1)
          },
        ],
      },
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **InstanceStatuses** (*list*) –
 - One or more instance status descriptions.
 - * (*dict*) –

Describes the status of an instance.

- **InstanceId** (*string*) –
The ID of the instance.
- **AvailabilityZone** (*string*) –
The Availability Zone of the instance.
- **Events** (*list*) –
Any scheduled events associated with the instance.
- (*dict*) –
Describes a scheduled event for an instance.
- **Code** (*string*) –
The event code.
- **Description** (*string*) –
A description of the event.
After a scheduled event is completed, it can still be described for up to a week. If the event has been completed, this description starts with the following text: [Completed].
- **NotBefore** (*datetime*) –
The earliest scheduled start time for the event.
- **NotAfter** (*datetime*) –
The latest scheduled end time for the event.
- **InstanceState** (*dict*) –
The intended state of the instance. DescribeInstanceState requires that an instance be in the `running` state.
- **Code** (*integer*) –
The low byte represents the state. The high byte is an opaque internal value and should be ignored.
 - 0: pending
 - 16: running
 - 32: shutting-down
 - 48: terminated
 - 64: stopping
 - 80: stopped
- **Name** (*string*) –
The current state of the instance.
- **SystemStatus** (*dict*) –
Reports impaired functionality that stems from issues related to the systems that support an instance, such as hardware failures and network connectivity problems.
- **Status** (*string*) –
The status.
- **Details** (*list*) –
The system instance health or application instance health.
- (*dict*) –
Describes the instance status.
- **Name** (*string*) –

The type of instance status.

- **Status** (*string*) –

The status.

- **ImpairedSince** (*datetime*) –

The time when a status check failed. For an instance that was launched and impaired, this is the time when the instance was launched.

- **InstanceStatus** (*dict*) –

Reports impaired functionality that stems from issues internal to the instance, such as impaired reachability.

- **Status** (*string*) –

The status.

- **Details** (*list*) –

The system instance health or application instance health.

- (*dict*) –

Describes the instance status.

- **Name** (*string*) –

The type of instance status.

- **Status** (*string*) –

The status.

- **ImpairedSince** (*datetime*) –

The time when a status check failed. For an instance that was launched and impaired, this is the time when the instance was launched.

- **NextToken** (*string*) –

The token to use to retrieve the next page of results. This value is `null` when there are no more results to return.

describe_instances (***kwargs*)

Describes one or more of your instances.

If you specify one or more instance IDs, Amazon EC2 returns information for those instances. If you do not specify instance IDs, Amazon EC2 returns information for all relevant instances. If you specify an instance ID that is not valid, an error is returned. If you specify an instance that you do not own, it is not included in the returned results.

Recently terminated instances might appear in the returned results. This interval is usually less than one hour.

Request Syntax

```
response = client.describe_instances(
    DryRun=True|False,
    InstanceIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
```

```
    ],
    },
],
NextToken='string',
MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceIds** (*list*) – One or more instance IDs.

Default: Describes all your instances.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `architecture` - The instance architecture (`i386` | `x86_64`).
 - `availability-zone` - The Availability Zone of the instance.
 - `block-device-mapping.attach-time` - The attach time for an EBS volume mapped to the instance, for example, `2010-09-15T17:15:20.000Z`.
 - `block-device-mapping.delete-on-termination` - A Boolean that indicates whether the EBS volume is deleted on instance termination.
 - `block-device-mapping.device-name` - The device name for the EBS volume (for example, `/dev/sdh` or `xvdh`).
 - `block-device-mapping.status` - The status for the EBS volume (`attaching` | `attached` | `detaching` | `detached`).
 - `block-device-mapping.volume-id` - The volume ID of the EBS volume.
 - `client-token` - The idempotency token you provided when you launched the instance.
 - `dns-name` - The public DNS name of the instance.
 - `group-id` - The ID of the security group for the instance. EC2-Classic only.
 - `group-name` - The name of the security group for the instance. EC2-Classic only.
 - `hypervisor` - The hypervisor type of the instance (`ovm` | `xen`).
 - `iam-instance-profile.arn` - The instance profile associated with the instance. Specified as an ARN.
 - `image-id` - The ID of the image used to launch the instance.
 - `instance-id` - The ID of the instance.
 - `instance-lifecycle` - Indicates whether this is a Spot Instance (`spot`).
 - `instance-state-code` - The state of the instance, as a 16-bit unsigned integer. The high byte is an opaque internal value and should be ignored. The low byte is set based on the state represented. The valid values are: 0 (pending), 16 (running), 32 (shutting-down), 48 (terminated), 64 (stopping), and 80 (stopped).
 - `instance-state-name` - The state of the instance (`pending` | `running` | `shutting-down` | `terminated` | `stopping` | `stopped`).
 - `instance-type` - The type of instance (for example, `t2.micro`).
 - `instance.group-id` - The ID of the security group for the instance.

- `instance.group-name` - The name of the security group for the instance.
- `ip-address` - The public IP address of the instance.
- `kernel-id` - The kernel ID.
- `key-name` - The name of the key pair used when the instance was launched.
- `launch-index` - When launching multiple instances, this is the index for the instance in the launch group (for example, 0, 1, 2, and so on).
- `launch-time` - The time when the instance was launched.
- `monitoring-state` - Indicates whether monitoring is enabled for the instance (`disabled` | `enabled`).
- `owner-id` - The AWS account ID of the instance owner.
- `placement-group-name` - The name of the placement group for the instance.
- `platform` - The platform. Use windows if you have Windows instances; otherwise, leave blank.
- `private-dns-name` - The private DNS name of the instance.
- `private-ip-address` - The private IP address of the instance.
- `product-code` - The product code associated with the AMI used to launch the instance.
- `product-code.type` - The type of product code (`devpay` | `marketplace`).
- `ramdisk-id` - The RAM disk ID.
- `reason` - The reason for the current state of the instance (for example, shows “User Initiated [date]” when you stop or terminate the instance). Similar to the `state-reason-code` filter.
- `requester-id` - The ID of the entity that launched the instance on your behalf (for example, AWS Management Console, Auto Scaling, and so on).
- `reservation-id` - The ID of the instance’s reservation. A reservation ID is created any time you launch an instance. A reservation ID has a one-to-one relationship with an instance launch request, but can be associated with more than one instance if you launch multiple instances using the same launch request. For example, if you launch one instance, you’ll get one reservation ID. If you launch ten instances using the same launch request, you’ll also get one reservation ID.
- `root-device-name` - The name of the root device for the instance (for example, `/dev/sda1` or `/dev/xvda`).
- `root-device-type` - The type of root device that the instance uses (`ebs` | `instance-store`).
- `source-dest-check` - Indicates whether the instance performs source/destination checking. A value of `true` means that checking is enabled, and `false` means checking is disabled. The value must be `false` for the instance to perform network address translation (NAT) in your VPC.
- `spot-instance-request-id` - The ID of the Spot Instance request.
- `state-reason-code` - The reason code for the state change.
- `state-reason-message` - A message that describes the state change.
- `subnet-id` - The ID of the subnet for the instance.
- `tag :key =*value*` - The key/value combination of a tag assigned to the resource, where `tag :key` is the tag’s key.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources as-

signed both the tag key Purpose (regardless of what the tag's value is), and the tag value X (regardless of what the tag's key is). If you want to list only resources where Purpose is X, see the `tag:key=*value*` filter.

- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `tenancy` - The tenancy of an instance (`dedicated`|`default`).
- `virtualization-type` - The virtualization type of the instance (`paravirtual`|`hvm`).
- `vpc-id` - The ID of the VPC that the instance is running in.
- `network-interface.description` - The description of the network interface.
- `network-interface.subnet-id` - The ID of the subnet for the network interface.
- `network-interface.vpc-id` - The ID of the VPC for the network interface.
- `network-interface.network-interface.id` - The ID of the network interface.
- `network-interface.owner-id` - The ID of the owner of the network interface.
- `network-interface.availability-zone` - The Availability Zone for the network interface.
- `network-interface.requester-id` - The requester ID for the network interface.
- `network-interface.requester-managed` - Indicates whether the network interface is being managed by AWS.
- `network-interface.status` - The status of the network interface (`available`)|`in-use`).
- `network-interface.mac-address` - The MAC address of the network interface.
- `network-interface.private-dns-name` - The private DNS name of the network interface.
- `network-interface.source-dest-check` - Whether the network interface performs source/destination checking. A value of `true` means checking is enabled, and `false` means checking is disabled. The value must be `false` for the network interface to perform network address translation (NAT) in your VPC.
- `network-interface.group-id` - The ID of a security group associated with the network interface.
- `network-interface.group-name` - The name of a security group associated with the network interface.
- `network-interface.attachment.attachment-id` - The ID of the interface attachment.
- `network-interface.attachment.instance-id` - The ID of the instance to which the network interface is attached.
- `network-interface.attachment.instance-owner-id` - The owner ID of the instance to which the network interface is attached.
- `network-interface.addresses.private-ip-address` - The private IP address associated with the network interface.
- `network-interface.attachment.device-index` - The device index to which the network interface is attached.
- `network-interface.attachment.status` - The status of the attachment (`attaching`|`attached`|`detaching`|`detached`).
- `network-interface.attachment.attach-time` - The time that the network interface was attached to an instance.

- `network-interface.attachment.delete-on-termination` - Specifies whether the attachment is deleted when an instance is terminated.
- `network-interface.addresses.primary` - Specifies whether the IP address of the network interface is the primary private IP address.
- `network-interface.addresses.association.public-ip` - The ID of the association of an Elastic IP address with a network interface.
- `network-interface.addresses.association.ip-owner-id` - The owner ID of the private IP address associated with the network interface.
- `association.public-ip` - The address of the Elastic IP address bound to the network interface.
- `association.ip-owner-id` - The owner of the Elastic IP address associated with the network interface.
- `association.allocation-id` - The allocation ID returned when you allocated the Elastic IP address for your network interface.
- `association.association-id` - The association ID returned when the network interface was associated with an IP address.

- (*dict*) -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

- **NextToken** (*string*) - The token to request the next page of results.
- **MaxResults** (*integer*) - The maximum number of results to return for the request in a single page. The remaining results of the initial request can be seen by sending another request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. You cannot specify this parameter and the `instance IDs` parameter in the same request.

Return type `dict`

Returns

Response Syntax

```
{
  'Reservations': [
    {
      'ReservationId': 'string',
      'OwnerId': 'string',
      'RequesterId': 'string',
      'Groups': [
        {
          'GroupName': 'string',
          'GroupId': 'string'
        }
      ],
    },
  ],
  'Instances': [
```

```
{
    'InstanceId': 'string',
    'ImageId': 'string',
    'State': {
        'Code': 123,
        'Name': 'pending'|'running'|'shutting-down'|'terminated'|'
    },
    'PrivateDnsName': 'string',
    'PublicDnsName': 'string',
    'StateTransitionReason': 'string',
    'KeyName': 'string',
    'AmiLaunchIndex': 123,
    'ProductCodes': [
        {
            'ProductCodeId': 'string',
            'ProductCodeType': 'devpay'|'marketplace'
        },
    ],
    'InstanceType': 't1.micro'|'m1.small'|'m1.medium'|'m1.large'|'
    'LaunchTime': datetime(2015, 1, 1),
    'Placement': {
        'AvailabilityZone': 'string',
        'GroupName': 'string',
        'Tenancy': 'default'|'dedicated'
    },
    'KernelId': 'string',
    'RamdiskId': 'string',
    'Platform': 'Windows',
    'Monitoring': {
        'State': 'disabled'|'disabling'|'enabled'|'pending'
    },
    'SubnetId': 'string',
    'VpcId': 'string',
    'PrivateIpAddress': 'string',
    'PublicIpAddress': 'string',
    'StateReason': {
        'Code': 'string',
        'Message': 'string'
    },
    'Architecture': 'i386'|'x86_64',
    'RootDeviceType': 'ebs'|'instance-store',
    'RootDeviceName': 'string',
    'BlockDeviceMappings': [
        {
            'DeviceName': 'string',
            'Ebs': {
                'VolumeId': 'string',
                'Status': 'attaching'|'attached'|'detaching'|'deta
                'AttachTime': datetime(2015, 1, 1),
                'DeleteOnTermination': True|False
            }
        },
    ],
    'VirtualizationType': 'hvm'|'paravirtual',
    'InstanceLifecycle': 'spot',
    'SpotInstanceRequestId': 'string',
    'ClientToken': 'string',
    'Tags': [
```

```

        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
    'SecurityGroups': [
        {
            'GroupName': 'string',
            'GroupId': 'string'
        },
    ],
    'SourceDestCheck': True|False,
    'Hypervisor': 'ovm'|'xen',
    'NetworkInterfaces': [
        {
            'NetworkInterfaceId': 'string',
            'SubnetId': 'string',
            'VpcId': 'string',
            'Description': 'string',
            'OwnerId': 'string',
            'Status': 'available'|'attaching'|'in-use'|'detaching',
            'MacAddress': 'string',
            'PrivateIpAddress': 'string',
            'PrivateDnsName': 'string',
            'SourceDestCheck': True|False,
            'Groups': [
                {
                    'GroupName': 'string',
                    'GroupId': 'string'
                },
            ],
        },
        {
            'Attachment': {
                'AttachmentId': 'string',
                'DeviceIndex': 123,
                'Status': 'attaching'|'attached'|'detaching'|'detached',
                'AttachTime': datetime(2015, 1, 1),
                'DeleteOnTermination': True|False
            },
            'Association': {
                'PublicIp': 'string',
                'PublicDnsName': 'string',
                'IpOwnerId': 'string'
            },
        },
        {
            'PrivateIpAddresses': [
                {
                    'PrivateIpAddress': 'string',
                    'PrivateDnsName': 'string',
                    'Primary': True|False,
                    'Association': {
                        'PublicIp': 'string',
                        'PublicDnsName': 'string',
                        'IpOwnerId': 'string'
                    },
                },
            ],
        },
    ],
    'IamInstanceProfile': {

```

```
        'Arn': 'string',
        'Id': 'string'
    },
    'EbsOptimized': True|False,
    'SriovNetSupport': 'string'
},
]
},
],
'NextToken': 'string'
}
```

Response Structure

- *(dict)* –
 - **Reservations** (*list*) –

One or more reservations.

 - * *(dict)* –

Describes a reservation.

 - **ReservationId** (*string*) –

The ID of the reservation.
 - **OwnerId** (*string*) –

The ID of the AWS account that owns the reservation.
 - **RequesterId** (*string*) –

The ID of the requester that launched the instances on your behalf (for example, AWS Management Console or Auto Scaling).
 - **Groups** (*list*) –

One or more security groups.
 - *(dict)* –

Describes a security group.
 - **GroupName** (*string*) –

The name of the security group.
 - **GroupId** (*string*) –

The ID of the security group.
 - **Instances** (*list*) –

One or more instances.
 - *(dict)* –

Describes an instance.
 - **InstanceId** (*string*) –

The ID of the instance.
 - **ImageId** (*string*) –

The ID of the AMI used to launch the instance.
 - **State** (*dict*) –

The current state of the instance.
 - **Code** (*integer*) –

The low byte represents the state. The high byte is an opaque internal value and should be ignored.

- 0 : pending
- 16 : running
- 32 : shutting-down
- 48 : terminated
- 64 : stopping
- 80 : stopped
- **Name** (*string*) –
The current state of the instance.
- **PrivateDnsName** (*string*) –
The private DNS name assigned to the instance. This DNS name can only be used inside the Amazon EC2 network. This name is not available until the instance enters the `running` state.
- **PublicDnsName** (*string*) –
The public DNS name assigned to the instance. This name is not available until the instance enters the `running` state.
- **StateTransitionReason** (*string*) –
The reason for the most recent state transition. This might be an empty string.
- **KeyName** (*string*) –
The name of the key pair, if this instance was launched with an associated key pair.
- **AmiLaunchIndex** (*integer*) –
The AMI launch index, which can be used to find this instance in the launch group.
- **ProductCodes** (*list*) –
The product codes attached to this instance.
- (*dict*) –
Describes a product code.
- **ProductCodeId** (*string*) –
The product code.
- **ProductCodeType** (*string*) –
The type of product code.
- **InstanceType** (*string*) –
The instance type.
- **LaunchTime** (*datetime*) –
The time the instance was launched.
- **Placement** (*dict*) –
The location where the instance launched.
- **AvailabilityZone** (*string*) –
The Availability Zone of the instance.
- **GroupName** (*string*) –
The name of the placement group the instance is in (for cluster compute instances).
- **Tenancy** (*string*) –

The tenancy of the instance (if the instance is running in a VPC). An instance with a tenancy of `dedicated` runs on single-tenant hardware.

- **KernelId** (*string*) –

The kernel associated with this instance.

- **RamdiskId** (*string*) –

The RAM disk associated with this instance.

- **Platform** (*string*) –

The value is `Windows` for Windows instances; otherwise blank.

- **Monitoring** (*dict*) –

The monitoring information for the instance.

- **State** (*string*) –

Indicates whether monitoring is enabled for the instance.

- **SubnetId** (*string*) –

The ID of the subnet in which the instance is running.

- **VpcId** (*string*) –

The ID of the VPC in which the instance is running.

- **PrivateIpAddress** (*string*) –

The private IP address assigned to the instance.

- **PublicIpAddress** (*string*) –

The public IP address assigned to the instance.

- **StateReason** (*dict*) –

The reason for the most recent state transition.

- **Code** (*string*) –

The reason code for the state change.

- **Message** (*string*) –

The message for the state change.

- `Server.SpotInstanceTermination` : A Spot Instance was terminated due to an increase in the market price.

- `Server.InternalError` : An internal error occurred during instance launch, resulting in termination.

- `Server.InsufficientInstanceCapacity` : There was insufficient instance capacity to satisfy the launch request.

- `Client.InternalError` : A client error caused the instance to terminate on launch.

- `Client.InstanceInitiatedShutdown` : The instance was shut down using the shutdown `-h` command from the instance.

- `Client.UserInitiatedShutdown` : The instance was shut down using the Amazon EC2 API.

- `Client.VolumeLimitExceeded` : The volume limit was exceeded.

- `Client.InvalidSnapshot.NotFound` : The specified snapshot was not found.

- **Architecture** (*string*) –

The architecture of the image.

- **RootDeviceType** (*string*) –

The root device type used by the AMI. The AMI can use an EBS volume or an instance store volume.

- **RootDeviceName** (*string*) –

The root device name (for example, `/dev/sda1` or `/dev/xvda`).

- **BlockDeviceMappings** (*list*) –

Any block device mapping entries for the instance.

- (*dict*) –

Describes a block device mapping.

- **DeviceName** (*string*) –

The device name exposed to the instance (for example, `/dev/sdh` or `xvdh`).

- **Ebs** (*dict*) –

Parameters used to automatically set up EBS volumes when the instance is launched.

- **VolumeId** (*string*) –

The ID of the EBS volume.

- **Status** (*string*) –

The attachment state.

- **AttachTime** (*datetime*) –

The time stamp when the attachment initiated.

- **DeleteOnTermination** (*boolean*) –

Indicates whether the volume is deleted on instance termination.

- **VirtualizationType** (*string*) –

The virtualization type of the instance.

- **InstanceLifecycle** (*string*) –

Indicates whether this is a Spot Instance.

- **SpotInstanceRequestId** (*string*) –

The ID of the Spot Instance request.

- **ClientToken** (*string*) –

The idempotency token you provided when you launched the instance.

- **Tags** (*list*) –

Any tags assigned to the instance.

- (*dict*) –

Describes a tag.

- **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

- **Value** (*string*) –
The value of the tag.
Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.
- **SecurityGroups** (*list*) –
One or more security groups for the instance.
- (*dict*) –
Describes a security group.
- **GroupName** (*string*) –
The name of the security group.
- **GroupId** (*string*) –
The ID of the security group.
- **SourceDestCheck** (*boolean*) –
Specifies whether to enable an instance launched in a VPC to perform NAT. This controls whether source/destination checking is enabled on the instance. A value of `true` means checking is enabled, and `false` means checking is disabled. The value must be `false` for the instance to perform NAT. For more information, see [NAT Instances](#) in the *Amazon Virtual Private Cloud User Guide*.
- **Hypervisor** (*string*) –
The hypervisor type of the instance.
- **NetworkInterfaces** (*list*) –
[EC2-VPC] One or more network interfaces for the instance.
- (*dict*) –
Describes a network interface.
- **NetworkInterfaceId** (*string*) –
The ID of the network interface.
- **SubnetId** (*string*) –
The ID of the subnet.
- **VpcId** (*string*) –
The ID of the VPC.
- **Description** (*string*) –
The description.
- **OwnerId** (*string*) –
The ID of the AWS account that created the network interface.
- **Status** (*string*) –
The status of the network interface.
- **MacAddress** (*string*) –
The MAC address.
- **PrivateIpAddress** (*string*) –
The IP address of the network interface within the subnet.
- **PrivateDnsName** (*string*) –
The private DNS name.

- **SourceDestCheck** (*boolean*) –
Indicates whether to validate network traffic to or from this network interface.
- **Groups** (*list*) –
One or more security groups.
- (*dict*) –
Describes a security group.
- **GroupName** (*string*) –
The name of the security group.
- **GroupId** (*string*) –
The ID of the security group.
- **Attachment** (*dict*) –
The network interface attachment.
- **AttachmentId** (*string*) –
The ID of the network interface attachment.
- **DeviceIndex** (*integer*) –
The index of the device on the instance for the network interface attachment.
- **Status** (*string*) –
The attachment state.
- **AttachTime** (*datetime*) –
The time stamp when the attachment initiated.
- **DeleteOnTermination** (*boolean*) –
Indicates whether the network interface is deleted when the instance is terminated.
- **Association** (*dict*) –
The association information for an Elastic IP associated with the network interface.
- **PublicIp** (*string*) –
The public IP address or Elastic IP address bound to the network interface.
- **PublicDnsName** (*string*) –
The public DNS name.
- **IpOwnerId** (*string*) –
The ID of the owner of the Elastic IP address.
- **PrivateIpAddresses** (*list*) –
The private IP addresses associated with the network interface.
- (*dict*) –
Describes a private IP address.
- **PrivateIpAddress** (*string*) –
The private IP address of the network interface.
- **PrivateDnsName** (*string*) –
The private DNS name.

- **Primary** (*boolean*) –
Indicates whether this IP address is the primary private IP address of the network interface.
- **Association** (*dict*) –
The association information for an Elastic IP address for the network interface.
- **PublicIp** (*string*) –
The public IP address or Elastic IP address bound to the network interface.
- **PublicDnsName** (*string*) –
The public DNS name.
- **IpOwnerId** (*string*) –
The ID of the owner of the Elastic IP address.
- **IamInstanceProfile** (*dict*) –
The IAM instance profile associated with the instance.
- **Arn** (*string*) –
The Amazon Resource Name (ARN) of the instance profile.
- **Id** (*string*) –
The ID of the instance profile.
- **EbsOptimized** (*boolean*) –
Indicates whether the instance is optimized for EBS I/O. This optimization provides dedicated throughput to Amazon EBS and an optimized configuration stack to provide optimal I/O performance. This optimization isn't available with all instance types. Additional usage charges apply when using an EBS Optimized instance.
- **SriovNetSupport** (*string*) –
Specifies whether enhanced networking is enabled.
- **NextToken** (*string*) –
The token to use to retrieve the next page of results. This value is `null` when there are no more results to return.

describe_internet_gateways (***kwargs*)

Describes one or more of your Internet gateways.

Request Syntax

```
response = client.describe_internet_gateways(  
    DryRun=True|False,  
    InternetGatewayIds=[  
        'string',  
    ],  
    Filters=[  
        {  
            'Name': 'string',  
            'Values': [  
                'string',  
            ]  
        },  
    ],  
)
```

```
]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InternetGatewayIds** (*list*) – One or more Internet gateway IDs.

Default: Describes all your Internet gateways.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `attachment.state` - The current state of the attachment between the gateway and the VPC (`available`). Present only if a VPC is attached.
 - `attachment.vpc-id` - The ID of an attached VPC.
 - `internet-gateway-id` - The ID of the Internet gateway.
 - `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
 - `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “tag-key=Purpose” and the filter “tag-value=X”, you get any resources assigned both the tag key Purpose (regardless of what the tag’s value is), and the tag value X (regardless of what the tag’s key is). If you want to list only resources where Purpose is X, see the `tag :key =*value*` filter.
 - `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.

– (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'InternetGateways': [
    {
      'InternetGatewayId': 'string',
      'Attachments': [
        {
          'VpcId': 'string',
          'State': 'attaching'|'attached'|'detaching'|'detached'
        },
      ],
      'Tags': [
        {
          'Key': 'string',
```

```
        'Value': 'string'
    },
    ]
},
]
```

Response Structure

- *(dict)* –

- **InternetGateways** (*list*) –

Information about one or more Internet gateways.

- * *(dict)* –

Describes an Internet gateway.

- **InternetGatewayId** (*string*) –

The ID of the Internet gateway.

- **Attachments** (*list*) –

Any VPCs attached to the Internet gateway.

- *(dict)* –

Describes the attachment of a VPC to an Internet gateway.

- **VpcId** (*string*) –

The ID of the VPC.

- **State** (*string*) –

The current state of the attachment.

- **Tags** (*list*) –

Any tags assigned to the Internet gateway.

- *(dict)* –

Describes a tag.

- **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

- **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

describe_key_pairs (***kwargs*)

Describes one or more of your key pairs.

For more information about key pairs, see [Key Pairs](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.describe_key_pairs(
    DryRun=True|False,
    KeyNames=[
        'string',
    ],
    Filters=[
```

```

    {
        'Name': 'string',
        'Values': [
            'string',
        ]
    },
]
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **KeyNames** (*list*) – One or more key pair names.

Default: Describes all your key pairs.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `fingerprint` - The fingerprint of the key pair.
 - `key-name` - The name of the key pair.
 - (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

- * **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

- * **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

- (*string*) –

Return type dict

Returns

Response Syntax

```

{
    'KeyPairs': [
        {
            'KeyName': 'string',
            'KeyFingerprint': 'string'
        },
    ]
}

```

Response Structure

- (*dict*) –
 - **KeyPairs** (*list*) –

Information about one or more key pairs.

- * (*dict*) –

Describes a key pair.

- **KeyName** (*string*) –

The name of the key pair.

- **KeyFingerprint** (*string*) –

If you used `CreateKeyPair` to create the key pair, this is the SHA-1 digest of the DER encoded private key. If you used `ImportKeyPair` to provide AWS the public key, this is the MD5 public key fingerprint as specified in section 4 of RFC4716.

describe_moving_addresses (***kwargs*)

Describes your Elastic IP addresses that are being moved to the EC2-VPC platform, or that are being restored to the EC2-Classic platform. This request does not return information about any other Elastic IP addresses in your account.

Request Syntax

```
response = client.describe_moving_addresses(
    DryRun=True|False,
    PublicIps=[
        'string',
    ],
    NextToken='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **PublicIps** (*list*) – One or more Elastic IP addresses.
 - (*string*) –
- **NextToken** (*string*) – The token to use to retrieve the next page of results.
- **Filters** (*list*) – One or more filters.
 - `moving-status` - The status of the Elastic IP address (`MovingToVpc` | `RestoringToClassic`).

- (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

- * **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

- * **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

- (*string*) –

- **MaxResults** (*integer*) – The maximum number of results to return for the request in a single page. The remaining results of the initial request can be seen by sending another request with the returned `NextToken` value. This value can be

between 5 and 1000; if `MaxResults` is given a value outside of this range, an error is returned.

Default: If no value is provided, the default is 1000.

Return type dict

Returns

Response Syntax

```
{
  'MovingAddressStatuses': [
    {
      'PublicIp': 'string',
      'MoveStatus': 'movingToVpc'|'restoringToClassic'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (dict) –
 - **MovingAddressStatuses** (list) –

The status for each Elastic IP address.

 - * (dict) –

Describes the status of a moving Elastic IP address.

 - **PublicIp** (string) –

The Elastic IP address.
 - **MoveStatus** (string) –

The status of the Elastic IP address that's being moved to the EC2-VPC platform, or restored to the EC2-Classic platform.
 - **NextToken** (string) –

The token to use to retrieve the next page of results. This value is `null` when there are no more results to return.

describe_network_acls (**kwargs)

Describes one or more of your network ACLs.

For more information about network ACLs, see [Network ACLs](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.describe_network_acls(
    DryRun=True|False,
    NetworkAclIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **NetworkAclIds** (*list*) – One or more network ACL IDs.

Default: Describes all your network ACLs.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `association.association-id` - The ID of an association ID for the ACL.
 - `association.network-acl-id` - The ID of the network ACL involved in the association.
 - `association.subnet-id` - The ID of the subnet involved in the association.
 - `default` - Indicates whether the ACL is the default network ACL for the VPC.
 - `entry.cidr` - The CIDR range specified in the entry.
 - `entry.egress` - Indicates whether the entry applies to egress traffic.
 - `entry.icmp.code` - The ICMP code specified in the entry, if any.
 - `entry.icmp.type` - The ICMP type specified in the entry, if any.
 - `entry.port-range.from` - The start of the port range specified in the entry.
 - `entry.port-range.to` - The end of the port range specified in the entry.
 - `entry.protocol` - The protocol specified in the entry (`tcp` | `udp` | `icmp` or a protocol number).
 - `entry.rule-action` - Allows or denies the matching traffic (`allow` | `deny`).
 - `entry.rule-number` - The number of an entry (in other words, rule) in the ACL's set of entries.
 - `network-acl-id` - The ID of the network ACL.
 - `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
 - `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “tag-key=Purpose” and the filter “tag-value=X”, you get any resources assigned both the tag key Purpose (regardless of what the tag's value is), and the tag value X (regardless of what the tag's key is). If you want to list only resources where Purpose is X, see the `tag :key =*value*` filter.
 - `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
 - `vpc-id` - The ID of the VPC for the network ACL.
- (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'NetworkAcls': [
    {
      'NetworkAclId': 'string',
      'VpcId': 'string',
      'IsDefault': True|False,
      'Entries': [
        {
          'RuleNumber': 123,
          'Protocol': 'string',
          'RuleAction': 'allow'|'deny',
          'Egress': True|False,
          'CidrBlock': 'string',
          'IcmpTypeCode': {
            'Type': 123,
            'Code': 123
          },
          'PortRange': {
            'From': 123,
            'To': 123
          }
        },
      ],
    },
    'Associations': [
      {
        'NetworkAclAssociationId': 'string',
        'NetworkAclId': 'string',
        'SubnetId': 'string'
      },
    ],
    'Tags': [
      {
        'Key': 'string',
        'Value': 'string'
      },
    ],
  ],
}
```

Response Structure

• (*dict*) –

– **NetworkAcls** (*list*) –

Information about one or more network ACLs.

* (*dict*) –

Describes a network ACL.

· **NetworkAclId** (*string*) –

The ID of the network ACL.

· **VpcId** (*string*) –

The ID of the VPC for the network ACL.

- **IsDefault** (*boolean*) –

Indicates whether this is the default network ACL for the VPC.

- **Entries** (*list*) –

One or more entries (rules) in the network ACL.

- (*dict*) –

Describes an entry in a network ACL.

- **RuleNumber** (*integer*) –

The rule number for the entry. ACL entries are processed in ascending order by rule number.

- **Protocol** (*string*) –

The protocol. A value of -1 means all protocols.

- **RuleAction** (*string*) –

Indicates whether to allow or deny the traffic that matches the rule.

- **Egress** (*boolean*) –

Indicates whether the rule is an egress rule (applied to traffic leaving the subnet).

- **CidrBlock** (*string*) –

The network range to allow or deny, in CIDR notation.

- **IcmpTypeCode** (*dict*) –

ICMP protocol: The ICMP type and code.

- **Type** (*integer*) –

The ICMP code. A value of -1 means all codes for the specified ICMP type.

- **Code** (*integer*) –

The ICMP type. A value of -1 means all types.

- **PortRange** (*dict*) –

TCP or UDP protocols: The range of ports the rule applies to.

- **From** (*integer*) –

The first port in the range.

- **To** (*integer*) –

The last port in the range.

- **Associations** (*list*) –

Any associations between the network ACL and one or more subnets

- (*dict*) –

Describes an association between a network ACL and a subnet.

- **NetworkAclAssociationId** (*string*) –

The ID of the association between a network ACL and a subnet.

- **NetworkAclId** (*string*) –

The ID of the network ACL.

- **SubnetId** (*string*) –
The ID of the subnet.
- **Tags** (*list*) –
Any tags assigned to the network ACL.
- (*dict*) –
Describes a tag.
- **Key** (*string*) –
The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`
- **Value** (*string*) –
The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

describe_network_interface_attribute (***kwargs*)

Describes a network interface attribute. You can specify only one attribute at a time.

Request Syntax

```
response = client.describe_network_interface_attribute(
    DryRun=True|False,
    NetworkInterfaceId='string',
    Attribute='description'|'groupSet'|'sourceDestCheck'|'attachment'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **NetworkInterfaceId** (*string*) – **[REQUIRED]**
The ID of the network interface.
- **Attribute** (*string*) – The attribute of the network interface.

Return type dict

Returns

Response Syntax

```
{
  'NetworkInterfaceId': 'string',
  'Description': {
    'Value': 'string'
  },
  'SourceDestCheck': {
    'Value': True|False
  },
  'Groups': [
    {
      'GroupName': 'string',
      'GroupId': 'string'
    },
  ],
}
```

```
'Attachment': {
    'AttachmentId': 'string',
    'InstanceId': 'string',
    'InstanceOwnerId': 'string',
    'DeviceIndex': 123,
    'Status': 'attaching'|'attached'|'detaching'|'detached',
    'AttachTime': datetime(2015, 1, 1),
    'DeleteOnTermination': True|False
}
```

Response Structure

- *(dict)* –
 - **NetworkInterfaceId** (*string*) –
The ID of the network interface.
 - **Description** (*dict*) –
The description of the network interface.
 - * **Value** (*string*) –
Valid values are case-sensitive and vary by action.
 - **SourceDestCheck** (*dict*) –
Indicates whether source/destination checking is enabled.
 - * **Value** (*boolean*) –
Valid values are `true` or `false`.
 - **Groups** (*list*) –
The security groups associated with the network interface.
 - * (*dict*) –
Describes a security group.
 - **GroupName** (*string*) –
The name of the security group.
 - **GroupId** (*string*) –
The ID of the security group.
 - **Attachment** (*dict*) –
The attachment (if any) of the network interface.
 - * **AttachmentId** (*string*) –
The ID of the network interface attachment.
 - * **InstanceId** (*string*) –
The ID of the instance.
 - * **InstanceOwnerId** (*string*) –
The AWS account ID of the owner of the instance.
 - * **DeviceIndex** (*integer*) –
The device index of the network interface attachment on the instance.
 - * **Status** (*string*) –
The attachment state.
 - * **AttachTime** (*datetime*) –
The timestamp indicating when the attachment initiated.

* **DeleteOnTermination** (*boolean*) –

Indicates whether the network interface is deleted when the instance is terminated.

describe_network_interfaces (***kwargs*)

Describes one or more of your network interfaces.

Request Syntax

```
response = client.describe_network_interfaces(
    DryRun=True|False,
    NetworkInterfaceIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **NetworkInterfaceIds** (*list*) – One or more network interface IDs.

Default: Describes all your network interfaces.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `addresses.private-ip-address` - The private IP addresses associated with the network interface.
 - `addresses.primary` - Whether the private IP address is the primary IP address associated with the network interface.
 - `addresses.association.public-ip` - The association ID returned when the network interface was associated with the Elastic IP address.
 - `addresses.association.owner-id` - The owner ID of the addresses associated with the network interface.
 - `association.association-id` - The association ID returned when the network interface was associated with an IP address.
 - `association.allocation-id` - The allocation ID returned when you allocated the Elastic IP address for your network interface.
 - `association.ip-owner-id` - The owner of the Elastic IP address associated with the network interface.
 - `association.public-ip` - The address of the Elastic IP address bound to the network interface.
 - `association.public-dns-name` - The public DNS name for the network interface.
 - `attachment.attachment-id` - The ID of the interface attachment.
 - `attachment.instance-id` - The ID of the instance to which the network interface is attached.

- `attachment.instance-owner-id` - The owner ID of the instance to which the network interface is attached.
- `attachment.device-index` - The device index to which the network interface is attached.
- `attachment.status` - The status of the attachment (`attaching` | `attached` | `detaching` | `detached`).
- `attachment.attach.time` - The time that the network interface was attached to an instance.
- `attachment.delete-on-termination` - Indicates whether the attachment is deleted when an instance is terminated.
- `availability-zone` - The Availability Zone of the network interface.
- `description` - The description of the network interface.
- `group-id` - The ID of a security group associated with the network interface.
- `group-name` - The name of a security group associated with the network interface.
- `mac-address` - The MAC address of the network interface.
- `network-interface-id` - The ID of the network interface.
- `owner-id` - The AWS account ID of the network interface owner.
- `private-ip-address` - The private IP address or addresses of the network interface.
- `private-dns-name` - The private DNS name of the network interface.
- `requester-id` - The ID of the entity that launched the instance on your behalf (for example, AWS Management Console, Auto Scaling, and so on).
- `requester-managed` - Indicates whether the network interface is being managed by an AWS service (for example, AWS Management Console, Auto Scaling, and so on).
- `source-check` - Indicates whether the network interface performs source/destination checking. A value of `true` means checking is enabled, and `false` means checking is disabled. The value must be `false` for the network interface to perform Network Address Translation (NAT) in your VPC.
- `status` - The status of the network interface. If the network interface is not attached to an instance, the status is `available`; if a network interface is attached to an instance the status is `in-use`.
- `subnet-id` - The ID of the subnet for the network interface.
- `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag’s value is), and the tag value `X` (regardless of what the tag’s key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `vpc-id` - The ID of the VPC for the network interface.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'NetworkInterfaces': [
    {
      'NetworkInterfaceId': 'string',
      'SubnetId': 'string',
      'VpcId': 'string',
      'AvailabilityZone': 'string',
      'Description': 'string',
      'OwnerId': 'string',
      'RequesterId': 'string',
      'RequesterManaged': True|False,
      'Status': 'available'|'attaching'|'in-use'|'detaching',
      'MacAddress': 'string',
      'PrivateIpAddress': 'string',
      'PrivateDnsName': 'string',
      'SourceDestCheck': True|False,
      'Groups': [
        {
          'GroupName': 'string',
          'GroupId': 'string'
        },
      ],
    },
    'Attachment': {
      'AttachmentId': 'string',
      'InstanceId': 'string',
      'InstanceOwnerId': 'string',
      'DeviceIndex': 123,
      'Status': 'attaching'|'attached'|'detaching'|'detached',
      'AttachTime': datetime(2015, 1, 1),
      'DeleteOnTermination': True|False
    },
    'Association': {
      'PublicIp': 'string',
      'PublicDnsName': 'string',
      'IpOwnerId': 'string',
      'AllocationId': 'string',
      'AssociationId': 'string'
    },
    'TagSet': [
      {
        'Key': 'string',
        'Value': 'string'
      },
    ],
    'PrivateIpAddresses': [
      {
        'PrivateIpAddress': 'string',
        'PrivateDnsName': 'string',
        'Primary': True|False,
```

```
        'Association': {
            'PublicIp': 'string',
            'PublicDnsName': 'string',
            'IpOwnerId': 'string',
            'AllocationId': 'string',
            'AssociationId': 'string'
        },
    ],
},
]
```

Response Structure

- *(dict)* –

- **NetworkInterfaces** (*list*) –

Information about one or more network interfaces.

- * *(dict)* –

Describes a network interface.

- **NetworkInterfaceId** (*string*) –

The ID of the network interface.

- **SubnetId** (*string*) –

The ID of the subnet.

- **VpcId** (*string*) –

The ID of the VPC.

- **AvailabilityZone** (*string*) –

The Availability Zone.

- **Description** (*string*) –

A description.

- **OwnerId** (*string*) –

The AWS account ID of the owner of the network interface.

- **RequesterId** (*string*) –

The ID of the entity that launched the instance on your behalf (for example, AWS Management Console or Auto Scaling).

- **RequesterManaged** (*boolean*) –

Indicates whether the network interface is being managed by AWS.

- **Status** (*string*) –

The status of the network interface.

- **MacAddress** (*string*) –

The MAC address.

- **PrivateIpAddress** (*string*) –

The IP address of the network interface within the subnet.

- **PrivateDnsName** (*string*) –

The private DNS name.

- **SourceDestCheck** (*boolean*) –

Indicates whether traffic to or from the instance is validated.

- **Groups** (*list*) –
Any security groups for the network interface.
- (*dict*) –
Describes a security group.
- **GroupName** (*string*) –
The name of the security group.
- **GroupId** (*string*) –
The ID of the security group.
- **Attachment** (*dict*) –
The network interface attachment.
- **AttachmentId** (*string*) –
The ID of the network interface attachment.
- **InstanceId** (*string*) –
The ID of the instance.
- **InstanceOwnerId** (*string*) –
The AWS account ID of the owner of the instance.
- **DeviceIndex** (*integer*) –
The device index of the network interface attachment on the instance.
- **Status** (*string*) –
The attachment state.
- **AttachTime** (*datetime*) –
The timestamp indicating when the attachment initiated.
- **DeleteOnTermination** (*boolean*) –
Indicates whether the network interface is deleted when the instance is terminated.
- **Association** (*dict*) –
The association information for an Elastic IP associated with the network interface.
- **PublicIp** (*string*) –
The address of the Elastic IP address bound to the network interface.
- **PublicDnsName** (*string*) –
The public DNS name.
- **IpOwnerId** (*string*) –
The ID of the Elastic IP address owner.
- **AllocationId** (*string*) –
The allocation ID.
- **AssociationId** (*string*) –
The association ID.
- **TagSet** (*list*) –
Any tags assigned to the network interface.
- (*dict*) –
Describes a tag.

- **Key** (*string*) –
The key of the tag.
Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`
- **Value** (*string*) –
The value of the tag.
Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.
- **PrivateIpAddresses** (*list*) –
The private IP addresses associated with the network interface.
- (*dict*) –
Describes the private IP address of a network interface.
- **PrivateIpAddress** (*string*) –
The private IP address.
- **PrivateDnsName** (*string*) –
The private DNS name.
- **Primary** (*boolean*) –
Indicates whether this IP address is the primary private IP address of the network interface.
- **Association** (*dict*) –
The association information for an Elastic IP address associated with the network interface.
- **PublicIp** (*string*) –
The address of the Elastic IP address bound to the network interface.
- **PublicDnsName** (*string*) –
The public DNS name.
- **IpOwnerId** (*string*) –
The ID of the Elastic IP address owner.
- **AllocationId** (*string*) –
The allocation ID.
- **AssociationId** (*string*) –
The association ID.

describe_placement_groups (***kwargs*)

Describes one or more of your placement groups. For more information about placement groups and cluster instances, see [Cluster Instances](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.describe_placement_groups(
    DryRun=True|False,
    GroupNames=[
        'string',
    ],
    Filters=[
        {
```

```

        'Name': 'string',
        'Values': [
            'string',
        ]
    },
]
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **GroupNames** (*list*) – One or more placement group names.

Default: Describes all your placement groups, or only those otherwise specified.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `group-name` - The name of the placement group.
 - `state` - The state of the placement group (`pending` | `available` | `deleting` | `deleted`).
 - `strategy` - The strategy of the placement group (`cluster`).
 - (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

- * **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

- * **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

- (*string*) –

Return type dict

Returns

Response Syntax

```

{
    'PlacementGroups': [
        {
            'GroupName': 'string',
            'Strategy': 'cluster',
            'State': 'pending' | 'available' | 'deleting' | 'deleted'
        },
    ]
}

```

Response Structure

- (*dict*) –
 - **PlacementGroups** (*list*) –

One or more placement groups.

 - * (*dict*) –

Describes a placement group.

- **GroupName** (*string*) –
The name of the placement group.
- **Strategy** (*string*) –
The placement strategy.
- **State** (*string*) –
The state of the placement group.

describe_prefix_lists (***kwargs*)

Describes available AWS services in a prefix list format, which includes the prefix list name and prefix list ID of the service and the IP address range for the service. A prefix list ID is required for creating an outbound security group rule that allows traffic from a VPC to access an AWS service through a VPC endpoint.

Request Syntax

```
response = client.describe_prefix_lists(
    DryRun=True|False,
    PrefixListIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxResults=123,
    NextToken='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **PrefixListIds** (*list*) – One or more prefix list IDs.
 - (*string*) –
- **Filters** (*list*) – One or more filters.
 - `prefix-list-id`: The ID of a prefix list.
 - `prefix-list-name`: The name of a prefix list.
 - (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

- * **Name** (*string*) –
The name of the filter. Filter names are case-sensitive.
- * **Values** (*list*) –
One or more filter values. Filter values are case-sensitive.
 - (*string*) –

- **MaxResults** (*integer*) – The maximum number of items to return for this request. The request returns a token that you can specify in a subsequent call to get the next set of results.

Constraint: If the value specified is greater than 1000, we return only 1000 items.

- **NextToken** (*string*) – The token for the next set of items to return. (You received this token from a prior call.)

Return type dict

Returns

Response Syntax

```
{
  'PrefixLists': [
    {
      'PrefixListId': 'string',
      'PrefixListName': 'string',
      'Cidrs': [
        'string',
      ]
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **PrefixLists** (*list*) –

All available prefix lists.

 - * (*dict*) –

Describes prefixes for AWS services.

 - **PrefixListId** (*string*) –

The ID of the prefix.
 - **PrefixListName** (*string*) –

The name of the prefix.
 - **Cidrs** (*list*) –

The IP address range of the AWS service.
 - (*string*) –
 - **NextToken** (*string*) –

The token to use when requesting the next set of items. If there are no additional items to return, the string is empty.

describe_regions (***kwargs*)

Describes one or more regions that are currently available to you.

For a list of the regions supported by Amazon EC2, see [Regions and Endpoints](#) .

Request Syntax

```
response = client.describe_regions(
    DryRun=True|False,
    RegionNames=[
        'string',
    ],
    Filters=[
```

```
{
    'Name': 'string',
    'Values': [
        'string',
    ]
},
]
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **RegionNames** (*list*) – The names of one or more regions.

- (*string*) –

- **Filters** (*list*) – One or more filters.

- `endpoint` - The endpoint of the region (for example, `ec2.us-east-1.amazonaws.com`).

- `region-name` - The name of the region (for example, `us-east-1`).

- (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

- * **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

- * **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'Regions': [
        {
            'RegionName': 'string',
            'Endpoint': 'string'
        },
    ]
}
```

Response Structure

- (*dict*) –

- **Regions** (*list*) –

Information about one or more regions.

- * (*dict*) –

Describes a region.

- **RegionName** (*string*) –

The name of the region.

- **Endpoint** (*string*) –
The region service endpoint.

describe_reserved_instances (***kwargs*)

Describes one or more of the Reserved Instances that you purchased.

For more information about Reserved Instances, see [Reserved Instances](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.describe_reserved_instances(
    DryRun=True|False,
    ReservedInstancesIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    OfferingType='Heavy Utilization'|'Medium Utilization'|'Light Utilization'|'No Upfront'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **ReservedInstancesIds** (*list*) – One or more Reserved Instance IDs.

Default: Describes all your Reserved Instances, or only those otherwise specified.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `availability-zone` - The Availability Zone where the Reserved Instance can be used.
 - `duration` - The duration of the Reserved Instance (one year or three years), in seconds (31536000 | 94608000).
 - `end` - The time when the Reserved Instance expires (for example, 2015-08-07T11:54:42.000Z).
 - `fixed-price` - The purchase price of the Reserved Instance (for example, 9800.0).
 - `instance-type` - The instance type on which the Reserved Instance can be used.
 - `product-description` - The Reserved Instance product platform description. Instances that include (Amazon VPC) in the product platform description will only be displayed to EC2-Classical account holders and are for use with Amazon VPC. (Linux/UNIX | Linux/UNIX (Amazon VPC) | SUSE Linux | SUSE Linux (Amazon VPC) | Red Hat Enterprise Linux | Red Hat Enterprise Linux (Amazon VPC) | Windows | Windows (Amazon VPC) | Windows with SQL Server Standard | Windows with SQL Server Standard (Amazon VPC) | Windows with SQL Server

Web | Windows with SQL Server Web (Amazon VPC) |
Windows with SQL Server Enterprise | Windows with
SQL Server Enterprise (Amazon VPC)).

- reserved-instances-id - The ID of the Reserved Instance.
- start - The time at which the Reserved Instance purchase request was placed (for example, 2014-08-07T11:54:42.000Z).
- state - The state of the Reserved Instance (payment-pending | active | payment-failed | retired).
- tag :key =*value* - The key/value combination of a tag assigned to the resource.
- tag-key - The key of a tag assigned to the resource. This filter is independent of the tag-value filter. For example, if you use both the filter “tag-key=Purpose” and the filter “tag-value=X”, you get any resources assigned both the tag key Purpose (regardless of what the tag’s value is), and the tag value X (regardless of what the tag’s key is). If you want to list only resources where Purpose is X, see the tag :key =*value* filter.
- tag-value - The value of a tag assigned to the resource. This filter is independent of the tag-key filter.
- usage-price - The usage price of the Reserved Instance, per hour (for example, 0.84).
- (dict) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

- **OfferingType** (*string*) – The Reserved Instance offering type. If you are using tools that predate the 2011-11-01 API version, you only have access to the Medium Utilization Reserved Instance offering type.

Return type dict

Returns

Response Syntax

```
{
  'ReservedInstances': [
    {
      'ReservedInstancesId': 'string',
      'InstanceType': 't1.micro'|'m1.small'|'m1.medium'|'m1.large'|'m1.xlarge',
      'AvailabilityZone': 'string',
      'Start': datetime(2015, 1, 1),
      'End': datetime(2015, 1, 1),
      'Duration': 123,
      'UsagePrice': ...,
      'FixedPrice': ...,
      'InstanceCount': 123,
      'ProductDescription': 'Linux/UNIX'|'Linux/UNIX (Amazon VPC)'|'Windows',
      'State': 'payment-pending'|'active'|'payment-failed'|'retired',
      'Tags': [
        {
```

```

        'Key': 'string',
        'Value': 'string'
    },
],
'InstanceTenancy': 'default'|'dedicated',
'CurrencyCode': 'USD',
'OfferingType': 'Heavy Utilization'|'Medium Utilization'|'Light Utilization',
'RecurringCharges': [
    {
        'Frequency': 'Hourly',
        'Amount': 123.0
    },
]
},
]
}

```

Response Structure

- *(dict)* –
 - **ReservedInstances** (*list*) –

A list of Reserved Instances.

 - * *(dict)* –

Describes a Reserved Instance.

 - **ReservedInstancesId** (*string*) –

The ID of the Reserved Instance.
 - **InstanceType** (*string*) –

The instance type on which the Reserved Instance can be used.
 - **AvailabilityZone** (*string*) –

The Availability Zone in which the Reserved Instance can be used.
 - **Start** (*datetime*) –

The date and time the Reserved Instance started.
 - **End** (*datetime*) –

The time when the Reserved Instance expires.
 - **Duration** (*integer*) –

The duration of the Reserved Instance, in seconds.
 - **UsagePrice** (*float*) –

The usage price of the Reserved Instance, per hour.
 - **FixedPrice** (*float*) –

The purchase price of the Reserved Instance.
 - **InstanceCount** (*integer*) –

The number of Reserved Instances purchased.
 - **ProductDescription** (*string*) –

The Reserved Instance product platform description.
 - **State** (*string*) –

The state of the Reserved Instance purchase.
 - **Tags** (*list*) –

- Any tags assigned to the resource.
- *(dict)* –
Describes a tag.
- **Key** (*string*) –
The key of the tag.
Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`
- **Value** (*string*) –
The value of the tag.
Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.
- **InstanceTenancy** (*string*) –
The tenancy of the reserved instance.
- **CurrencyCode** (*string*) –
The currency of the Reserved Instance. It's specified using ISO 4217 standard currency codes. At this time, the only supported currency is `USD`.
- **OfferingType** (*string*) –
The Reserved Instance offering type.
- **RecurringCharges** (*list*) –
The recurring charge tag assigned to the resource.
- *(dict)* –
Describes a recurring charge.
- **Frequency** (*string*) –
The frequency of the recurring charge.
- **Amount** (*float*) –
The amount of the recurring charge.

describe_reserved_instances_listings (***kwargs*)

Describes your account's Reserved Instance listings in the Reserved Instance Marketplace.

The Reserved Instance Marketplace matches sellers who want to resell Reserved Instance capacity that they no longer need with buyers who want to purchase additional capacity. Reserved Instances bought and sold through the Reserved Instance Marketplace work like any other Reserved Instances.

As a seller, you choose to list some or all of your Reserved Instances, and you specify the upfront price to receive for them. Your Reserved Instances are then listed in the Reserved Instance Marketplace and are available for purchase.

As a buyer, you specify the configuration of the Reserved Instance to purchase, and the Marketplace matches what you're searching for with what's available. The Marketplace first sells the lowest priced Reserved Instances to you, and continues to sell available Reserved Instance listings to you until your demand is met. You are charged based on the total price of all of the listings that you purchase.

For more information, see [Reserved Instance Marketplace](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```

response = client.describe_reserved_instances_listings(
    ReservedInstancesId='string',
    ReservedInstancesListingId='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)

```

Parameters

- **ReservedInstancesId** (*string*) – One or more Reserved Instance IDs.
- **ReservedInstancesListingId** (*string*) – One or more Reserved Instance Listing IDs.
- **Filters** (*list*) – One or more filters.
 - reserved-instances-id - The ID of the Reserved Instances.
 - reserved-instances-listing-id - The ID of the Reserved Instances listing.
 - status - The status of the Reserved Instance listing (pending | active | cancelled | closed).
 - status-message - The reason for the status.
 - (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

Return type dict

Returns

Response Syntax

```

{
    'ReservedInstancesListings': [
        {
            'ReservedInstancesListingId': 'string',
            'ReservedInstancesId': 'string',
            'CreateDate': datetime(2015, 1, 1),
            'UpdateDate': datetime(2015, 1, 1),
            'Status': 'active' | 'pending' | 'cancelled' | 'closed',
            'StatusMessage': 'string',
            'InstanceCounts': [
                {
                    'State': 'available' | 'sold' | 'cancelled' | 'pending',
                    'InstanceCount': 123
                },
            ],
            'PriceSchedules': [

```

```
        {
            'Term': 123,
            'Price': 123.0,
            'CurrencyCode': 'USD',
            'Active': True|False
        },
    ],
    'Tags': [
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
    'ClientToken': 'string'
},
]
```

Response Structure

- (*dict*) –

- **ReservedInstancesListings** (*list*) –

Information about the Reserved Instance listing.

- * (*dict*) –

Describes a Reserved Instance listing.

- **ReservedInstancesListingId** (*string*) –

The ID of the Reserved Instance listing.

- **ReservedInstancesId** (*string*) –

The ID of the Reserved Instance.

- **CreateDate** (*datetime*) –

The time the listing was created.

- **UpdateDate** (*datetime*) –

The last modified timestamp of the listing.

- **Status** (*string*) –

The status of the Reserved Instance listing.

- **StatusMessage** (*string*) –

The reason for the current status of the Reserved Instance listing. The response can be blank.

- **InstanceCounts** (*list*) –

The number of instances in this state.

- (*dict*) –

Describes a Reserved Instance listing state.

- **State** (*string*) –

The states of the listed Reserved Instances.

- **InstanceCount** (*integer*) –

The number of listed Reserved Instances in the state specified by the `state`.

- **PriceSchedules** (*list*) –

The price of the Reserved Instance listing.

- *(dict)* –
Describes the price for a Reserved Instance.
- **Term** (*integer*) –
The number of months remaining in the reservation. For example, 2 is the second to the last month before the capacity reservation expires.
- **Price** (*float*) –
The fixed price for the term.
- **CurrencyCode** (*string*) –
The currency for transacting the Reserved Instance resale. At this time, the only supported currency is USD .
- **Active** (*boolean*) –
The current price schedule, as determined by the term remaining for the Reserved Instance in the listing.

A specific price schedule is always in effect, but only one price schedule can be active at any time. Take, for example, a Reserved Instance listing that has five months remaining in its term. When you specify price schedules for five months and two months, this means that schedule 1, covering the first three months of the remaining term, will be active during months 5, 4, and 3. Then schedule 2, covering the last two months of the term, will be active for months 2 and 1.
- **Tags** (*list*) –
Any tags assigned to the resource.
- *(dict)* –
Describes a tag.
- **Key** (*string*) –
The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`
- **Value** (*string*) –
The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.
- **ClientToken** (*string*) –
A unique, case-sensitive key supplied by the client to ensure that the request is idempotent. For more information, see [Ensuring Idempotency](#) .

describe_reserved_instances_modifications (***kwargs*)

Describes the modifications made to your Reserved Instances. If no parameter is specified, information about all your Reserved Instances modification requests is returned. If a modification ID is specified, only information about the specific modification is returned.

For more information, see [Modifying Reserved Instances](#) in the Amazon Elastic Compute Cloud User Guide.

Request Syntax

```
response = client.describe_reserved_instances_modifications(
    ReservedInstancesModificationIds=[
        'string',
    ],
    NextToken='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)
```

Parameters

- **ReservedInstancesModificationIds** (*list*) – IDs for the submitted modification request.
 - (*string*) –
- **NextToken** (*string*) – The token to retrieve the next page of results.
- **Filters** (*list*) – One or more filters.
 - `client-token` - The idempotency token for the modification request.
 - `create-date` - The time when the modification request was created.
 - `effective-date` - The time when the modification becomes effective.
 - `modification-result.reserved-instances-id` - The ID for the Reserved Instances created as part of the modification request. This ID is only available when the status of the modification is fulfilled.
 - `modification-result.target-configuration.availability-zone` - The Availability Zone for the new Reserved Instances.
 - `modification-result.target-configuration.instance-count` - The number of new Reserved Instances.
 - `modification-result.target-configuration.instance-type` - The instance type of the new Reserved Instances.
 - `modification-result.target-configuration.platform` - The network platform of the new Reserved Instances (EC2-Classic | EC2-VPC).
 - `reserved-instances-id` - The ID of the Reserved Instances modified.
 - `reserved-instances-modification-id` - The ID of the modification request.
 - `status` - The status of the Reserved Instances modification request (`processing` | `fulfilled` | `failed`).
 - `status-message` - The reason for the status.
 - `update-date` - The time when the modification request was last updated.
- (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

 - * **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.
 - * **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'ReservedInstancesModifications': [
    {
      'ReservedInstancesModificationId': 'string',
      'ReservedInstancesIds': [
        {
          'ReservedInstancesId': 'string'
        },
      ],
      'ModificationResults': [
        {
          'ReservedInstancesId': 'string',
          'TargetConfiguration': {
            'AvailabilityZone': 'string',
            'Platform': 'string',
            'InstanceCount': 123,
            'InstanceType': 't1.micro'|'m1.small'|'m1.medium'|'m1.large'
          }
        },
      ],
      'CreateDate': datetime(2015, 1, 1),
      'UpdateDate': datetime(2015, 1, 1),
      'EffectiveDate': datetime(2015, 1, 1),
      'Status': 'string',
      'StatusMessage': 'string',
      'ClientToken': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **ReservedInstancesModifications** (*list*) –

The Reserved Instance modification information.

* (*dict*) –

Describes a Reserved Instance modification.

- **ReservedInstancesModificationId** (*string*) –

A unique ID for the Reserved Instance modification.
- **ReservedInstancesIds** (*list*) –

The IDs of one or more Reserved Instances.
- (*dict*) –

Describes the ID of a Reserved Instance.
- **ReservedInstancesId** (*string*) –

The ID of the Reserved Instance.
- **ModificationResults** (*list*) –

Contains target configurations along with their corresponding new Reserved Instance IDs.

- *(dict)* –

- **ReservedInstancesId** (*string*) –

The ID for the Reserved Instances that were created as part of the modification request. This field is only available when the modification is fulfilled.

- **TargetConfiguration** (*dict*) –

The target Reserved Instances configurations supplied as part of the modification request.

- **AvailabilityZone** (*string*) –

The Availability Zone for the modified Reserved Instances.

- **Platform** (*string*) –

The network platform of the modified Reserved Instances, which is either EC2-Classic or EC2-VPC.

- **InstanceCount** (*integer*) –

The number of modified Reserved Instances.

- **InstanceType** (*string*) –

The instance type for the modified Reserved Instances.

- **CreateDate** (*datetime*) –

The time when the modification request was created.

- **UpdateDate** (*datetime*) –

The time when the modification request was last updated.

- **EffectiveDate** (*datetime*) –

The time for the modification to become effective.

- **Status** (*string*) –

The status of the Reserved Instances modification request.

- **StatusMessage** (*string*) –

The reason for the status.

- **ClientToken** (*string*) –

A unique, case-sensitive key supplied by the client to ensure that the request is idempotent. For more information, see [Ensuring Idempotency](#).

- **NextToken** (*string*) –

The token to use to retrieve the next page of results. This value is `null` when there are no more results to return.

describe_reserved_instances_offerings (***kwargs*)

Describes Reserved Instance offerings that are available for purchase. With Reserved Instances, you purchase the right to launch instances for a period of time. During that time period, you do not receive insufficient capacity errors, and you pay a lower usage rate than the rate charged for On-Demand instances for the actual time used.

For more information, see [Reserved Instance Marketplace](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```

response = client.describe_reserved_instances_offerings(
    DryRun=True|False,
    ReservedInstancesOfferingIds=[
        'string',
    ],
    InstanceType='t1.micro'|'m1.small'|'m1.medium'|'m1.large'|'m1.xlarge'|'m3.medium'|'m3.l
    AvailabilityZone='string',
    ProductDescription='Linux/UNIX'|'Linux/UNIX (Amazon VPC)'|'Windows'|'Windows (Amazon VP
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    InstanceTenancy='default'|'dedicated',
    OfferingType='Heavy Utilization'|'Medium Utilization'|'Light Utilization'|'No Upfront'|
    NextToken='string',
    MaxResults=123,
    IncludeMarketplace=True|False,
    MinDuration=123,
    MaxDuration=123,
    MaxInstanceCount=123
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **ReservedInstancesOfferingIds** (*list*) – One or more Reserved Instances offering IDs.
 - (*string*) –
- **InstanceType** (*string*) – The instance type on which the Reserved Instance can be used. For more information, see [Instance Types](#) in the *Amazon Elastic Compute Cloud User Guide*.
- **AvailabilityZone** (*string*) – The Availability Zone in which the Reserved Instance can be used.
- **ProductDescription** (*string*) – The Reserved Instance product platform description. Instances that include (Amazon VPC) in the description are for use with Amazon VPC.
- **Filters** (*list*) – One or more filters.
 - `availability-zone` - The Availability Zone where the Reserved Instance can be used.
 - `duration` - The duration of the Reserved Instance (for example, one year or three years), in seconds (31536000|94608000).
 - `fixed-price` - The purchase price of the Reserved Instance (for example, 9800.0).
 - `instance-type` - The instance type on which the Reserved Instance can be used.
 - `marketplace` - Set to `true` to show only Reserved Instance Marketplace offerings. When this filter is not used, which is the default behavior, all offerings from AWS and Reserved Instance Marketplace are listed.
 - `product-description` - The Reserved Instance product platform

description. Instances that include (Amazon VPC) in the product platform description will only be displayed to EC2-Classic account holders and are for use with Amazon VPC. (Linux/UNIX | Linux/UNIX (Amazon VPC) | SUSE Linux | SUSE Linux (Amazon VPC) | Red Hat Enterprise Linux | Red Hat Enterprise Linux (Amazon VPC) | Windows | Windows (Amazon VPC) | Windows with SQL Server Standard | Windows with SQL Server Standard (Amazon VPC) | Windows with SQL Server Web | Windows with SQL Server Web (Amazon VPC) | Windows with SQL Server Enterprise | Windows with SQL Server Enterprise (Amazon VPC))

- reserved-instances-offering-id - The Reserved Instances offering ID.
- usage-price - The usage price of the Reserved Instance, per hour (for example, 0.84).
- (dict) -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

- **InstanceTenancy** (*string*) - The tenancy of the Reserved Instance offering. A Reserved Instance with `dedicated` tenancy runs on single-tenant hardware and can only be launched within a VPC.

Default: `default`

- **OfferingType** (*string*) - The Reserved Instance offering type. If you are using tools that predate the 2011-11-01 API version, you only have access to the `Medium Utilization` Reserved Instance offering type.
- **NextToken** (*string*) - The token to retrieve the next page of results.
- **MaxResults** (*integer*) - The maximum number of results to return for the request in a single page. The remaining results of the initial request can be seen by sending another request with the returned `NextToken` value. The maximum is 100.

Default: 100

- **IncludeMarketplace** (*boolean*) - Include Marketplace offerings in the response.
- **MinDuration** (*integer*) - The minimum duration (in seconds) to filter when searching for offerings.

Default: 2592000 (1 month)

- **MaxDuration** (*integer*) - The maximum duration (in seconds) to filter when searching for offerings.

Default: 94608000 (3 years)

- **MaxInstanceCount** (*integer*) - The maximum number of instances to filter when searching for offerings.

Default: 20

Return type dict

Returns

Response Syntax

```
{
  'ReservedInstancesOfferings': [
    {
      'ReservedInstancesOfferingId': 'string',
      'InstanceType': 't1.micro'|'m1.small'|'m1.medium'|'m1.large'|'m1.xlarge',
      'AvailabilityZone': 'string',
      'Duration': 123,
      'UsagePrice': ...,
      'FixedPrice': ...,
      'ProductDescription': 'Linux/UNIX'|'Linux/UNIX (Amazon VPC)'|'Windows',
      'InstanceTenancy': 'default'|'dedicated',
      'CurrencyCode': 'USD',
      'OfferingType': 'Heavy Utilization'|'Medium Utilization'|'Light Utilization',
      'RecurringCharges': [
        {
          'Frequency': 'Hourly',
          'Amount': 123.0
        },
      ],
      'Marketplace': True|False,
      'PricingDetails': [
        {
          'Price': 123.0,
          'Count': 123
        },
      ],
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **ReservedInstancesOfferings** (*list*) –
 - A list of Reserved Instances offerings.
 - * (*dict*) –
 - Describes a Reserved Instance offering.
 - **ReservedInstancesOfferingId** (*string*) –
 - The ID of the Reserved Instance offering.
 - **InstanceType** (*string*) –
 - The instance type on which the Reserved Instance can be used.
 - **AvailabilityZone** (*string*) –
 - The Availability Zone in which the Reserved Instance can be used.
 - **Duration** (*integer*) –
 - The duration of the Reserved Instance, in seconds.
 - **UsagePrice** (*float*) –
 - The usage price of the Reserved Instance, per hour.
 - **FixedPrice** (*float*) –

The purchase price of the Reserved Instance.

- **ProductDescription** (*string*) –

The Reserved Instance product platform description.

- **InstanceTenancy** (*string*) –

The tenancy of the reserved instance.

- **CurrencyCode** (*string*) –

The currency of the Reserved Instance offering you are purchasing. It's specified using ISO 4217 standard currency codes. At this time, the only supported currency is USD .

- **OfferingType** (*string*) –

The Reserved Instance offering type.

- **RecurringCharges** (*list*) –

The recurring charge tag assigned to the resource.

- (*dict*) –

Describes a recurring charge.

- **Frequency** (*string*) –

The frequency of the recurring charge.

- **Amount** (*float*) –

The amount of the recurring charge.

- **Marketplace** (*boolean*) –

Indicates whether the offering is available through the Reserved Instance Marketplace (resale) or AWS. If it's a Reserved Instance Marketplace offering, this is `true` .

- **PricingDetails** (*list*) –

The pricing details of the Reserved Instance offering.

- (*dict*) –

Describes a Reserved Instance offering.

- **Price** (*float*) –

The price per instance.

- **Count** (*integer*) –

The number of instances available for the price.

- **NextToken** (*string*) –

The token to use to retrieve the next page of results. This value is `null` when there are no more results to return.

describe_route_tables (***kwargs*)

Describes one or more of your route tables.

Each subnet in your VPC must be associated with a route table. If a subnet is not explicitly associated with any route table, it is implicitly associated with the main route table. This command does not return the subnet ID for implicit associations.

For more information about route tables, see [Route Tables](#) in the *Amazon Virtual Private Cloud User Guide* .

Request Syntax

```

response = client.describe_route_tables(
    DryRun=True|False,
    RouteTableIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **RouteTableIds** (*list*) – One or more route table IDs.

Default: Describes all your route tables.

– (*string*) –

- **Filters** (*list*) – One or more filters.
 - `association.route-table-association-id` - The ID of an association ID for the route table.
 - `association.route-table-id` - The ID of the route table involved in the association.
 - `association.subnet-id` - The ID of the subnet involved in the association.
 - `association.main` - Indicates whether the route table is the main route table for the VPC.
 - `route-table-id` - The ID of the route table.
 - `route.destination-cidr-block` - The CIDR range specified in a route in the table.
 - `route.destination-prefix-list-id` - The ID (prefix) of the AWS service specified in a route in the table.
 - `route.gateway-id` - The ID of a gateway specified in a route in the table.
 - `route.instance-id` - The ID of an instance specified in a route in the table.
 - `route.origin` - Describes how the route was created. `CreateRouteTable` indicates that the route was automatically created when the route table was created; `CreateRoute` indicates that the route was manually added to the route table; `EnableVgwRoutePropagation` indicates that the route was propagated by route propagation.
 - `route.state` - The state of a route in the route table (`active` | `blackhole`). The `blackhole` state indicates that the route's target isn't available (for example, the specified gateway isn't attached to the VPC, the specified NAT instance has been terminated, and so on).
 - `route.vpc-peering-connection-id` - The ID of a VPC peering connection specified in a route in the table.
 - `tag :key =*value*` - The key/value combination of a tag assigned to the

resource.

- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag’s value is), and the tag value `X` (regardless of what the tag’s key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key=*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `vpc-id` - The ID of the VPC for the route table.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

Return type dict

Returns

Response Syntax

```
{
  'RouteTables': [
    {
      'RouteTableId': 'string',
      'VpcId': 'string',
      'Routes': [
        {
          'DestinationCidrBlock': 'string',
          'DestinationPrefixListId': 'string',
          'GatewayId': 'string',
          'InstanceId': 'string',
          'InstanceOwnerId': 'string',
          'NetworkInterfaceId': 'string',
          'VpcPeeringConnectionId': 'string',
          'State': 'active'|'blackhole',
          'Origin': 'CreateRouteTable'|'CreateRoute'|'EnableVgwRouteProp
        },
      ],
      'Associations': [
        {
          'RouteTableAssociationId': 'string',
          'RouteTableId': 'string',
          'SubnetId': 'string',
          'Main': True|False
        },
      ],
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        },
      ],
    },
  ],
}
```



```

    ],
    'PropagatingVgws': [
        {
            'GatewayId': 'string'
        },
    ],
]
},
]
}

```

Response Structure

- *(dict)* –
 - **RouteTables** (*list*) –

Information about one or more route tables.

- * *(dict)* –

Describes a route table.

- **RouteTableId** (*string*) –
The ID of the route table.
- **VpcId** (*string*) –
The ID of the VPC.
- **Routes** (*list*) –
The routes in the route table.
- *(dict)* –
Describes a route in a route table.
- **DestinationCidrBlock** (*string*) –
The CIDR block used for the destination match.
- **DestinationPrefixListId** (*string*) –
The prefix of the AWS service.
- **GatewayId** (*string*) –
The ID of a gateway attached to your VPC.
- **InstanceId** (*string*) –
The ID of a NAT instance in your VPC.
- **InstanceOwnerId** (*string*) –
The AWS account ID of the owner of the instance.
- **NetworkInterfaceId** (*string*) –
The ID of the network interface.
- **VpcPeeringConnectionId** (*string*) –
The ID of the VPC peering connection.
- **State** (*string*) –
The state of the route. The `blackhole` state indicates that the route's target isn't available (for example, the specified gateway isn't attached to the VPC, or the specified NAT instance has been terminated).
- **Origin** (*string*) –
Describes how the route was created.
- `CreateRouteTable` indicates that route was automatically created when the route table was created.

- `CreateRoute` indicates that the route was manually added to the route table.
- `EnableVgwRoutePropagation` indicates that the route was propagated by route propagation.
- **Associations** (*list*) –
The associations between the route table and one or more subnets.
- (*dict*) –
Describes an association between a route table and a subnet.
- **RouteTableAssociationId** (*string*) –
The ID of the association between a route table and a subnet.
- **RouteTableId** (*string*) –
The ID of the route table.
- **SubnetId** (*string*) –
The ID of the subnet. A subnet ID is not returned for an implicit association.
- **Main** (*boolean*) –
Indicates whether this is the main route table.
- **Tags** (*list*) –
Any tags assigned to the route table.
- (*dict*) –
Describes a tag.
- **Key** (*string*) –
The key of the tag.
Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`
- **Value** (*string*) –
The value of the tag.
Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.
- **PropagatingVgws** (*list*) –
Any virtual private gateway (VGW) propagating routes.
- (*dict*) –
Describes a virtual private gateway propagating route.
- **GatewayId** (*string*) –
The ID of the virtual private gateway (VGW).

describe_security_groups (***kwargs*)

Describes one or more of your security groups.

A security group is for use with instances either in the EC2-Classical platform or in a specific VPC. For more information, see [Amazon EC2 Security Groups](#) in the *Amazon Elastic Compute Cloud User Guide* and [Security Groups for Your VPC](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```

response = client.describe_security_groups(
    DryRun=True|False,
    GroupNames=[
        'string',
    ],
    GroupIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **GroupNames** (*list*) – [EC2-Classic and default VPC only] One or more security group names. You can specify either the security group name or the security group ID. For security groups in a nondefault VPC, use the `group-name` filter to describe security groups by name.

Default: Describes all your security groups.

– (*string*) –

- **GroupIds** (*list*) – One or more security group IDs. Required for security groups in a nondefault VPC.

Default: Describes all your security groups.

– (*string*) –

- **Filters** (*list*) – One or more filters. If using multiple filters for rules, the results include security groups for which any combination of rules - not necessarily a single rule - match all filters.

- `description` - The description of the security group.
- `egress.ip-permission.prefix-list-id` - The ID (prefix) of the AWS service to which the security group allows access.
- `group-id` - The ID of the security group.
- `group-name` - The name of the security group.
- `ip-permission.cidr` - A CIDR range that has been granted permission.
- `ip-permission.from-port` - The start of port range for the TCP and UDP protocols, or an ICMP type number.
- `ip-permission.group-id` - The ID of a security group that has been granted permission.
- `ip-permission.group-name` - The name of a security group that has been granted permission.
- `ip-permission.protocol` - The IP protocol for the permission (`tcp` | `udp` | `icmp` or a protocol number).
- `ip-permission.to-port` - The end of port range for the TCP and UDP protocols, or an ICMP code.
- `ip-permission.user-id` - The ID of an AWS account that has been

granted permission.

- `owner-id` - The AWS account ID of the owner of the security group.
- `tag-key` - The key of a tag assigned to the security group.
- `tag-value` - The value of a tag assigned to the security group.
- `vpc-id` - The ID of the VPC specified when the security group was created.

- (*dict*) -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

Return type dict

Returns

Response Syntax

```
{
  'SecurityGroups': [
    {
      'OwnerId': 'string',
      'GroupName': 'string',
      'GroupId': 'string',
      'Description': 'string',
      'IpPermissions': [
        {
          'IpProtocol': 'string',
          'FromPort': 123,
          'ToPort': 123,
          'UserIdGroupPairs': [
            {
              'UserId': 'string',
              'GroupName': 'string',
              'GroupId': 'string'
            },
          ],
          'IpRanges': [
            {
              'CidrIp': 'string'
            },
          ],
          'PrefixListIds': [
            {
              'PrefixListId': 'string'
            },
          ],
        },
      ],
      'IpPermissionsEgress': [
        {
          'IpProtocol': 'string',
          'FromPort': 123,
```

```

        'ToPort': 123,
        'UserIdGroupPairs': [
            {
                'UserId': 'string',
                'GroupName': 'string',
                'GroupId': 'string'
            },
        ],
        'IpRanges': [
            {
                'CidrIp': 'string'
            },
        ],
        'PrefixListIds': [
            {
                'PrefixListId': 'string'
            },
        ],
    ],
    'VpcId': 'string',
    'Tags': [
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
],
},
}

```

Response Structure

- (*dict*) –
 - **SecurityGroups** (*list*) –

Information about one or more security groups.

- * (*dict*) –

Describes a security group

- **OwnerId** (*string*) –
The AWS account ID of the owner of the security group.
- **GroupName** (*string*) –
The name of the security group.
- **GroupId** (*string*) –
The ID of the security group.
- **Description** (*string*) –
A description of the security group.
- **IpPermissions** (*list*) –
One or more inbound rules associated with the security group.
- (*dict*) –
Describes a security group rule.
- **IpProtocol** (*string*) –
The protocol.

When you call `DescribeSecurityGroups`, the protocol value returned is the number. Exception: For TCP, UDP, and ICMP, the value returned is the name (for example, `tcp`, `udp`, or `icmp`). For a list of protocol numbers, see [Protocol Numbers](#). (VPC only) When you call `AuthorizeSecurityGroupIngress`, you can use `-1` to specify all.

- **FromPort** (*integer*) –

The start of port range for the TCP and UDP protocols, or an ICMP type number. A value of `-1` indicates all ICMP types.

- **ToPort** (*integer*) –

The end of port range for the TCP and UDP protocols, or an ICMP code. A value of `-1` indicates all ICMP codes for the specified ICMP type.

- **UserIdGroupPairs** (*list*) –

One or more security group and AWS account ID pairs.

- (*dict*) –

Describes a security group and AWS account ID pair.

- **UserId** (*string*) –

The ID of an AWS account. EC2-Classic only.

- **GroupName** (*string*) –

The name of the security group. In a request, use this parameter for a security group in EC2-Classic or a default VPC only. For a security group in a nondefault VPC, use `GroupId`.

- **GroupId** (*string*) –

The ID of the security group.

- **IpRanges** (*list*) –

One or more IP ranges.

- (*dict*) –

Describes an IP range.

- **CidrIp** (*string*) –

The CIDR range. You can either specify a CIDR range or a source security group, not both.

- **PrefixListIds** (*list*) –

(Valid for `AuthorizeSecurityGroupEgress`, `RevokeSecurityGroupEgress` and `DescribeSecurityGroups` only) One or more prefix list IDs for an AWS service. In an `AuthorizeSecurityGroupEgress` request, this is the AWS service that you want to access through a VPC endpoint from instances associated with the security group.

- (*dict*) –

The ID of the prefix.

- **PrefixListId** (*string*) –

The ID of the prefix.

- **IpPermissionsEgress** (*list*) –

[EC2-VPC] One or more outbound rules associated with the security group.

- *(dict)* –
Describes a security group rule.
- **IpProtocol** (*string*) –
The protocol.

When you call `DescribeSecurityGroups`, the protocol value returned is the number. Exception: For TCP, UDP, and ICMP, the value returned is the name (for example, `tcp`, `udp`, or `icmp`). For a list of protocol numbers, see [Protocol Numbers](#).
(VPC only) When you call `AuthorizeSecurityGroupIngress`, you can use `-1` to specify all.
- **FromPort** (*integer*) –
The start of port range for the TCP and UDP protocols, or an ICMP type number. A value of `-1` indicates all ICMP types.
- **ToPort** (*integer*) –
The end of port range for the TCP and UDP protocols, or an ICMP code. A value of `-1` indicates all ICMP codes for the specified ICMP type.
- **UserIdGroupPairs** (*list*) –
One or more security group and AWS account ID pairs.
- *(dict)* –
Describes a security group and AWS account ID pair.
- **UserId** (*string*) –
The ID of an AWS account. EC2-Classic only.
- **GroupName** (*string*) –
The name of the security group. In a request, use this parameter for a security group in EC2-Classic or a default VPC only. For a security group in a nondefault VPC, use `GroupId`.
- **GroupId** (*string*) –
The ID of the security group.
- **IpRanges** (*list*) –
One or more IP ranges.
- *(dict)* –
Describes an IP range.
- **CidrIp** (*string*) –
The CIDR range. You can either specify a CIDR range or a source security group, not both.
- **PrefixListIds** (*list*) –
(Valid for `AuthorizeSecurityGroupEgress`, `RevokeSecurityGroupEgress` and `DescribeSecurityGroups` only) One or more prefix list IDs for an AWS service. In an `AuthorizeSecurityGroupEgress` request, this is the AWS service that you want to access through a VPC endpoint from instances associated with the security group.
- *(dict)* –
The ID of the prefix.

- **PrefixListId** (*string*) –
The ID of the prefix.
- **VpcId** (*string*) –
[EC2-VPC] The ID of the VPC for the security group.
- **Tags** (*list*) –
Any tags assigned to the security group.
- (*dict*) –
Describes a tag.
- **Key** (*string*) –
The key of the tag.
Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`
- **Value** (*string*) –
The value of the tag.
Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

describe_snapshot_attribute (***kwargs*)

Describes the specified attribute of the specified snapshot. You can specify only one attribute at a time.

For more information about EBS snapshots, see [Amazon EBS Snapshots](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.describe_snapshot_attribute(  
    DryRun=True|False,  
    SnapshotId='string',  
    Attribute='productCodes'|'createVolumePermission'  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SnapshotId** (*string*) – **[REQUIRED]**
The ID of the EBS snapshot.
- **Attribute** (*string*) – **[REQUIRED]**
The snapshot attribute you would like to view.

Return type dict

Returns

Response Syntax

```
{  
    'SnapshotId': 'string',  
    'CreateVolumePermissions': [  
        {  
            'UserId': 'string',  
            'Group': 'all'  
        },  
    ],  
}
```



```

    ],
    'ProductCodes': [
        {
            'ProductCodeId': 'string',
            'ProductCodeType': 'devpay'|'marketplace'
        },
    ]
}

```

Response Structure

- *(dict)* –
 - **SnapshotId** (*string*) –
The ID of the EBS snapshot.
 - **CreateVolumePermissions** (*list*) –
A list of permissions for creating volumes from the snapshot.
 - * *(dict)* –
Describes the user or group to be added or removed from the permissions for a volume.
 - **UserId** (*string*) –
The specific AWS account ID that is to be added or removed from a volume’s list of create volume permissions.
 - **Group** (*string*) –
The specific group that is to be added or removed from a volume’s list of create volume permissions.
 - **ProductCodes** (*list*) –
A list of product codes.
 - * *(dict)* –
Describes a product code.
 - **ProductCodeId** (*string*) –
The product code.
 - **ProductCodeType** (*string*) –
The type of product code.

describe_snapshots (***kwargs*)

Describes one or more of the EBS snapshots available to you. Available snapshots include public snapshots available for any AWS account to launch, private snapshots that you own, and private snapshots owned by another AWS account but for which you’ve been given explicit create volume permissions.

The create volume permissions fall into the following categories:

- *public* : The owner of the snapshot granted create volume permissions for the snapshot to the all group. All AWS accounts have create volume permissions for these snapshots.
- *explicit* : The owner of the snapshot granted create volume permissions to a specific AWS account.
- *implicit* : An AWS account has implicit create volume permissions for all snapshots it owns.

The list of snapshots returned can be modified by specifying snapshot IDs, snapshot owners, or AWS accounts with create volume permissions. If no options are specified, Amazon EC2 returns all snapshots for which you have create volume permissions.

If you specify one or more snapshot IDs, only snapshots that have the specified IDs are returned. If you specify an invalid snapshot ID, an error is returned. If you specify a snapshot ID for which you do not have access, it is not included in the returned results.

If you specify one or more snapshot owners, only snapshots from the specified owners and for which you have access are returned. The results can include the AWS account IDs of the specified owners, `amazon` for snapshots owned by Amazon, or `self` for snapshots that you own.

If you specify a list of restorable users, only snapshots with create snapshot permissions for those users are returned. You can specify AWS account IDs (if you own the snapshots), `self` for snapshots for which you own or have explicit permissions, or `all` for public snapshots.

If you are describing a long list of snapshots, you can paginate the output to make the list more manageable. The `MaxResults` parameter sets the maximum number of results returned in a single page. If the list of results exceeds your `MaxResults` value, then that number of results is returned along with a `NextToken` value that can be passed to a subsequent `DescribeSnapshots` request to retrieve the remaining results.

For more information about EBS snapshots, see [Amazon EBS Snapshots](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.describe_snapshots(
    DryRun=True|False,
    SnapshotIds=[
        'string',
    ],
    OwnerIds=[
        'string',
    ],
    RestorableByUserIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SnapshotIds** (*list*) – One or more snapshot IDs.
Default: Describes snapshots for which you have launch permissions.
 - (*string*) –
- **OwnerIds** (*list*) – Returns the snapshots owned by the specified owner. Multiple owners can be specified.
 - (*string*) –
- **RestorableByUserIds** (*list*) – One or more AWS accounts IDs that can create volumes from the snapshot.
 - (*string*) –
- **Filters** (*list*) – One or more filters.

- `description` - A description of the snapshot.
- `owner-alias` - The AWS account alias (for example, `amazon`) that owns the snapshot.
- `owner-id` - The ID of the AWS account that owns the snapshot.
- `progress` - The progress of the snapshot, as a percentage (for example, 80%).
- `snapshot-id` - The snapshot ID.
- `start-time` - The time stamp when the snapshot was initiated.
- `status` - The status of the snapshot (`pending` | `completed` | `error`).
- `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag’s value is), and the tag value `X` (regardless of what the tag’s key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `volume-id` - The ID of the volume the snapshot is for.
- `volume-size` - The size of the volume, in GiB.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

- **NextToken** (*string*) - The `NextToken` value returned from a previous paginated `DescribeSnapshots` request where `MaxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `NextToken` value. This value is `null` when there are no more results to return.
- **MaxResults** (*integer*) - The maximum number of snapshot results returned by `DescribeSnapshots` in paginated output. When this parameter is used, `DescribeSnapshots` only returns `MaxResults` results in a single page along with a `NextToken` response element. The remaining results of the initial request can be seen by sending another `DescribeSnapshots` request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. If this parameter is not used, then `DescribeSnapshots` returns all results. You cannot specify this parameter and the `snapshot IDs` parameter in the same request.

Return type dict

Returns

Response Syntax

```
{
  'Snapshots': [
    {
      'SnapshotId': 'string',
      'VolumeId': 'string',
      'State': 'pending'|'completed'|'error',
      'StateMessage': 'string',
      'StartTime': datetime(2015, 1, 1),
      'Progress': 'string',
      'OwnerId': 'string',
      'Description': 'string',
      'VolumeSize': 123,
      'OwnerAlias': 'string',
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        },
      ],
      'Encrypted': True|False,
      'KmsKeyId': 'string',
      'DataEncryptionKeyId': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **Snapshots** (*list*) –

Information about the snapshots.

- * (*dict*) –

Describes a snapshot.

- **SnapshotId** (*string*) –

The ID of the snapshot. Each snapshot receives a unique identifier when it is created.
- **VolumeId** (*string*) –

The ID of the volume that was used to create the snapshot.
- **State** (*string*) –

The snapshot state.
- **StateMessage** (*string*) –

Encrypted Amazon EBS snapshots are copied asynchronously. If a snapshot copy operation fails (for example, if the proper AWS Key Management Service (AWS KMS) permissions are not obtained) this field displays error state details to help you diagnose why the error occurred. This parameter is only returned by the DescribeSnapshots API operation.
- **StartTime** (*datetime*) –

The time stamp when the snapshot was initiated.
- **Progress** (*string*) –

The progress of the snapshot, as a percentage.

- **OwnerId** (*string*) –
The AWS account ID of the EBS snapshot owner.
- **Description** (*string*) –
The description for the snapshot.
- **VolumeSize** (*integer*) –
The size of the volume, in GiB.
- **OwnerAlias** (*string*) –
The AWS account alias (for example, `amazon`, `self`) or AWS account ID that owns the snapshot.
- **Tags** (*list*) –
Any tags assigned to the snapshot.
- (*dict*) –
Describes a tag.
- **Key** (*string*) –
The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws` :
- **Value** (*string*) –
The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.
- **Encrypted** (*boolean*) –
Indicates whether the snapshot is encrypted.
- **KmsKeyId** (*string*) –
The full ARN of the AWS Key Management Service (AWS KMS) customer master key (CMK) that was used to protect the volume encryption key for the parent volume.
- **DataEncryptionKeyId** (*string*) –
The data encryption key identifier for the snapshot. This value is a unique identifier that corresponds to the data encryption key that was used to encrypt the original volume or snapshot copy. Because data encryption keys are inherited by volumes created from snapshots, and vice versa, if snapshots share the same data encryption key identifier, then they belong to the same volume/snapshot lineage. This parameter is only returned by the `DescribeSnapshots` API operation.

– **NextToken** (*string*) –

The `NextToken` value to include in a future `DescribeSnapshots` request. When the results of a `DescribeSnapshots` request exceed `MaxResults`, this value can be used to retrieve the next page of results. This value is `null` when there are no more results to return.

describe_spot_datafeed_subscription (***kwargs*)

Describes the data feed for Spot instances. For more information, see [Spot Instance Data Feed](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.describe_spot_datafeed_subscription(
    DryRun=True|False
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns

Response Syntax

```
{
  'SpotDatafeedSubscription': {
    'OwnerId': 'string',
    'Bucket': 'string',
    'Prefix': 'string',
    'State': 'Active'|'Inactive',
    'Fault': {
      'Code': 'string',
      'Message': 'string'
    }
  }
}
```

Response Structure

- (*dict*) –

Contains the output of `DescribeSpotDatafeedSubscription`.

- **SpotDatafeedSubscription** (*dict*) –

The Spot instance data feed subscription.

- * **OwnerId** (*string*) –

The AWS account ID of the account.

- * **Bucket** (*string*) –

The Amazon S3 bucket where the Spot instance data feed is located.

- * **Prefix** (*string*) –

The prefix that is prepended to data feed files.

- * **State** (*string*) –

The state of the Spot instance data feed subscription.

- * **Fault** (*dict*) –

The fault codes for the Spot instance request, if any.

- **Code** (*string*) –

The reason code for the Spot instance state change.

- **Message** (*string*) –

The message for the Spot instance state change.

describe_spot_fleet_instances (***kwargs*)

Describes the running instances for the specified Spot fleet.

Request Syntax

```
response = client.describe_spot_fleet_instances(
    DryRun=True|False,
    SpotFleetRequestId='string',
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SpotFleetRequestId** (*string*) – **[REQUIRED]**
The ID of the Spot fleet request.
- **NextToken** (*string*) – The token for the next set of results.
- **MaxResults** (*integer*) – The maximum number of results to return in a single call. Specify a value between 1 and 1000. The default value is 1000. To retrieve the remaining results, make another call with the returned `NextToken` value.

Return type dict

Returns

Response Syntax

```
{
  'SpotFleetRequestId': 'string',
  'ActiveInstances': [
    {
      'InstanceType': 'string',
      'InstanceId': 'string',
      'SpotInstanceRequestId': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output of `DescribeSpotFleetInstances`.

- **SpotFleetRequestId** (*string*) –

The ID of the Spot fleet request.

- **ActiveInstances** (*list*) –

The running instances. Note that this list is refreshed periodically and might be out of date.

- * (*dict*) –

Describes a running instance in a Spot fleet.

- **InstanceType** (*string*) –

The instance type.

- **InstanceId** (*string*) –

The ID of the instance.

- **SpotInstanceRequestId** (*string*) –

The ID of the Spot instance request.

– **NextToken** (*string*) –

The token required to retrieve the next set of results. This value is `null` when there are no more results to return.

describe_spot_fleet_request_history (***kwargs*)

Describes the events for the specified Spot fleet request during the specified time.

Spot fleet events are delayed by up to 30 seconds before they can be described. This ensures that you can query by the last evaluated time and not miss a recorded event.

Request Syntax

```
response = client.describe_spot_fleet_request_history(
    DryRun=True|False,
    SpotFleetRequestId='string',
    EventType='instanceChange'|'fleetRequestChange'|'error',
    StartTime=datetime(2015, 1, 1),
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SpotFleetRequestId** (*string*) – **[REQUIRED]**
The ID of the Spot fleet request.
- **EventType** (*string*) – The type of events to describe. By default, all events are described.
- **StartTime** (*datetime*) – **[REQUIRED]**
The starting date and time for the events, in UTC format (for example, `YYYY-MM-DD T*HH*:MM:SS Z`).
- **NextToken** (*string*) – The token for the next set of results.
- **MaxResults** (*integer*) – The maximum number of results to return in a single call. Specify a value between 1 and 1000. The default value is 1000. To retrieve the remaining results, make another call with the returned `NextToken` value.

Return type dict

Returns

Response Syntax

```
{
  'SpotFleetRequestId': 'string',
  'StartTime': datetime(2015, 1, 1),
  'LastEvaluatedTime': datetime(2015, 1, 1),
  'HistoryRecords': [
    {
      'Timestamp': datetime(2015, 1, 1),
      'EventType': 'instanceChange'|'fleetRequestChange'|'error',
      'EventInformation': {
        'InstanceId': 'string',
        'EventSubType': 'string',
        'EventDescription': 'string'
      }
    }
  ],
}
```



```

],
  'NextToken': 'string'
}

```

Response Structure

- (*dict*) –

Contains the output of DescribeSpotFleetRequestHistory.

- **SpotFleetRequestId** (*string*) –

The ID of the Spot fleet request.

- **StartTime** (*datetime*) –

The starting date and time for the events, in UTC format (for example, `YYYY-MM-DD T*HH* :MM :SS Z`).

- **LastEvaluatedTime** (*datetime*) –

The last date and time for the events, in UTC format (for example, `YYYY-MM-DD T*HH* :MM :SS Z`). All records up to this time were retrieved.

If `nextToken` indicates that there are more results, this value is not present.

- **HistoryRecords** (*list*) –

Information about the events in the history of the Spot fleet request.

- * (*dict*) –

Describes an event in the history of the Spot fleet request.

- **Timestamp** (*datetime*) –

The date and time of the event, in UTC format (for example, `YYYY-MM-DD T*HH* :MM :SS Z`).

- **EventType** (*string*) –

The event type.

- `error` - Indicates an error with the Spot fleet request.
- `fleetRequestChange` - Indicates a change in the status or configuration of the Spot fleet request.
- `instanceChange` - Indicates that an instance was launched or terminated.
- **EventInformation** (*dict*) –

Information about the event.

- **InstanceId** (*string*) –

The ID of the instance. This information is available only for `instanceChange` events.

- **EventSubType** (*string*) –

The event.

The following are the error events.

- `iamFleetRoleInvalid` - Spot fleet did not have the required permissions either to launch or terminate an instance.
- `spotFleetRequestConfigurationInvalid` - The configuration is not valid. For more information, see the description.
- `spotInstanceCountLimitExceeded` - You've reached the limit on the number of Spot instances that you can launch.

The following are the `fleetRequestChange` events.

- `active` - The Spot fleet has been validated and Amazon EC2 is attempting to maintain the target number of running Spot instances.
- `cancelled` - The Spot fleet is canceled and has no running Spot instances. The Spot fleet will be deleted two days after its instances were terminated.
- `cancelled_running` - The Spot fleet is canceled and will not launch additional Spot instances, but its existing Spot instances continue to run until they are interrupted or terminated.
- `cancelled_terminating` - The Spot fleet is canceled and its Spot instances are terminating.
- `expired` - The Spot fleet request has expired. A subsequent event indicates that the instances were terminated, if the request was created with `TerminateInstancesWithExpiration` set.
- `price_update` - The bid price for a launch configuration was adjusted because it was too high. This change is permanent.
- `submitted` - The Spot fleet request is being evaluated and Amazon EC2 is preparing to launch the target number of Spot instances.

The following are the `instanceChange` events.

- `launched` - A bid was fulfilled and a new instance was launched.
- `terminated` - An instance was terminated by the user.
- **EventDescription** (*string*) –

The description of the event.

- **NextToken** (*string*) –

The token required to retrieve the next set of results. This value is `null` when there are no more results to return.

describe_spot_fleet_requests (***kwargs*)

Describes your Spot fleet requests.

Request Syntax

```
response = client.describe_spot_fleet_requests(
    DryRun=True|False,
    SpotFleetRequestIds=[
        'string',
    ],
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SpotFleetRequestIds** (*list*) – The IDs of the Spot fleet requests.
 - (*string*) –
- **NextToken** (*string*) – The token for the next set of results.

- **MaxResults** (*integer*) – The maximum number of results to return in a single call. Specify a value between 1 and 1000. The default value is 1000. To retrieve the remaining results, make another call with the returned NextToken value.

Return type dict

Returns

Response Syntax

```
{
  'SpotFleetRequestConfigs': [
    {
      'SpotFleetRequestId': 'string',
      'SpotFleetRequestState': 'submitted'|'active'|'cancelled'|'failed'|'cancelled_with_errors',
      'SpotFleetRequestConfig': {
        'ClientToken': 'string',
        'SpotPrice': 'string',
        'TargetCapacity': 123,
        'ValidFrom': datetime(2015, 1, 1),
        'ValidUntil': datetime(2015, 1, 1),
        'TerminateInstancesWithExpiration': True|False,
        'IamFleetRole': 'string',
        'LaunchSpecifications': [
          {
            'ImageId': 'string',
            'KeyName': 'string',
            'SecurityGroups': [
              {
                'GroupName': 'string',
                'GroupId': 'string'
              },
            ],
            'UserData': 'string',
            'AddressingType': 'string',
            'InstanceType': 't1.micro'|'m1.small'|'m1.medium'|'m1.large',
            'Placement': {
              'AvailabilityZone': 'string',
              'GroupName': 'string'
            },
            'KernelId': 'string',
            'RamdiskId': 'string',
            'BlockDeviceMappings': [
              {
                'VirtualName': 'string',
                'DeviceName': 'string',
                'Ebs': {
                  'SnapshotId': 'string',
                  'VolumeSize': 123,
                  'DeleteOnTermination': True|False,
                  'VolumeType': 'standard'|'io1'|'gp2',
                  'Iops': 123,
                  'Encrypted': True|False
                },
                'NoDevice': 'string'
              },
            ],
            'Monitoring': {
              'Enabled': True|False
            },
          },
        ],
      },
    },
  ],
}
```

```
        'SubnetId': 'string',
        'NetworkInterfaces': [
            {
                'NetworkInterfaceId': 'string',
                'DeviceIndex': 123,
                'SubnetId': 'string',
                'Description': 'string',
                'PrivateIpAddress': 'string',
                'Groups': [
                    'string',
                ],
                'DeleteOnTermination': True|False,
                'PrivateIpAddresses': [
                    {
                        'PrivateIpAddress': 'string',
                        'Primary': True|False
                    },
                ],
                'SecondaryPrivateIpAddressCount': 123,
                'AssociatePublicIpAddress': True|False
            },
        ],
        'IamInstanceProfile': {
            'Arn': 'string',
            'Name': 'string'
        },
        'EbsOptimized': True|False,
        'WeightedCapacity': 123.0,
        'SpotPrice': 'string'
    },
    ],
    'AllocationStrategy': 'lowestPrice'|'diversified'
}
},
],
'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output of DescribeSpotFleetRequests.

- **SpotFleetRequestConfigs** (*list*) –

Information about the configuration of your Spot fleet.

- * (*dict*) –

Describes a Spot fleet request.

- **SpotFleetRequestId** (*string*) –

The ID of the Spot fleet request.

- **SpotFleetRequestState** (*string*) –

The state of the Spot fleet request.

- **SpotFleetRequestConfig** (*dict*) –

Information about the configuration of the Spot fleet request.

- **ClientToken** (*string*) –

A unique, case-sensitive identifier you provide to ensure

idempotency of your listings. This helps avoid duplicate listings. For more information, see [Ensuring Idempotency](#).

- **SpotPrice** (*string*) –

The bid price per unit hour.

- **TargetCapacity** (*integer*) –

The number of units to request. You can choose to set the target capacity in terms of instances or a performance characteristic that is important to your application workload, such as vCPUs, memory, or I/O.

- **ValidFrom** (*datetime*) –

The start date and time of the request, in UTC format (for example, `YYYY-MM-DD T*HH* :MM :SS Z`). The default is to start fulfilling the request immediately.

- **ValidUntil** (*datetime*) –

The end date and time of the request, in UTC format (for example, `YYYY-MM-DD T*HH* :MM :SS Z`). At this point, no new Spot instance requests are placed or enabled to fulfill the request.

- **TerminateInstancesWithExpiration** (*boolean*) –

Indicates whether running Spot instances should be terminated when the Spot fleet request expires.

- **IamFleetRole** (*string*) –

Grants the Spot fleet permission to terminate Spot instances on your behalf when you cancel its Spot fleet request using `CancelSpotFleetRequests` or when the Spot fleet request expires, if you set `terminateInstancesWithExpiration`.

- **LaunchSpecifications** (*list*) –

Information about the launch specifications for the Spot fleet request.

- (*dict*) –

Describes the launch specification for one or more Spot instances.

- **ImageId** (*string*) –

The ID of the AMI.

- **KeyName** (*string*) –

The name of the key pair.

- **SecurityGroups** (*list*) –

One or more security groups. To request an instance in a non-default VPC, you must specify the ID of the security group. To request an instance in EC2-Classical or a default VPC, you can specify the name or the ID of the security group.

- (*dict*) –

Describes a security group.

- **GroupName** (*string*) –

The name of the security group.

- **GroupId** (*string*) –

- The ID of the security group.
- **UserData** (*string*) –
The Base64-encoded MIME user data to make available to the instances.
- **AddressingType** (*string*) –
Deprecated.
- **InstanceType** (*string*) –
The instance type.
- **Placement** (*dict*) –
The placement information.
- **AvailabilityZone** (*string*) –
The Availability Zone.
- **GroupName** (*string*) –
The name of the placement group (for cluster instances).
- **KernelId** (*string*) –
The ID of the kernel.
- **RamdiskId** (*string*) –
The ID of the RAM disk.
- **BlockDeviceMappings** (*list*) –
One or more block device mapping entries.
- (*dict*) –
Describes a block device mapping.
- **VirtualName** (*string*) –
The virtual device name (ephemeral N). Instance store volumes are numbered starting from 0. An instance type with 2 available instance store volumes can specify mappings for `ephemeral0` and `ephemeral1`. The number of available instance store volumes depends on the instance type. After you connect to the instance, you must mount the volume.

Constraints: For M3 instances, you must specify instance store volumes in the block device mapping for the instance. When you launch an M3 instance, we ignore any instance store volumes specified in the block device mapping for the AMI.
- **DeviceName** (*string*) –
The device name exposed to the instance (for example, `/dev/sdh` or `xvdh`).
- **Ebs** (*dict*) –
Parameters used to automatically set up EBS volumes when the instance is launched.
- **SnapshotId** (*string*) –
The ID of the snapshot.
- **VolumeSize** (*integer*) –
The size of the volume, in GiB.

Constraints: 1-1024 for standard volumes, 1-16384 for gp2 volumes, and 4-16384 for io1 volumes. If you specify a snapshot, the volume size must be equal to or larger than the snapshot size.

Default: If you're creating the volume from a snapshot and don't specify a volume size, the default is the snapshot size.

- **DeleteOnTermination** (*boolean*) –

Indicates whether the EBS volume is deleted on instance termination.

- **VolumeType** (*string*) –

The volume type. gp2 for General Purpose (SSD) volumes, io1 for Provisioned IOPS (SSD) volumes, and standard for Magnetic volumes.

Default: standard

- **Iops** (*integer*) –

The number of I/O operations per second (IOPS) that the volume supports. For Provisioned IOPS (SSD) volumes, this represents the number of IOPS that are provisioned for the volume. For General Purpose (SSD) volumes, this represents the baseline performance of the volume and the rate at which the volume accumulates I/O credits for bursting. For more information on General Purpose (SSD) baseline performance, I/O credits, and bursting, see [Amazon EBS Volume Types](#) in the *Amazon Elastic Compute Cloud User Guide*.

Constraint: Range is 100 to 20000 for Provisioned IOPS (SSD) volumes and 3 to 10000 for General Purpose (SSD) volumes.

Condition: This parameter is required for requests to create io1 volumes; it is not used in requests to create standard or gp2 volumes.

- **Encrypted** (*boolean*) –

Indicates whether the EBS volume is encrypted. Encrypted Amazon EBS volumes may only be attached to instances that support Amazon EBS encryption.

- **NoDevice** (*string*) –

Suppresses the specified device included in the block device mapping of the AMI.

- **Monitoring** (*dict*) –

Enable or disable monitoring for the instances.

- **Enabled** (*boolean*) –

Enables monitoring for the instance.

Default: false

- **SubnetId** (*string*) –

The ID of the subnet in which to launch the instances.

- **NetworkInterfaces** (*list*) –

One or more network interfaces.

- *(dict)* –
Describes a network interface.
- **NetworkInterfaceId** (*string*) –
The ID of the network interface.
- **DeviceIndex** (*integer*) –
The index of the device on the instance for the network interface attachment. If you are specifying a network interface in a RunInstances request, you must provide the device index.
- **SubnetId** (*string*) –
The ID of the subnet associated with the network string. Applies only if creating a network interface when launching an instance.
- **Description** (*string*) –
The description of the network interface. Applies only if creating a network interface when launching an instance.
- **PrivateIpAddress** (*string*) –
The private IP address of the network interface. Applies only if creating a network interface when launching an instance.
- **Groups** (*list*) –
The IDs of the security groups for the network interface. Applies only if creating a network interface when launching an instance.
- (*string*) –
- **DeleteOnTermination** (*boolean*) –
If set to `true`, the interface is deleted when the instance is terminated. You can specify `true` only if creating a new network interface when launching an instance.
- **PrivateIpAddresses** (*list*) –
One or more private IP addresses to assign to the network interface. Only one private IP address can be designated as primary.
- (*dict*) –
Describes a secondary private IP address for a network interface.
- **PrivateIpAddress** (*string*) –
The private IP addresses.
- **Primary** (*boolean*) –
Indicates whether the private IP address is the primary private IP address. Only one IP address can be designated as primary.
- **SecondaryPrivateIpAddressCount** (*integer*) –
The number of secondary private IP addresses. You can't specify this option and specify more than one private IP address using the private IP addresses option.
- **AssociatePublicIpAddress** (*boolean*) –
Indicates whether to assign a public IP address to an instance you launch in a VPC. The public IP address can only be assigned to a network interface for `eth0`, and can only be as-

signed to a new network interface, not an existing one. You cannot specify more than one network interface in the request. If launching into a default subnet, the default value is `true`.

- **IamInstanceProfile** (*dict*) –

The IAM instance profile.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) of the instance profile.

- **Name** (*string*) –

The name of the instance profile.

- **EbsOptimized** (*boolean*) –

Indicates whether the instances are optimized for EBS I/O. This optimization provides dedicated throughput to Amazon EBS and an optimized configuration stack to provide optimal EBS I/O performance. This optimization isn't available with all instance types. Additional usage charges apply when using an EBS Optimized instance.

Default: `false`

- **WeightedCapacity** (*float*) –

The number of units provided by the specified instance type. These are the same units that you chose to set the target capacity in terms (instances or a performance characteristic such as vCPUs, memory, or I/O).

If the target capacity divided by this value is not a whole number, we round the number of instances to the next whole number. If this value is not specified, the default is 1.

- **SpotPrice** (*string*) –

The bid price per unit hour for the specified instance type. If this value is not specified, the default is the Spot bid price specified for the fleet. To determine the bid price per unit hour, divide the Spot bid price by the value of `WeightedCapacity`.

- **AllocationStrategy** (*string*) –

Determines how to allocate the target capacity across the Spot pools specified by the Spot fleet request. The default is `lowestPrice`.

- **NextToken** (*string*) –

The token required to retrieve the next set of results. This value is `null` when there are no more results to return.

describe_spot_instance_requests (***kwargs*)

Describes the Spot instance requests that belong to your account. Spot instances are instances that Amazon EC2 launches when the bid price that you specify exceeds the current Spot price. Amazon EC2 periodically sets the Spot price based on available Spot instance capacity and current Spot instance requests. For more information, see [Spot Instance Requests](#) in the *Amazon Elastic Compute Cloud User Guide*.

You can use `DescribeSpotInstanceRequests` to find a running Spot instance by examining the response. If the status of the Spot instance is `fulfilled`, the instance ID appears in the response and contains the identifier of the instance. Alternatively, you can use `DescribeInstances` with a filter to look for instances where the instance lifecycle is `spot`.

Request Syntax

```
response = client.describe_spot_instance_requests(
    DryRun=True|False,
    SpotInstanceRequestIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SpotInstanceRequestIds** (*list*) – One or more Spot instance request IDs.
 - (*string*) –
- **Filters** (*list*) – One or more filters.
 - `availability-zone-group` - The Availability Zone group.
 - `create-time` - The time stamp when the Spot instance request was created.
 - `fault-code` - The fault code related to the request.
 - `fault-message` - The fault message related to the request.
 - `instance-id` - The ID of the instance that fulfilled the request.
 - `launch-group` - The Spot instance launch group.
 - `launch.block-device-mapping.delete-on-termination` - Indicates whether the Amazon EBS volume is deleted on instance termination.
 - `launch.block-device-mapping.device-name` - The device name for the Amazon EBS volume (for example, `/dev/sdh`).
 - `launch.block-device-mapping.snapshot-id` - The ID of the snapshot used for the Amazon EBS volume.
 - `launch.block-device-mapping.volume-size` - The size of the Amazon EBS volume, in GiB.
 - `launch.block-device-mapping.volume-type` - The type of the Amazon EBS volume (`gp2` | `standard` | `io1`).
 - `launch.group-id` - The security group for the instance.
 - `launch.image-id` - The ID of the AMI.
 - `launch.instance-type` - The type of instance (for example, `m1.small`).
 - `launch.kernel-id` - The kernel ID.
 - `launch.key-name` - The name of the key pair the instance launched with.
 - `launch.monitoring-enabled` - Whether monitoring is enabled for the Spot instance.
 - `launch.ramdisk-id` - The RAM disk ID.
 - `network-interface.network-interface-id` - The ID of the network interface.

- `network-interface.device-index` - The index of the device for the network interface attachment on the instance.
- `network-interface.subnet-id` - The ID of the subnet for the instance.
- `network-interface.description` - A description of the network interface.
- `network-interface.private-ip-address` - The primary private IP address of the network interface.
- `network-interface.delete-on-termination` - Indicates whether the network interface is deleted when the instance is terminated.
- `network-interface.group-id` - The ID of the security group associated with the network interface.
- `network-interface.group-name` - The name of the security group associated with the network interface.
- `network-interface.addresses.primary` - Indicates whether the IP address is the primary private IP address.
- `product-description` - The product description associated with the instance (Linux/UNIX|Windows).
- `spot-instance-request-id` - The Spot instance request ID.
- `spot-price` - The maximum hourly price for any Spot instance launched to fulfill the request.
- `state` - The state of the Spot instance request (`open` | `active` | `closed` | `cancelled` | `failed`). Spot bid status information can help you track your Amazon EC2 Spot instance requests. For more information, see [Spot Bid Status](#) in the Amazon Elastic Compute Cloud User Guide.
- `status-code` - The short code describing the most recent evaluation of your Spot instance request.
- `status-message` - The message explaining the status of the Spot instance request.
- `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key Purpose (regardless of what the tag’s value is), and the tag value X (regardless of what the tag’s key is). If you want to list only resources where Purpose is X, see the `tag :key =*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `type` - The type of Spot instance request (`one-time` | `persistent`).
- `launched-availability-zone` - The Availability Zone in which the bid is launched.
- `valid-from` - The start date of the request.
- `valid-until` - The end date of the request.

- (*dict*) -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (string) –

Return type dict

Returns

Response Syntax

```
{
  'SpotInstanceRequests': [
    {
      'SpotInstanceRequestId': 'string',
      'SpotPrice': 'string',
      'Type': 'one-time'|'persistent',
      'State': 'open'|'active'|'closed'|'cancelled'|'failed',
      'Fault': {
        'Code': 'string',
        'Message': 'string'
      },
      'Status': {
        'Code': 'string',
        'UpdateTime': datetime(2015, 1, 1),
        'Message': 'string'
      },
      'ValidFrom': datetime(2015, 1, 1),
      'ValidUntil': datetime(2015, 1, 1),
      'LaunchGroup': 'string',
      'AvailabilityZoneGroup': 'string',
      'LaunchSpecification': {
        'ImageId': 'string',
        'KeyName': 'string',
        'SecurityGroups': [
          {
            'GroupName': 'string',
            'GroupId': 'string'
          },
        ],
      },
      'UserData': 'string',
      'AddressingType': 'string',
      'InstanceType': 't1.micro'|'m1.small'|'m1.medium'|'m1.large'|'m1.x',
      'Placement': {
        'AvailabilityZone': 'string',
        'GroupName': 'string'
      },
      'KernelId': 'string',
      'RamdiskId': 'string',
      'BlockDeviceMappings': [
        {
          'VirtualName': 'string',
          'DeviceName': 'string',
          'Ebs': {
            'SnapshotId': 'string',
            'VolumeSize': 123,
            'DeleteOnTermination': True|False,
            'VolumeType': 'standard'|'io1'|'gp2',
            'Iops': 123,
            'Encrypted': True|False
          },
          'NoDevice': 'string'
        },
      ],
    },
  ],
}
```

```

        },
    ],
    'SubnetId': 'string',
    'NetworkInterfaces': [
        {
            'NetworkInterfaceId': 'string',
            'DeviceIndex': 123,
            'SubnetId': 'string',
            'Description': 'string',
            'PrivateIpAddress': 'string',
            'Groups': [
                'string',
            ],
            'DeleteOnTermination': True|False,
            'PrivateIpAddresses': [
                {
                    'PrivateIpAddress': 'string',
                    'Primary': True|False
                },
            ],
            'SecondaryPrivateIpAddressCount': 123,
            'AssociatePublicIpAddress': True|False
        },
    ],
    'IamInstanceProfile': {
        'Arn': 'string',
        'Name': 'string'
    },
    'EbsOptimized': True|False,
    'Monitoring': {
        'Enabled': True|False
    }
},
'InstanceId': 'string',
'CreateTime': datetime(2015, 1, 1),
'ProductDescription': 'Linux/UNIX'|'Linux/UNIX (Amazon VPC)'|'Windows'
'Tags': [
    {
        'Key': 'string',
        'Value': 'string'
    },
],
'LaunchedAvailabilityZone': 'string'
},
]
}

```

Response Structure

- (*dict*) –

Contains the output of DescribeSpotInstanceRequests.

- **SpotInstanceRequests** (*list*) –

One or more Spot instance requests.

- * (*dict*) –

Describe a Spot instance request.

- **SpotInstanceRequestId** (*string*) –

The ID of the Spot instance request.

- **SpotPrice** (*string*) –

The maximum hourly price (bid) for any Spot instance launched to fulfill the request.

- **Type** (*string*) –

The Spot instance request type.

- **State** (*string*) –

The state of the Spot instance request. Spot bid status information can help you track your Spot instance requests. For more information, see [Spot Bid Status](#) in the *Amazon Elastic Compute Cloud User Guide*.

- **Fault** (*dict*) –

The fault codes for the Spot instance request, if any.

- **Code** (*string*) –

The reason code for the Spot instance state change.

- **Message** (*string*) –

The message for the Spot instance state change.

- **Status** (*dict*) –

The status code and status message describing the Spot instance request.

- **Code** (*string*) –

The status code.

- **UpdateTime** (*datetime*) –

The date and time of the most recent status update, in UTC format (for example, *YYYY-MM-DD T*HH* :MM :SS Z*).

- **Message** (*string*) –

The description for the status code.

- **ValidFrom** (*datetime*) –

The start date of the request, in UTC format (for example, *YYYY-MM-DD T*HH* :MM :SS Z*). If this is a one-time request, the request becomes active at this date and time and remains active until all instances launch, the request expires, or the request is canceled. If the request is persistent, the request becomes active at this date and time and remains active until it expires or is canceled.

- **ValidUntil** (*datetime*) –

The end date of the request, in UTC format (for example, *YYYY-MM-DD T*HH* :MM :SS Z*). If this is a one-time request, the request remains active until all instances launch, the request is canceled, or this date is reached. If the request is persistent, it remains active until it is canceled or this date is reached.

- **LaunchGroup** (*string*) –

The instance launch group. Launch groups are Spot instances that launch together and terminate together.

- **AvailabilityZoneGroup** (*string*) –

The Availability Zone group. If you specify the same Availability Zone group for all Spot instance requests, all Spot instances are launched in the same Availability Zone.

- **LaunchSpecification** (*dict*) –

Additional information for launching instances.

- **ImageId** (*string*) –

The ID of the AMI.

- **KeyName** (*string*) –

The name of the key pair.

- **SecurityGroups** (*list*) –

One or more security groups. To request an instance in a non-default VPC, you must specify the ID of the security group. To request an instance in EC2-Classic or a default VPC, you can specify the name or the ID of the security group.

- (*dict*) –

Describes a security group.

- **GroupName** (*string*) –

The name of the security group.

- **GroupId** (*string*) –

The ID of the security group.

- **UserData** (*string*) –

The Base64-encoded MIME user data to make available to the instances.

- **AddressingType** (*string*) –

Deprecated.

- **InstanceType** (*string*) –

The instance type.

- **Placement** (*dict*) –

The placement information for the instance.

- **AvailabilityZone** (*string*) –

The Availability Zone.

- **GroupName** (*string*) –

The name of the placement group (for cluster instances).

- **KernelId** (*string*) –

The ID of the kernel.

- **RamdiskId** (*string*) –

The ID of the RAM disk.

- **BlockDeviceMappings** (*list*) –

One or more block device mapping entries.

- (*dict*) –

Describes a block device mapping.

- **VirtualName** (*string*) –

The virtual device name (ephemeral N). Instance store volumes are numbered starting from 0. An instance type with 2 available instance store volumes can specify mappings for

`ephemeral0` and `ephemeral1`. The number of available instance store volumes depends on the instance type. After you connect to the instance, you must mount the volume.

Constraints: For M3 instances, you must specify instance store volumes in the block device mapping for the instance. When you launch an M3 instance, we ignore any instance store volumes specified in the block device mapping for the AMI.

- **DeviceName** (*string*) –

The device name exposed to the instance (for example, `/dev/sdh` or `xvdh`).

- **Ebs** (*dict*) –

Parameters used to automatically set up EBS volumes when the instance is launched.

- **SnapshotId** (*string*) –

The ID of the snapshot.

- **VolumeSize** (*integer*) –

The size of the volume, in GiB.

Constraints: 1–1024 for standard volumes, 1–16384 for gp2 volumes, and 4–16384 for io1 volumes. If you specify a snapshot, the volume size must be equal to or larger than the snapshot size.

Default: If you're creating the volume from a snapshot and don't specify a volume size, the default is the snapshot size.

- **DeleteOnTermination** (*boolean*) –

Indicates whether the EBS volume is deleted on instance termination.

- **VolumeType** (*string*) –

The volume type. `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, and `standard` for Magnetic volumes.

Default: `standard`

- **Iops** (*integer*) –

The number of I/O operations per second (IOPS) that the volume supports. For Provisioned IOPS (SSD) volumes, this represents the number of IOPS that are provisioned for the volume. For General Purpose (SSD) volumes, this represents the baseline performance of the volume and the rate at which the volume accumulates I/O credits for bursting. For more information on General Purpose (SSD) baseline performance, I/O credits, and bursting, see [Amazon EBS Volume Types](#) in the *Amazon Elastic Compute Cloud User Guide*.

Constraint: Range is 100 to 20000 for Provisioned IOPS (SSD) volumes and 3 to 10000 for General Purpose (SSD) volumes.

Condition: This parameter is required for requests to create `io1` volumes; it is not used in requests to create `standard`

or gp2 volumes.

- **Encrypted** (*boolean*) –

Indicates whether the EBS volume is encrypted. Encrypted Amazon EBS volumes may only be attached to instances that support Amazon EBS encryption.

- **NoDevice** (*string*) –

Suppresses the specified device included in the block device mapping of the AMI.

- **SubnetId** (*string*) –

The ID of the subnet in which to launch the instance.

- **NetworkInterfaces** (*list*) –

One or more network interfaces.

- (*dict*) –

Describes a network interface.

- **NetworkInterfaceId** (*string*) –

The ID of the network interface.

- **DeviceIndex** (*integer*) –

The index of the device on the instance for the network interface attachment. If you are specifying a network interface in a RunInstances request, you must provide the device index.

- **SubnetId** (*string*) –

The ID of the subnet associated with the network string. Applies only if creating a network interface when launching an instance.

- **Description** (*string*) –

The description of the network interface. Applies only if creating a network interface when launching an instance.

- **PrivateIpAddress** (*string*) –

The private IP address of the network interface. Applies only if creating a network interface when launching an instance.

- **Groups** (*list*) –

The IDs of the security groups for the network interface. Applies only if creating a network interface when launching an instance.

- (*string*) –

- **DeleteOnTermination** (*boolean*) –

If set to `true`, the interface is deleted when the instance is terminated. You can specify `true` only if creating a new network interface when launching an instance.

- **PrivateIpAddresses** (*list*) –

One or more private IP addresses to assign to the network interface. Only one private IP address can be designated as primary.

- (*dict*) –

Describes a secondary private IP address for a network interface.

- **PrivateIpAddress** (*string*) –
The private IP addresses.
- **Primary** (*boolean*) –
Indicates whether the private IP address is the primary private IP address. Only one IP address can be designated as primary.
- **SecondaryPrivateIpAddressCount** (*integer*) –
The number of secondary private IP addresses. You can't specify this option and specify more than one private IP address using the private IP addresses option.
- **AssociatePublicIpAddress** (*boolean*) –
Indicates whether to assign a public IP address to an instance you launch in a VPC. The public IP address can only be assigned to a network interface for eth0, and can only be assigned to a new network interface, not an existing one. You cannot specify more than one network interface in the request. If launching into a default subnet, the default value is `true`.
- **IamInstanceProfile** (*dict*) –
The IAM instance profile.
- **Arn** (*string*) –
The Amazon Resource Name (ARN) of the instance profile.
- **Name** (*string*) –
The name of the instance profile.
- **EbsOptimized** (*boolean*) –
Indicates whether the instance is optimized for EBS I/O. This optimization provides dedicated throughput to Amazon EBS and an optimized configuration stack to provide optimal EBS I/O performance. This optimization isn't available with all instance types. Additional usage charges apply when using an EBS Optimized instance.

Default: `false`
- **Monitoring** (*dict*) –
Describes the monitoring for the instance.
- **Enabled** (*boolean*) –
Indicates whether monitoring is enabled for the instance.
- **InstanceId** (*string*) –
The instance ID, if an instance has been launched to fulfill the Spot instance request.
- **CreateTime** (*datetime*) –
The date and time when the Spot instance request was created, in UTC format (for example, `YYYY-MM-DD T*HH* :MM :SS Z`).
- **ProductDescription** (*string*) –
The product description associated with the Spot instance.
- **Tags** (*list*) –
Any tags assigned to the resource.

- *(dict)* –
Describes a tag.
- **Key** (*string*) –
The key of the tag.
Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`
- **Value** (*string*) –
The value of the tag.
Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.
- **LaunchedAvailabilityZone** (*string*) –
The Availability Zone in which the bid is launched.

describe_spot_price_history (***kwargs*)

Describes the Spot price history. The prices returned are listed in chronological order, from the oldest to the most recent, for up to the past 90 days. For more information, see [Spot Instance Pricing History](#) in the *Amazon Elastic Compute Cloud User Guide*.

When you specify a start and end time, this operation returns the prices of the instance types within the time range that you specified and the time when the price changed. The price is valid within the time period that you specified; the response merely indicates the last time that the price changed.

Request Syntax

```
response = client.describe_spot_price_history(
    DryRun=True|False,
    StartTime=datetime(2015, 1, 1),
    EndTime=datetime(2015, 1, 1),
    InstanceTypes=[
        't1.micro'|'m1.small'|'m1.medium'|'m1.large'|'m1.xlarge'|'m3.medium'|'m3.large'|'m3'
    ],
    ProductDescriptions=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    AvailabilityZone='string',
    MaxResults=123,
    NextToken='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **StartTime** (*datetime*) – The date and time, up to the past 90 days, from which to start retrieving the price history data, in UTC format (for example, `YYYY-MM-DD T*HH* :MM :SS Z`).
- **EndTime** (*datetime*) – The date and time, up to the current date, from which to stop retrieving the price history data, in UTC format (for example, `YYYY-MM-DD T*HH* :MM :SS Z`).
- **InstanceTypes** (*list*) – Filters the results by the specified instance types.
 - (*string*) –
- **ProductDescriptions** (*list*) – Filters the results by the specified basic product descriptions.
 - (*string*) –
- **Filters** (*list*) – One or more filters.
 - `availability-zone` - The Availability Zone for which prices should be returned.
 - `instance-type` - The type of instance (for example, `m1.small`).
 - `product-description` - The product description for the Spot price (`Linux/UNIX|SUSE Linux|Windows|Linux/UNIX (Amazon VPC)|SUSE Linux (Amazon VPC)|Windows (Amazon VPC)`).
 - `spot-price` - The Spot price. The value must match exactly (or use wildcards; greater than or less than comparison is not supported).
 - `timestamp` - The timestamp of the Spot price history, in UTC format (for example, `YYYY-MM-DD T*HH* :MM :SS Z`). You can use wildcards (`*` and `?`). Greater than or less than comparison is not supported.
 - (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

 - * **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.
 - * **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.
 - (*string*) –
- **AvailabilityZone** (*string*) – Filters the results by the specified Availability Zone.
- **MaxResults** (*integer*) – The maximum number of results to return in a single call. Specify a value between 1 and 1000. The default value is 1000. To retrieve the remaining results, make another call with the returned `NextToken` value.
- **NextToken** (*string*) – The token for the next set of results.

Return type dict

Returns

Response Syntax

```
{
  'SpotPriceHistory': [
    {
      'InstanceType': 't1.micro'|'m1.small'|'m1.medium'|'m1.large'|'m1.xlarge',
      'ProductDescription': 'Linux/UNIX'|'Linux/UNIX (Amazon VPC)'|'Windows',
      'SpotPrice': 'string',
      'Timestamp': datetime(2015, 1, 1),
      'AvailabilityZone': 'string'
    },
  ],
}
```

```

    'NextToken': 'string'
}

```

Response Structure

- *(dict)* –

Contains the output of DescribeSpotPriceHistory.

- **SpotPriceHistory** (*list*) –

The historical Spot prices.

- * *(dict)* –

Describes the maximum hourly price (bid) for any Spot instance launched to fulfill the request.

- **InstanceType** (*string*) –

The instance type.

- **ProductDescription** (*string*) –

A general description of the AMI.

- **SpotPrice** (*string*) –

The maximum price (bid) that you are willing to pay for a Spot instance.

- **Timestamp** (*datetime*) –

The date and time the request was created, in UTC format (for example, *YYYY-MM-DD T*HH* :MM :SS Z*).

- **AvailabilityZone** (*string*) –

The Availability Zone.

- **NextToken** (*string*) –

The token required to retrieve the next set of results. This value is `null` when there are no more results to return.

describe_subnets (***kwargs*)

Describes one or more of your subnets.

For more information about subnets, see [Your VPC and Subnets](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```

response = client.describe_subnets(
    DryRun=True|False,
    SubnetIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SubnetIds** (*list*) – One or more subnet IDs.

Default: Describes all your subnets.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `availabilityZone` - The Availability Zone for the subnet. You can also use `availability-zone` as the filter name.
 - `available-ip-address-count` - The number of IP addresses in the subnet that are available.
 - `cidrBlock` - The CIDR block of the subnet. The CIDR block you specify must exactly match the subnet's CIDR block for information to be returned for the subnet. You can also use `cidr` or `cidr-block` as the filter names.
 - `defaultForAz` - Indicates whether this is the default subnet for the Availability Zone. You can also use `default-for-az` as the filter name.
 - `state` - The state of the subnet (`pending`|`available`).
 - `subnet-id` - The ID of the subnet.
 - `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
 - `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag's value is), and the tag value `X` (regardless of what the tag's key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
 - `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
 - `vpc-id` - The ID of the VPC for the subnet.
- (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'Subnets': [
    {
      'SubnetId': 'string',
      'State': 'pending'|'available',
      'VpcId': 'string',
```

```

        'CidrBlock': 'string',
        'AvailableIpAddressCount': 123,
        'AvailabilityZone': 'string',
        'DefaultForAz': True|False,
        'MapPublicIpOnLaunch': True|False,
        'Tags': [
            {
                'Key': 'string',
                'Value': 'string'
            },
        ]
    },
]
}

```

Response Structure

- (*dict*) –

- **Subnets** (*list*) –

Information about one or more subnets.

- * (*dict*) –

Describes a subnet.

- **SubnetId** (*string*) –

The ID of the subnet.

- **State** (*string*) –

The current state of the subnet.

- **VpcId** (*string*) –

The ID of the VPC the subnet is in.

- **CidrBlock** (*string*) –

The CIDR block assigned to the subnet.

- **AvailableIpAddressCount** (*integer*) –

The number of unused IP addresses in the subnet. Note that the IP addresses for any stopped instances are considered unavailable.

- **AvailabilityZone** (*string*) –

The Availability Zone of the subnet.

- **DefaultForAz** (*boolean*) –

Indicates whether this is the default subnet for the Availability Zone.

- **MapPublicIpOnLaunch** (*boolean*) –

Indicates whether instances launched in this subnet receive a public IP address.

- **Tags** (*list*) –

Any tags assigned to the subnet.

- (*dict*) –

Describes a tag.

- **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

· **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

describe_tags (***kwargs*)

Describes one or more of the tags for your EC2 resources.

For more information about tags, see [Tagging Your Resources](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.describe_tags(
    DryRun=True|False,
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxResults=123,
    NextToken='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Filters** (*list*) – One or more filters.

- key - The tag key.
- resource-id - The resource ID.
- resource-type - The resource type (customer-gateway | dhcp-options | image | instance | internet-gateway | network-acl | network-interface | reserved-instances | route-table | security-group | snapshot | spot-instances-request | subnet | volume | vpc | vpn-connection | vpn-gateway).
- value - The tag value.
- (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

- **MaxResults** (*integer*) – The maximum number of results to return for the request in a single page. The remaining results of the initial request can be seen by sending another request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned.
- **NextToken** (*string*) – The token to retrieve the next page of results.

Return type dict

Returns

Response Syntax

```
{
  'Tags': [
    {
      'ResourceId': 'string',
      'ResourceType': 'customer-gateway'|'dhcp-options'|'image'|'instance'|'...',
      'Key': 'string',
      'Value': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **Tags** (*list*) –

A list of tags.

 - * (*dict*) –

Describes a tag.

 - **ResourceId** (*string*) –

The ID of the resource. For example, ami-1a2b3c4d.
 - **ResourceType** (*string*) –

The resource type.
 - **Key** (*string*) –

The tag key.
 - **Value** (*string*) –

The tag value.
 - **NextToken** (*string*) –

The token to use to retrieve the next page of results. This value is `null` when there are no more results to return..

describe_volume_attribute (**kwargs)

Describes the specified attribute of the specified volume. You can specify only one attribute at a time.

For more information about EBS volumes, see [Amazon EBS Volumes](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.describe_volume_attribute(
    DryRun=True|False,
    VolumeId='string',
```

```
Attribute='autoEnableIO'|'productCodes'  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VolumeId** (*string*) – **[REQUIRED]**
The ID of the volume.
- **Attribute** (*string*) – The instance attribute.

Return type dict**Returns****Response Syntax**

```
{  
  'VolumeId': 'string',  
  'AutoEnableIO': {  
    'Value': True|False  
  },  
  'ProductCodes': [  
    {  
      'ProductCodeId': 'string',  
      'ProductCodeType': 'devpay'|'marketplace'  
    },  
  ]  
}
```

Response Structure

- (*dict*) –
 - **VolumeId** (*string*) –
The ID of the volume.
 - **AutoEnableIO** (*dict*) –
The state of `autoEnableIO` attribute.
 - * **Value** (*boolean*) –
Valid values are `true` or `false`.
 - **ProductCodes** (*list*) –
A list of product codes.
 - * (*dict*) –
Describes a product code.
 - **ProductCodeId** (*string*) –
The product code.
 - **ProductCodeType** (*string*) –
The type of product code.

describe_volume_status (***kwargs*)

Describes the status of the specified volumes. Volume status provides the result of the checks performed on your volumes to determine events that can impair the performance of your volumes. The performance of a volume can be affected if an issue occurs on the volume's underlying host. If the volume's underlying host experiences a power outage or system issue, after the system is restored, there could be data

inconsistencies on the volume. Volume events notify you if this occurs. Volume actions notify you if any action needs to be taken in response to the event.

The `DescribeVolumeStatus` operation provides the following information about the specified volumes:

Status : Reflects the current status of the volume. The possible values are `ok`, `impaired`, `warning`, or `insufficient-data`. If all checks pass, the overall status of the volume is `ok`. If the check fails, the overall status is `impaired`. If the status is `insufficient-data`, then the checks may still be taking place on your volume at the time. We recommend that you retry the request. For more information on volume status, see [Monitoring the Status of Your Volumes](#).

Events : Reflect the cause of a volume status and may require you to take action. For example, if your volume returns an `impaired` status, then the volume event might be `potential-data-inconsistency`. This means that your volume has been affected by an issue with the underlying host, has all I/O operations disabled, and may have inconsistent data.

Actions : Reflect the actions you may have to take in response to an event. For example, if the status of the volume is `impaired` and the volume event shows `potential-data-inconsistency`, then the action shows `enable-volume-io`. This means that you may want to enable the I/O operations for the volume by calling the `EnableVolumeIO` action and then check the volume for data consistency.

Note: Volume status is based on the volume status checks, and does not reflect the volume state. Therefore, volume status does not indicate volumes in the `error` state (for example, when a volume is incapable of accepting I/O.)

Request Syntax

```
response = client.describe_volume_status(
    DryRun=True|False,
    VolumeIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VolumeIds** (*list*) – One or more volume IDs.
Default: Describes all your volumes.
 - (*string*) –
- **Filters** (*list*) – One or more filters.
 - `action.code` - The action code for the event (for example, `enable-volume-io`).
 - `action.description` - A description of the action.

- `action.event-id` - The event ID associated with the action.
- `availability-zone` - The Availability Zone of the instance.
- `event.description` - A description of the event.
- `event.event-id` - The event ID.
- `event.event-type` - The event type (for `io-enabled`: `passed` | `failed`; for `io-performance`: `io-performance:degraded` | `io-performance:severely-degraded` | `io-performance:stalled`).
- `event.not-after` - The latest end time for the event.
- `event.not-before` - The earliest start time for the event.
- `volume-status.details-name` - The cause for `volume-status.status` (`io-enabled` | `io-performance`).
- `volume-status.details-status` - The status of `volume-status.details-name` (for `io-enabled`: `passed` | `failed`; for `io-performance`: `normal` | `degraded` | `severely-degraded` | `stalled`).
- `volume-status.status` - The status of the volume (`ok` | `impaired` | `warning` | `insufficient-data`).

- (*dict*) -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

- **NextToken** (*string*) - The `NextToken` value to include in a future `DescribeVolumeStatus` request. When the results of the request exceed `MaxResults`, this value can be used to retrieve the next page of results. This value is `null` when there are no more results to return.
- **MaxResults** (*integer*) - The maximum number of volume results returned by `DescribeVolumeStatus` in paginated output. When this parameter is used, the request only returns `MaxResults` results in a single page along with a `NextToken` response element. The remaining results of the initial request can be seen by sending another request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. If this parameter is not used, then `DescribeVolumeStatus` returns all results. You cannot specify this parameter and the volume IDs parameter in the same request.

Return type `dict`

Returns

Response Syntax

```
{
  'VolumeStatuses': [
    {
      'VolumeId': 'string',
      'AvailabilityZone': 'string',
      'VolumeStatus': {
        'Status': 'ok'|'impaired'|'insufficient-data',
```

```

        'Details': [
            {
                'Name': 'io-enabled'|'io-performance',
                'Status': 'string'
            },
        ]
    },
    'Events': [
        {
            'EventType': 'string',
            'Description': 'string',
            'NotBefore': datetime(2015, 1, 1),
            'NotAfter': datetime(2015, 1, 1),
            'EventId': 'string'
        },
    ],
    'Actions': [
        {
            'Code': 'string',
            'Description': 'string',
            'EventType': 'string',
            'EventId': 'string'
        },
    ],
},
'NextToken': 'string'
}

```

Response Structure

- *(dict)* –
 - **VolumeStatuses** (*list*) –

A list of volumes.

 - * *(dict)* –

Describes the volume status.

 - **VolumeId** (*string*) –

The volume ID.
 - **AvailabilityZone** (*string*) –

The Availability Zone of the volume.
 - **VolumeStatus** (*dict*) –

The volume status.
 - **Status** (*string*) –

The status of the volume.
 - **Details** (*list*) –

The details of the volume status.
 - *(dict)* –

Describes a volume status.
 - **Name** (*string*) –

The name of the volume status.
 - **Status** (*string*) –

The intended status of the volume status.

- **Events** (*list*) –
A list of events associated with the volume.
- (*dict*) –
Describes a volume status event.
- **EventType** (*string*) –
The type of this event.
- **Description** (*string*) –
A description of the event.
- **NotBefore** (*datetime*) –
The earliest start time of the event.
- **NotAfter** (*datetime*) –
The latest end time of the event.
- **EventId** (*string*) –
The ID of this event.
- **Actions** (*list*) –
The details of the operation.
- (*dict*) –
Describes a volume status operation code.
- **Code** (*string*) –
The code identifying the operation, for example, `enable-volume-io`.
- **Description** (*string*) –
A description of the operation.
- **EventType** (*string*) –
The event type associated with this operation.
- **EventId** (*string*) –
The ID of the event associated with this operation.
- **NextToken** (*string*) –
The token to use to retrieve the next page of results. This value is `null` when there are no more results to return.

describe_volumes (***kwargs*)

Describes the specified EBS volumes.

If you are describing a long list of volumes, you can paginate the output to make the list more manageable. The `MaxResults` parameter sets the maximum number of results returned in a single page. If the list of results exceeds your `MaxResults` value, then that number of results is returned along with a `NextToken` value that can be passed to a subsequent `DescribeVolumes` request to retrieve the remaining results.

For more information about EBS volumes, see [Amazon EBS Volumes](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.describe_volumes(  
    DryRun=True|False,  
    VolumeIds=[  
        'string',
```

```

],
Filters=[
    {
        'Name': 'string',
        'Values': [
            'string',
        ]
    },
],
NextToken='string',
MaxResults=123
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VolumeIds** (*list*) – One or more volume IDs.
 - (*string*) –
- **Filters** (*list*) – One or more filters.
 - `attachment.attach-time` - The time stamp when the attachment initiated.
 - `attachment.delete-on-termination` - Whether the volume is deleted on instance termination.
 - `attachment.device` - The device name that is exposed to the instance (for example, `/dev/sda1`).
 - `attachment.instance-id` - The ID of the instance the volume is attached to.
 - `attachment.status` - The attachment state (`attaching` | `attached` | `detaching` | `detached`).
 - `availability-zone` - The Availability Zone in which the volume was created.
 - `create-time` - The time stamp when the volume was created.
 - `encrypted` - The encryption status of the volume.
 - `size` - The size of the volume, in GiB.
 - `snapshot-id` - The snapshot from which the volume was created.
 - `status` - The status of the volume (`creating` | `available` | `in-use` | `deleting` | `deleted` | `error`).
 - `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
 - `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag’s value is), and the tag value `X` (regardless of what the tag’s key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
 - `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
 - `volume-id` - The volume ID.
 - `volume-type` - The Amazon EBS volume type. This can be `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, or `standard` for Magnetic volumes.
 - (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

- **NextToken** (*string*) – The NextToken value returned from a previous paginated DescribeVolumes request where MaxResults was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the NextToken value. This value is null when there are no more results to return.
- **MaxResults** (*integer*) – The maximum number of volume results returned by DescribeVolumes in paginated output. When this parameter is used, DescribeVolumes only returns MaxResults results in a single page along with a NextToken response element. The remaining results of the initial request can be seen by sending another DescribeVolumes request with the returned NextToken value. This value can be between 5 and 1000; if MaxResults is given a value larger than 1000, only 1000 results are returned. If this parameter is not used, then DescribeVolumes returns all results. You cannot specify this parameter and the volume IDs parameter in the same request.

Return type dict

Returns

Response Syntax

```
{
  'Volumes': [
    {
      'VolumeId': 'string',
      'Size': 123,
      'SnapshotId': 'string',
      'AvailabilityZone': 'string',
      'State': 'creating'|'available'|'in-use'|'deleting'|'deleted'|'error',
      'CreateTime': datetime(2015, 1, 1),
      'Attachments': [
        {
          'VolumeId': 'string',
          'InstanceId': 'string',
          'Device': 'string',
          'State': 'attaching'|'attached'|'detaching'|'detached',
          'AttachTime': datetime(2015, 1, 1),
          'DeleteOnTermination': True|False
        },
      ],
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        },
      ],
      'VolumeType': 'standard'|'io1'|'gp2',
      'Iops': 123,
      'Encrypted': True|False,
```



```

        'KmsKeyId': 'string'
    },
],
'NextToken': 'string'
}

```

Response Structure

- *(dict)* –

- **Volumes** (*list*) –

Information about the volumes.

- * *(dict)* –

Describes a volume.

- **VolumeId** (*string*) –

The ID of the volume.

- **Size** (*integer*) –

The size of the volume, in GiBs.

- **SnapshotId** (*string*) –

The snapshot from which the volume was created, if applicable.

- **AvailabilityZone** (*string*) –

The Availability Zone for the volume.

- **State** (*string*) –

The volume state.

- **CreateTime** (*datetime*) –

The time stamp when volume creation was initiated.

- **Attachments** (*list*) –

Information about the volume attachments.

- *(dict)* –

Describes volume attachment details.

- **VolumeId** (*string*) –

The ID of the volume.

- **InstanceId** (*string*) –

The ID of the instance.

- **Device** (*string*) –

The device name.

- **State** (*string*) –

The attachment state of the volume.

- **AttachTime** (*datetime*) –

The time stamp when the attachment initiated.

- **DeleteOnTermination** (*boolean*) –

Indicates whether the EBS volume is deleted on instance termination.

- **Tags** (*list*) –

Any tags assigned to the volume.

- *(dict)* –

Describes a tag.

- **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

- **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

- **VolumeType** (*string*) –

The volume type. This can be `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, or `standard` for Magnetic volumes.

- **Iops** (*integer*) –

The number of I/O operations per second (IOPS) that the volume supports. For Provisioned IOPS (SSD) volumes, this represents the number of IOPS that are provisioned for the volume. For General Purpose (SSD) volumes, this represents the baseline performance of the volume and the rate at which the volume accumulates I/O credits for bursting. For more information on General Purpose (SSD) baseline performance, I/O credits, and bursting, see [Amazon EBS Volume Types](#) in the *Amazon Elastic Compute Cloud User Guide*.

Constraint: Range is 100 to 20000 for Provisioned IOPS (SSD) volumes and 3 to 10000 for General Purpose (SSD) volumes.

Condition: This parameter is required for requests to create `io1` volumes; it is not used in requests to create `standard` or `gp2` volumes.

- **Encrypted** (*boolean*) –

Indicates whether the volume will be encrypted.

- **KmsKeyId** (*string*) –

The full ARN of the AWS Key Management Service (AWS KMS) customer master key (CMK) that was used to protect the volume encryption key for the volume.

- **NextToken** (*string*) –

The `NextToken` value to include in a future `DescribeVolumes` request. When the results of a `DescribeVolumes` request exceed `MaxResults`, this value can be used to retrieve the next page of results. This value is `null` when there are no more results to return.

describe_vpc_attribute (***kwargs*)

Describes the specified attribute of the specified VPC. You can specify only one attribute at a time.

Request Syntax

```
response = client.describe_vpc_attribute(
    DryRun=True|False,
```

```
VpcId='string',
Attribute='enableDnsSupport'|'enableDnsHostnames'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcId** (*string*) – **[REQUIRED]**
The ID of the VPC.
- **Attribute** (*string*) – The VPC attribute.

Return type dict

Returns

Response Syntax

```
{
  'VpcId': 'string',
  'EnableDnsSupport': {
    'Value': True|False
  },
  'EnableDnsHostnames': {
    'Value': True|False
  }
}
```

Response Structure

- (*dict*) –
 - **VpcId** (*string*) –
The ID of the VPC.
 - **EnableDnsSupport** (*dict*) –
Indicates whether DNS resolution is enabled for the VPC. If this attribute is `true`, the Amazon DNS server resolves DNS hostnames for your instances to their corresponding IP addresses; otherwise, it does not.
* **Value** (*boolean*) –
Valid values are `true` or `false`.
 - **EnableDnsHostnames** (*dict*) –
Indicates whether the instances launched in the VPC get DNS hostnames. If this attribute is `true`, instances in the VPC get DNS hostnames; otherwise, they do not.
* **Value** (*boolean*) –
Valid values are `true` or `false`.

describe_vpc_classic_link (***kwargs*)

Describes the ClassicLink status of one or more VPCs.

Request Syntax

```
response = client.describe_vpc_classic_link(
    DryRun=True|False,
    VpcIds=[
        'string',
```

```
],
Filters=[
    {
        'Name': 'string',
        'Values': [
            'string',
        ]
    },
]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcIds** (*list*) – One or more VPCs for which you want to describe the ClassicLink status.
 - (*string*) –
- **Filters** (*list*) – One or more filters.
 - `is-classic-link-enabled` - Whether the VPC is enabled for ClassicLink (`true` | `false`).
 - `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
 - `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag’s value is), and the tag value `X` (regardless of what the tag’s key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
 - `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
 - (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'Vpcs': [
        {
            'VpcId': 'string',
            'ClassicLinkEnabled': True|False,
            'Tags': [
                {
```

```

        'Key': 'string',
        'Value': 'string'
    },
]
}

```

Response Structure

- *(dict)* –

- **Vpcs** (*list*) –

The ClassicLink status of one or more VPCs.

- * *(dict)* –

Describes whether a VPC is enabled for ClassicLink.

- **VpcId** (*string*) –

The ID of the VPC.

- **ClassicLinkEnabled** (*boolean*) –

Indicates whether the VPC is enabled for ClassicLink.

- **Tags** (*list*) –

Any tags assigned to the VPC.

- *(dict)* –

Describes a tag.

- **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

- **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

describe_vpc_endpoint_services (***kwargs*)

Describes all supported AWS services that can be specified when creating a VPC endpoint.

Request Syntax

```

response = client.describe_vpc_endpoint_services(
    DryRun=True|False,
    MaxResults=123,
    NextToken='string'
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **MaxResults** (*integer*) – The maximum number of items to return for this request. The request returns a token that you can specify in a subsequent call to get the next set of results.

Constraint: If the value is greater than 1000, we return only 1000 items.

- **NextToken** (*string*) – The token for the next set of items to return. (You received this token from a prior call.)

Return type dict

Returns

Response Syntax

```
{
    'ServiceNames': [
        'string',
    ],
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **ServiceNames** (*list*) –
A list of supported AWS services.
 - * (*string*) –
 - **NextToken** (*string*) –
The token to use when requesting the next set of items. If there are no additional items to return, the string is empty.

describe_vpc_endpoints (***kwargs*)

Describes one or more of your VPC endpoints.

Request Syntax

```
response = client.describe_vpc_endpoints(
    DryRun=True|False,
    VpcEndpointIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxResults=123,
    NextToken='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcEndpointIds** (*list*) – One or more endpoint IDs.
 - (*string*) –
- **Filters** (*list*) – One or more filters.
 - `service-name`: The name of the AWS service.
 - `vpc-id`: The ID of the VPC in which the endpoint resides.

- `vpc-endpoint-id`: The ID of the endpoint.
- `vpc-endpoint-state`: The state of the endpoint. (`pending` | `available` | `deleting` | `deleted`)
- (*dict*) -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

- * **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

- * **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

- (*string*) -

- **MaxResults** (*integer*) - The maximum number of items to return for this request. The request returns a token that you can specify in a subsequent call to get the next set of results.

Constraint: If the value is greater than 1000, we return only 1000 items.

- **NextToken** (*string*) - The token for the next set of items to return. (You received this token from a prior call.)

Return type dict

Returns

Response Syntax

```
{
  'VpcEndpoints': [
    {
      'VpcEndpointId': 'string',
      'VpcId': 'string',
      'ServiceName': 'string',
      'State': 'Pending'|'Available'|'Deleting'|'Deleted',
      'PolicyDocument': 'string',
      'RouteTableIds': [
        'string',
      ],
      'CreationTimestamp': datetime(2015, 1, 1)
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) -
- **VpcEndpoints** (*list*) -

Information about the endpoints.

- * (*dict*) -

Describes a VPC endpoint.

- **VpcEndpointId** (*string*) -

The ID of the VPC endpoint.

- **VpcId** (*string*) -

The ID of the VPC to which the endpoint is associated.

- **ServiceName** (*string*) –
The name of the AWS service to which the endpoint is associated.
- **State** (*string*) –
The state of the VPC endpoint.
- **PolicyDocument** (*string*) –
The policy document associated with the endpoint.
- **RouteTableIds** (*list*) –
One or more route tables associated with the endpoint.
- (*string*) –
- **CreationTimestamp** (*datetime*) –
The date and time the VPC endpoint was created.
- **NextToken** (*string*) –
The token to use when requesting the next set of items. If there are no additional items to return, the string is empty.

describe_vpc_peering_connections (**kwargs)
Describes one or more of your VPC peering connections.

Request Syntax

```
response = client.describe_vpc_peering_connections(
    DryRun=True|False,
    VpcPeeringConnectionIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcPeeringConnectionIds** (*list*) – One or more VPC peering connection IDs.
Default: Describes all your VPC peering connections.
 - (*string*) –
- **Filters** (*list*) – One or more filters.
 - `accepter-vpc-info.cidr-block` - The CIDR block of the peer VPC.
 - `accepter-vpc-info.owner-id` - The AWS account ID of the owner of the peer VPC.
 - `accepter-vpc-info.vpc-id` - The ID of the peer VPC.
 - `expiration-time` - The expiration date and time for the VPC peering connection.

- requester-vpc-info.cidr-block - The CIDR block of the requester's VPC.
- requester-vpc-info.owner-id - The AWS account ID of the owner of the requester VPC.
- requester-vpc-info.vpc-id - The ID of the requester VPC.
- status-code - The status of the VPC peering connection (pending-acceptance | failed | expired | provisioning | active | deleted | rejected).
- status-message - A message that provides more information about the status of the VPC peering connection, if applicable.
- tag :key =*value* - The key/value combination of a tag assigned to the resource.
- tag-key - The key of a tag assigned to the resource. This filter is independent of the tag-value filter. For example, if you use both the filter "tag-key=Purpose" and the filter "tag-value=X", you get any resources assigned both the tag key Purpose (regardless of what the tag's value is), and the tag value X (regardless of what the tag's key is). If you want to list only resources where Purpose is X, see the tag :key =*value* filter.
- tag-value - The value of a tag assigned to the resource. This filter is independent of the tag-key filter.
- vpc-peering-connection-id - The ID of the VPC peering connection.
- (dict) -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

Return type dict

Returns

Response Syntax

```
{
  'VpcPeeringConnections': [
    {
      'AcceptorVpcInfo': {
        'CidrBlock': 'string',
        'OwnerId': 'string',
        'VpcId': 'string'
      },
      'ExpirationTime': datetime(2015, 1, 1),
      'RequesterVpcInfo': {
        'CidrBlock': 'string',
        'OwnerId': 'string',
        'VpcId': 'string'
      },
      'Status': {
        'Code': 'initiating-request'|'pending-acceptance'|'active'|'delete',
        'Message': 'string'
      },
    },
  ],
}
```

```
        'Tags': [
            {
                'Key': 'string',
                'Value': 'string'
            },
        ],
        'VpcPeeringConnectionId': 'string'
    },
]
```

Response Structure

- (*dict*) –

- **VpcPeeringConnections** (*list*) –

Information about the VPC peering connections.

- * (*dict*) –

Describes a VPC peering connection.

- **AccepterVpcInfo** (*dict*) –

- The information of the peer VPC.

- **CidrBlock** (*string*) –

- The CIDR block for the VPC.

- **OwnerId** (*string*) –

- The AWS account ID of the VPC owner.

- **VpcId** (*string*) –

- The ID of the VPC.

- **ExpirationTime** (*datetime*) –

- The time that an unaccepted VPC peering connection will expire.

- **RequesterVpcInfo** (*dict*) –

- The information of the requester VPC.

- **CidrBlock** (*string*) –

- The CIDR block for the VPC.

- **OwnerId** (*string*) –

- The AWS account ID of the VPC owner.

- **VpcId** (*string*) –

- The ID of the VPC.

- **Status** (*dict*) –

- The status of the VPC peering connection.

- **Code** (*string*) –

- The status of the VPC peering connection.

- **Message** (*string*) –

- A message that provides more information about the status, if applicable.

- **Tags** (*list*) –

- Any tags assigned to the resource.

- (*dict*) –

Describes a tag.

- **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

- **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

- **VpcPeeringConnectionId** (*string*) –

The ID of the VPC peering connection.

describe_vpcs (***kwargs*)

Describes one or more of your VPCs.

Request Syntax

```
response = client.describe_vpcs(
    DryRun=True|False,
    VpcIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcIds** (*list*) – One or more VPC IDs.

Default: Describes all your VPCs.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `cidr` - The CIDR block of the VPC. The CIDR block you specify must exactly match the VPC's CIDR block for information to be returned for the VPC. Must contain the slash followed by one or two digits (for example, `/28`).
 - `dhcp-options-id` - The ID of a set of DHCP options.
 - `isDefault` - Indicates whether the VPC is the default VPC.
 - `state` - The state of the VPC (`pending`|`available`).
 - `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
 - `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter

“tag-key=Purpose” and the filter “tag-value=X”, you get any resources assigned both the tag key Purpose (regardless of what the tag’s value is), and the tag value X (regardless of what the tag’s key is). If you want to list only resources where Purpose is X, see the `tag:key=*value*` filter.

- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `vpc-id` - The ID of the VPC.
- *(dict)* –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

- * **Name** (*string*) –

- The name of the filter. Filter names are case-sensitive.

- * **Values** (*list*) –

- One or more filter values. Filter values are case-sensitive.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'Vpcs': [
    {
      'VpcId': 'string',
      'State': 'pending'|'available',
      'CidrBlock': 'string',
      'DhcpOptionsId': 'string',
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        },
      ],
      'InstanceTenancy': 'default'|'dedicated',
      'IsDefault': True|False
    },
  ],
}
```

Response Structure

- (*dict*) –
 - **Vpcs** (*list*) –

Information about one or more VPCs.

- * (*dict*) –

- Describes a VPC.

- **VpcId** (*string*) –

- The ID of the VPC.

- **State** (*string*) –

- The current state of the VPC.

- **CidrBlock** (*string*) –

The CIDR block for the VPC.

- **DhcpOptionsId** (*string*) –

The ID of the set of DHCP options you’ve associated with the VPC (or `default` if the default options are associated with the VPC).

- **Tags** (*list*) –

Any tags assigned to the VPC.

- (*dict*) –

Describes a tag.

- **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

- **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

- **InstanceTenancy** (*string*) –

The allowed tenancy of instances launched into the VPC.

- **IsDefault** (*boolean*) –

Indicates whether the VPC is the default VPC.

describe_vpn_connections (***kwargs*)

Describes one or more of your VPN connections.

For more information about VPN connections, see [Adding a Hardware Virtual Private Gateway to Your VPC](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.describe_vpn_connections(
    DryRun=True|False,
    VpnConnectionIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpnConnectionIds** (*list*) – One or more VPN connection IDs.

Default: Describes your VPN connections.

- (*string*) -
- **Filters** (*list*) - One or more filters.
 - `customer-gateway-configuration` - The configuration information for the customer gateway.
 - `customer-gateway-id` - The ID of a customer gateway associated with the VPN connection.
 - `state` - The state of the VPN connection (`pending` | `available` | `deleting` | `deleted`).
 - `option.static-routes-only` - Indicates whether the connection has static routes only. Used for devices that do not support Border Gateway Protocol (BGP).
 - `route.destination-cidr-block` - The destination CIDR block. This corresponds to the subnet used in a customer data center.
 - `bgp-asn` - The BGP Autonomous System Number (ASN) associated with a BGP device.
 - `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
 - `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag’s value is), and the tag value `X` (regardless of what the tag’s key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
 - `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
 - `type` - The type of VPN connection. Currently the only supported type is `ipsec.1`.
 - `vpn-connection-id` - The ID of the VPN connection.
 - `vpn-gateway-id` - The ID of a virtual private gateway associated with the VPN connection.
- (*dict*) -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

Return type `dict`

Returns

Response Syntax

```
{
  'VpnConnections': [
    {
      'VpnConnectionId': 'string',
      'State': 'pending'|'available'|'deleting'|'deleted',
      'CustomerGatewayConfiguration': 'string',
      'Type': 'ipsec.1',
      'CustomerGatewayId': 'string',
      'VpnGatewayId': 'string',
```

```

    'Tags': [
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
    'VgwTelemetry': [
        {
            'OutsideIpAddress': 'string',
            'Status': 'UP'|'DOWN',
            'LastStatusChange': datetime(2015, 1, 1),
            'StatusMessage': 'string',
            'AcceptedRouteCount': 123
        },
    ],
    'Options': {
        'StaticRoutesOnly': True|False
    },
    'Routes': [
        {
            'DestinationCidrBlock': 'string',
            'Source': 'Static',
            'State': 'pending'|'available'|'deleting'|'deleted'
        },
    ],
  ],
}

```

Response Structure

- *(dict)* –
- **VpnConnections** (*list*) –

Information about one or more VPN connections.

* *(dict)* –

Describes a VPN connection.

- **VpnConnectionId** (*string*) –
The ID of the VPN connection.
- **State** (*string*) –
The current state of the VPN connection.
- **CustomerGatewayConfiguration** (*string*) –
The configuration information for the VPN connection's customer gateway (in the native XML format). This element is always present in the CreateVpnConnection response; however, it's present in the DescribeVpnConnections response only if the VPN connection is in the pending or available state.
- **Type** (*string*) –
The type of VPN connection.
- **CustomerGatewayId** (*string*) –
The ID of the customer gateway at your end of the VPN connection.
- **VpnGatewayId** (*string*) –

The ID of the virtual private gateway at the AWS side of the VPN connection.

- **Tags** (*list*) –

Any tags assigned to the VPN connection.

- (*dict*) –

Describes a tag.

- **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

- **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

- **VgwTelemetry** (*list*) –

Information about the VPN tunnel.

- (*dict*) –

Describes telemetry for a VPN tunnel.

- **OutsideIpAddress** (*string*) –

The Internet-routable IP address of the virtual private gateway's outside interface.

- **Status** (*string*) –

The status of the VPN tunnel.

- **LastStatusChange** (*datetime*) –

The date and time of the last change in status.

- **StatusMessage** (*string*) –

If an error occurs, a description of the error.

- **AcceptedRouteCount** (*integer*) –

The number of accepted routes.

- **Options** (*dict*) –

The VPN connection options.

- **StaticRoutesOnly** (*boolean*) –

Indicates whether the VPN connection uses static routes only. Static routes must be used for devices that don't support BGP.

- **Routes** (*list*) –

The static routes associated with the VPN connection.

- (*dict*) –

Describes a static route for a VPN connection.

- **DestinationCidrBlock** (*string*) –

The CIDR block associated with the local subnet of the customer data center.

- **Source** (*string*) –

Indicates how the routes were provided.

· **State** (*string*) –

The current state of the static route.

describe_vpn_gateways (***kwargs*)

Describes one or more of your virtual private gateways.

For more information about virtual private gateways, see [Adding an IPsec Hardware VPN to Your VPC](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.describe_vpn_gateways(
    DryRun=True|False,
    VpnGatewayIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpnGatewayIds** (*list*) – One or more virtual private gateway IDs.

Default: Describes all your virtual private gateways.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `attachment.state` - The current state of the attachment between the gateway and the VPC (`attaching` | `attached` | `detaching` | `detached`).
 - `attachment.vpc-id` - The ID of an attached VPC.
 - `availability-zone` - The Availability Zone for the virtual private gateway.
 - `state` - The state of the virtual private gateway (`pending` | `available` | `deleting` | `deleted`).
 - `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
 - `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag’s value is), and the tag value `X` (regardless of what the tag’s key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
 - `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
 - `type` - The type of virtual private gateway. Currently the only supported type is `ipsec.1`.
 - `vpn-gateway-id` - The ID of the virtual private gateway.

– (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'VpnGateways': [
    {
      'VpnGatewayId': 'string',
      'State': 'pending'|'available'|'deleting'|'deleted',
      'Type': 'ipsec.1',
      'AvailabilityZone': 'string',
      'VpcAttachments': [
        {
          'VpcId': 'string',
          'State': 'attaching'|'attached'|'detaching'|'detached'
        },
      ],
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        },
      ],
    },
  ]
}
```

Response Structure

• (*dict*) –

– **VpnGateways** (*list*) –

Information about one or more virtual private gateways.

* (*dict*) –

Describes a virtual private gateway.

· **VpnGatewayId** (*string*) –

The ID of the virtual private gateway.

· **State** (*string*) –

The current state of the virtual private gateway.

· **Type** (*string*) –

The type of VPN connection the virtual private gateway supports.

· **AvailabilityZone** (*string*) –

The Availability Zone where the virtual private gateway was created.

- **VpcAttachments** (*list*) –

Any VPCs attached to the virtual private gateway.

- (*dict*) –

Describes an attachment between a virtual private gateway and a VPC.

- **VpcId** (*string*) –

The ID of the VPC.

- **State** (*string*) –

The current state of the attachment.

- **Tags** (*list*) –

Any tags assigned to the virtual private gateway.

- (*dict*) –

Describes a tag.

- **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

- **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

detach_classic_link_vpc (***kwargs*)

Unlinks (detaches) a linked EC2-Classic instance from a VPC. After the instance has been unlinked, the VPC security groups are no longer associated with it. An instance is automatically unlinked from a VPC when it's stopped.

Request Syntax

```
response = client.detach_classic_link_vpc(
    DryRun=True|False,
    InstanceId='string',
    VpcId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceId** (*string*) – **[REQUIRED]**
The ID of the instance to unlink from the VPC.
- **VpcId** (*string*) – **[REQUIRED]**
The ID of the VPC to which the instance is linked.

Return type dict

Returns

Response Syntax

```
{
    'Return': True|False
}
```

Response Structure

- (*dict*) –
 - **Return** (*boolean*) –

Returns `true` if the request succeeds; otherwise, it returns an error.

`detach_internet_gateway` (**kwargs)

Detaches an Internet gateway from a VPC, disabling connectivity between the Internet and the VPC. The VPC must not contain any running instances with Elastic IP addresses.

Request Syntax

```
response = client.detach_internet_gateway(
    DryRun=True|False,
    InternetGatewayId='string',
    VpcId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InternetGatewayId** (*string*) – [REQUIRED]
The ID of the Internet gateway.
- **VpcId** (*string*) – [REQUIRED]
The ID of the VPC.

Returns None

`detach_network_interface` (**kwargs)

Detaches a network interface from an instance.

Request Syntax

```
response = client.detach_network_interface(
    DryRun=True|False,
    AttachmentId='string',
    Force=True|False
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **AttachmentId** (*string*) – [REQUIRED]
The ID of the attachment.
- **Force** (*boolean*) – Specifies whether to force a detachment.

Returns None

`detach_volume` (**kwargs)

Detaches an EBS volume from an instance. Make sure to unmount any file systems on the device within

your operating system before detaching the volume. Failure to do so results in the volume being stuck in a busy state while detaching.

If an Amazon EBS volume is the root device of an instance, it can't be detached while the instance is running. To detach the root volume, stop the instance first.

When a volume with an AWS Marketplace product code is detached from an instance, the product code is no longer associated with the instance.

For more information, see [Detaching an Amazon EBS Volume](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.detach_volume(
    DryRun=True|False,
    VolumeId='string',
    InstanceId='string',
    Device='string',
    Force=True|False
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VolumeId** (*string*) – **[REQUIRED]**
The ID of the volume.
- **InstanceId** (*string*) – The ID of the instance.
- **Device** (*string*) – The device name.
- **Force** (*boolean*) – Forces detachment if the previous detachment attempt did not occur cleanly (for example, logging into an instance, unmounting the volume, and detaching normally). This option can lead to data loss or a corrupted file system. Use this option only as a last resort to detach a volume from a failed instance. The instance won't have an opportunity to flush file system caches or file system metadata. If you use this option, you must perform file system check and repair procedures.

Return type dict

Returns

Response Syntax

```
{
    'VolumeId': 'string',
    'InstanceId': 'string',
    'Device': 'string',
    'State': 'attaching'|'attached'|'detaching'|'detached',
    'AttachTime': datetime(2015, 1, 1),
    'DeleteOnTermination': True|False
}
```

Response Structure

- (*dict*) –
Information about the volume attachment.

- **VolumeId** (*string*) –
The ID of the volume.
- **InstanceId** (*string*) –
The ID of the instance.
- **Device** (*string*) –
The device name.
- **State** (*string*) –
The attachment state of the volume.
- **AttachTime** (*datetime*) –
The time stamp when the attachment initiated.
- **DeleteOnTermination** (*boolean*) –
Indicates whether the EBS volume is deleted on instance termination.

detach_vpn_gateway (***kwargs*)

Detaches a virtual private gateway from a VPC. You do this if you're planning to turn off the VPC and not use it anymore. You can confirm a virtual private gateway has been completely detached from a VPC by describing the virtual private gateway (any attachments to the virtual private gateway are also described).

You must wait for the attachment's state to switch to `detached` before you can delete the VPC or attach a different VPC to the virtual private gateway.

Request Syntax

```
response = client.detach_vpn_gateway(  
    DryRun=True|False,  
    VpnGatewayId='string',  
    VpcId='string'  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpnGatewayId** (*string*) – **[REQUIRED]**
The ID of the virtual private gateway.
- **VpcId** (*string*) – **[REQUIRED]**
The ID of the VPC.

Returns None

disable_vgw_route_propagation (***kwargs*)

Disables a virtual private gateway (VGW) from propagating routes to a specified route table of a VPC.

Request Syntax

```
response = client.disable_vgw_route_propagation(  
    RouteTableId='string',  
    GatewayId='string'  
)
```

Parameters

- **RouteTableId** (*string*) – **[REQUIRED]**
The ID of the route table.

- **GatewayId** (*string*) – [REQUIRED]

The ID of the virtual private gateway.

Returns None

disable_vpc_classic_link (***kwargs*)

Disables ClassicLink for a VPC. You cannot disable ClassicLink for a VPC that has EC2-Classic instances linked to it.

Request Syntax

```
response = client.disable_vpc_classic_link(
    DryRun=True|False,
    VpcId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcId** (*string*) – [REQUIRED]

The ID of the VPC.

Return type dict

Returns

Response Syntax

```
{
    'Return': True|False
}
```

Response Structure

- (*dict*) –
 - **Return** (*boolean*) –

Returns `true` if the request succeeds; otherwise, it returns an error.

disassociate_address (***kwargs*)

Disassociates an Elastic IP address from the instance or network interface it's associated with.

An Elastic IP address is for use in either the EC2-Classic platform or in a VPC. For more information, see [Elastic IP Addresses](#) in the *Amazon Elastic Compute Cloud User Guide*.

This is an idempotent operation. If you perform the operation more than once, Amazon EC2 doesn't return an error.

Request Syntax

```
response = client.disassociate_address(
    DryRun=True|False,
    PublicIp='string',
    AssociationId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If

you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **PublicIp** (*string*) – [EC2-Classic] The Elastic IP address. Required for EC2-Classic.
- **AssociationId** (*string*) – [EC2-VPC] The association ID. Required for EC2-VPC.

Returns None

disassociate_route_table (***kwargs*)

Disassociates a subnet from a route table.

After you perform this action, the subnet no longer uses the routes in the route table. Instead, it uses the routes in the VPC's main route table. For more information about route tables, see [Route Tables](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.disassociate_route_table(
    DryRun=True|False,
    AssociationId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **AssociationId** (*string*) – **[REQUIRED]**

The association ID representing the current association between the route table and subnet.

Returns None

enable_vgw_route_propagation (***kwargs*)

Enables a virtual private gateway (VGW) to propagate routes to the specified route table of a VPC.

Request Syntax

```
response = client.enable_vgw_route_propagation(
    RouteTableId='string',
    GatewayId='string'
)
```

Parameters

- **RouteTableId** (*string*) – **[REQUIRED]**

The ID of the route table.

- **GatewayId** (*string*) – **[REQUIRED]**

The ID of the virtual private gateway.

Returns None

enable_volume_io (***kwargs*)

Enables I/O operations for a volume that had I/O operations disabled because the data on the volume was potentially inconsistent.

Request Syntax


```
response = client.enable_volume_io(
    DryRun=True|False,
    VolumeId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VolumeId** (*string*) – **[REQUIRED]**

The ID of the volume.

Returns None

enable_vpc_classic_link (**kwargs)

Enables a VPC for ClassicLink. You can then link EC2-Classic instances to your ClassicLink-enabled VPC to allow communication over private IP addresses. You cannot enable your VPC for ClassicLink if any of your VPC's route tables have existing routes for address ranges within the 10.0.0.0/8 IP address range, excluding local routes for VPCs in the 10.0.0.0/16 and 10.1.0.0/16 IP address ranges. For more information, see [ClassicLink](#) in the Amazon Elastic Compute Cloud User Guide.

Request Syntax

```
response = client.enable_vpc_classic_link(
    DryRun=True|False,
    VpcId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcId** (*string*) – **[REQUIRED]**

The ID of the VPC.

Return type dict

Returns

Response Syntax

```
{
    'Return': True|False
}
```

Response Structure

- (*dict*) –
 - **Return** (*boolean*) –

Returns `true` if the request succeeds; otherwise, it returns an error.

generate_presigned_url (ClientMethod, Params=None, ExpiresIn=3600, HttpMethod=None)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.

- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_console_output (***kwargs*)

Gets the console output for the specified instance.

Instances do not have a physical monitor through which you can view their console output. They also lack physical controls that allow you to power up, reboot, or shut them down. To allow these actions, we provide them through the Amazon EC2 API and command line interface.

Instance console output is buffered and posted shortly after instance boot, reboot, and termination. Amazon EC2 preserves the most recent 64 KB output which is available for at least one hour after the most recent post.

For Linux instances, the instance console output displays the exact console output that would normally be displayed on a physical monitor attached to a computer. This output is buffered because the instance produces it and then posts it to a store where the instance's owner can retrieve it.

For Windows instances, the instance console output includes output from the EC2Config service.

Request Syntax

```
response = client.get_console_output(  
    DryRun=True|False,  
    InstanceId='string'  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceId** (*string*) – **[REQUIRED]**

The ID of the instance.

Return type dict

Returns

Response Syntax

```
{  
    'InstanceId': 'string',  
    'Timestamp': datetime(2015, 1, 1),  
    'Output': 'string'  
}
```

Response Structure

- (*dict*) –
 - **InstanceId** (*string*) –
The ID of the instance.
 - **Timestamp** (*datetime*) –
The time the output was last updated.
 - **Output** (*string*) –
The console output, Base64 encoded.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_password_data (***kwargs*)

Retrieves the encrypted administrator password for an instance running Windows.

The Windows password is generated at boot if the EC2Config service plugin, `Ec2SetPassword`, is enabled. This usually only happens the first time an AMI is launched, and then `Ec2SetPassword` is automatically disabled. The password is not generated for rebundled AMIs unless `Ec2SetPassword` is enabled before bundling.

The password is encrypted using the key pair that you specified when you launched the instance. You must provide the corresponding key pair file.

Password generation and encryption takes a few moments. We recommend that you wait up to 15 minutes after launching an instance before trying to retrieve the generated password.

Request Syntax

```
response = client.get_password_data(
    DryRun=True|False,
    InstanceId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceId** (*string*) – **[REQUIRED]**

The ID of the Windows instance.

Return type `dict`

Returns**Response Syntax**

```
{
    'InstanceId': 'string',
    'Timestamp': datetime(2015, 1, 1),
    'PasswordData': 'string'
}
```

Response Structure

- (*dict*) –
 - **InstanceId** (*string*) –

The ID of the Windows instance.

- **Timestamp** (*datetime*) –
The time the data was last updated.
- **PasswordData** (*string*) –
The password of the instance.

get_waiter (*waiter_name*)

import_image (***kwargs*)

Import single or multi-volume disk images or EBS snapshots into an Amazon Machine Image (AMI).

Request Syntax

```
response = client.import_image(
    DryRun=True|False,
    Description='string',
    DiskContainers=[
        {
            'Description': 'string',
            'Format': 'string',
            'Url': 'string',
            'UserBucket': {
                'S3Bucket': 'string',
                'S3Key': 'string'
            },
            'DeviceName': 'string',
            'SnapshotId': 'string'
        },
    ],
    LicenseType='string',
    Hypervisor='string',
    Architecture='string',
    Platform='string',
    ClientData={
        'UploadStart': datetime(2015, 1, 1),
        'UploadEnd': datetime(2015, 1, 1),
        'UploadSize': 123.0,
        'Comment': 'string'
    },
    ClientToken='string',
    RoleName='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Description** (*string*) – A description string for the import image task.
- **DiskContainers** (*list*) – Information about the disk containers.
 - (*dict*) –
Describes the disk container object for an import image task.
 - * **Description** (*string*) –
The description of the disk image.
 - * **Format** (*string*) –
The format of the disk image being imported.

Valid values: RAW | VHD | VMDK | OVA

* **Url** (*string*) –

The URL to the Amazon S3-based disk image being imported. The URL can either be a https URL (<https://..>) or an Amazon S3 URL (<s3://..>)

* **UserBucket** (*dict*) –

The S3 bucket for the disk image.

· **S3Bucket** (*string*) –

The name of the S3 bucket where the disk image is located.

· **S3Key** (*string*) –

The key for the disk image.

* **DeviceName** (*string*) –

The block device mapping for the disk.

* **SnapshotId** (*string*) –

The ID of the EBS snapshot to be used for importing the snapshot.

- **LicenseType** (*string*) – The license type to be used for the Amazon Machine Image (AMI) after importing.

Note: You may only use BYOL if you have existing licenses with rights to use these licenses in a third party cloud like AWS. For more information, see [VM Import/Export Prerequisites](#) in the *Amazon Elastic Compute Cloud User Guide*.

Valid values: AWS | BYOL

- **Hypervisor** (*string*) – The target hypervisor platform.

Valid values: xen

- **Architecture** (*string*) – The architecture of the virtual machine.

Valid values: i386 | x86_64

- **Platform** (*string*) – The operating system of the virtual machine.

Valid values: Windows | Linux

- **ClientData** (*dict*) – The client-specific data.

– **UploadStart** (*datetime*) –

The time that the disk upload starts.

– **UploadEnd** (*datetime*) –

The time that the disk upload ends.

– **UploadSize** (*float*) –

The size of the uploaded disk image, in GiB.

– **Comment** (*string*) –

A user-defined comment about the disk upload.

- **ClientToken** (*string*) – The token to enable idempotency for VM import requests.
- **RoleName** (*string*) – The name of the role to use when not using the default role, 'vmimport'.

Return type dict

Returns

Response Syntax

```
{
    'ImportTaskId': 'string',
    'Architecture': 'string',
    'LicenseType': 'string',
    'Platform': 'string',
    'Hypervisor': 'string',
    'Description': 'string',
    'SnapshotDetails': [
        {
            'DiskImageSize': 123.0,
            'Description': 'string',
            'Format': 'string',
            'Url': 'string',
            'UserBucket': {
                'S3Bucket': 'string',
                'S3Key': 'string'
            },
            'DeviceName': 'string',
            'SnapshotId': 'string',
            'Progress': 'string',
            'StatusMessage': 'string',
            'Status': 'string'
        },
    ],
    'ImageId': 'string',
    'Progress': 'string',
    'StatusMessage': 'string',
    'Status': 'string'
}
```

Response Structure

- *(dict)* –
 - **ImportTaskId** (*string*) –
The task ID of the import image task.
 - **Architecture** (*string*) –
The architecture of the virtual machine.
 - **LicenseType** (*string*) –
The license type of the virtual machine.
 - **Platform** (*string*) –
The operating system of the virtual machine.
 - **Hypervisor** (*string*) –
The target hypervisor of the import task.
 - **Description** (*string*) –
A description of the import task.
 - **SnapshotDetails** (*list*) –
Information about the snapshots.
 - * *(dict)* –
Describes the snapshot created from the imported disk.
 - **DiskImageSize** (*float*) –
The size of the disk in the snapshot, in GiB.
 - **Description** (*string*) –

- A description for the snapshot.
 - **Format** (*string*) –
 - The format of the disk image from which the snapshot is created.
 - **Url** (*string*) –
 - The URL used to access the disk image.
 - **UserBucket** (*dict*) –
 - Describes the S3 bucket for the disk image.
 - **S3Bucket** (*string*) –
 - The S3 bucket from which the disk image was created.
 - **S3Key** (*string*) –
 - The key from which the disk image was created.
 - **DeviceName** (*string*) –
 - The block device mapping for the snapshot.
 - **SnapshotId** (*string*) –
 - The snapshot ID of the disk being imported.
 - **Progress** (*string*) –
 - The percentage of progress for the task.
 - **StatusMessage** (*string*) –
 - A detailed status message for the snapshot creation.
 - **Status** (*string*) –
 - A brief status of the snapshot creation.
- **ImageId** (*string*) –
 - The ID of the Amazon Machine Image (AMI) created by the import task.
- **Progress** (*string*) –
 - The progress of the task.
- **StatusMessage** (*string*) –
 - A detailed status message of the import task.
- **Status** (*string*) –
 - A brief status of the task.

import_instance (**kwargs)

Creates an import instance task using metadata from the specified disk image. `ImportInstance` only supports single-volume VMs. To import multi-volume VMs, use `ImportImage`. After importing the image, you then upload it using the `ec2-import-volume` command in the EC2 command line tools. For more information, see [Using the Command Line Tools to Import Your Virtual Machine to Amazon EC2](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.import_instance(
    DryRun=True|False,
    Description='string',
    LaunchSpecification={
        'Architecture': 'i386'|'x86_64',
        'GroupNames': [
            'string',
        ],
    },
)
```

```
        'GroupIds': [
            'string',
        ],
        'AdditionalInfo': 'string',
        'UserData': {
            'Data': 'string'
        },
        'InstanceType': 't1.micro'|'m1.small'|'m1.medium'|'m1.large'|'m1.xlarge'|'m3.medium',
        'Placement': {
            'AvailabilityZone': 'string',
            'GroupName': 'string',
            'Tenancy': 'default'|'dedicated'
        },
        'Monitoring': True|False,
        'SubnetId': 'string',
        'InstanceInitiatedShutdownBehavior': 'stop'|'terminate',
        'PrivateIpAddress': 'string'
    },
    DiskImages=[
        {
            'Image': {
                'Format': 'VMDK'|'RAW'|'VHD',
                'Bytes': 123,
                'ImportManifestUrl': 'string'
            },
            'Description': 'string',
            'Volume': {
                'Size': 123
            }
        },
    ],
    Platform='Windows'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Description** (*string*) – A description for the instance being imported.
- **LaunchSpecification** (*dict*) – The launch specification.
 - **Architecture** (*string*) –
The architecture of the instance.
 - **GroupNames** (*list*) –
One or more security group names.
 - * (*string*) –
 - **GroupIds** (*list*) –
One or more security group IDs.
 - * (*string*) –
 - **AdditionalInfo** (*string*) –
Reserved.
 - **UserData** (*dict*) –
The Base64-encoded MIME user data to be made available to the instance.
 - * **Data** (*string*) –

The Base64-encoded MIME user data for the instance.

– **InstanceType** (*string*) –

The instance type. For more information about the instance types that you can import, see [Before You Get Started](#) in the Amazon Elastic Compute Cloud User Guide.

– **Placement** (*dict*) –

The placement information for the instance.

* **AvailabilityZone** (*string*) –

The Availability Zone of the instance.

* **GroupName** (*string*) –

The name of the placement group the instance is in (for cluster compute instances).

* **Tenancy** (*string*) –

The tenancy of the instance (if the instance is running in a VPC). An instance with a tenancy of `dedicated` runs on single-tenant hardware.

– **Monitoring** (*boolean*) –

Indicates whether monitoring is enabled.

– **SubnetId** (*string*) –

[EC2-VPC] The ID of the subnet in which to launch the instance.

– **InstanceInitiatedShutdownBehavior** (*string*) –

Indicates whether an instance stops or terminates when you initiate shutdown from the instance (using the operating system command for system shutdown).

– **PrivateIpAddress** (*string*) –

[EC2-VPC] An available IP address from the IP address range of the subnet.

• **DiskImages** (*list*) – The disk image.

– (*dict*) –

Describes a disk image.

* **Image** (*dict*) –

Information about the disk image.

· **Format** (*string*) – [REQUIRED]

The disk image format.

· **Bytes** (*integer*) – [REQUIRED]

The size of the disk image, in GiB.

· **ImportManifestUrl** (*string*) – [REQUIRED]

A presigned URL for the import manifest stored in Amazon S3 and presented here as an Amazon S3 presigned URL. For information about creating a presigned URL for an Amazon S3 object, read the “Query String Request Authentication Alternative” section of the [Authenticating REST Requests](#) topic in the *Amazon Simple Storage Service Developer Guide*.

* **Description** (*string*) –

A description of the disk image.

* **Volume** (*dict*) –

Information about the volume.

· **Size** (*integer*) – **[REQUIRED]**

The size of the volume, in GiB.

• **Platform** (*string*) – **[REQUIRED]**

The instance operating system.

Return type dict

Returns

Response Syntax

```
{
  'ConversionTask': {
    'ConversionTaskId': 'string',
    'ExpirationTime': 'string',
    'ImportInstance': {
      'Volumes': [
        {
          'BytesConverted': 123,
          'AvailabilityZone': 'string',
          'Image': {
            'Format': 'VMDK'|'RAW'|'VHD',
            'Size': 123,
            'ImportManifestUrl': 'string',
            'Checksum': 'string'
          },
          'Volume': {
            'Size': 123,
            'Id': 'string'
          },
          'Status': 'string',
          'StatusMessage': 'string',
          'Description': 'string'
        },
      ],
      'InstanceId': 'string',
      'Platform': 'Windows',
      'Description': 'string'
    },
    'ImportVolume': {
      'BytesConverted': 123,
      'AvailabilityZone': 'string',
      'Description': 'string',
      'Image': {
        'Format': 'VMDK'|'RAW'|'VHD',
        'Size': 123,
        'ImportManifestUrl': 'string',
        'Checksum': 'string'
      },
      'Volume': {
        'Size': 123,
        'Id': 'string'
      }
    },
    'State': 'active'|'cancelling'|'cancelled'|'completed',
    'StatusMessage': 'string',
```

```

        'Tags': [
            {
                'Key': 'string',
                'Value': 'string'
            },
        ]
    }
}

```

Response Structure

- (*dict*) –

- **ConversionTask** (*dict*) –

Information about the conversion task.

- * **ConversionTaskId** (*string*) –

The ID of the conversion task.

- * **ExpirationTime** (*string*) –

The time when the task expires. If the upload isn't complete before the expiration time, we automatically cancel the task.

- * **ImportInstance** (*dict*) –

If the task is for importing an instance, this contains information about the import instance task.

- **Volumes** (*list*) –

One or more volumes.

- (*dict*) –

Describes an import volume task.

- **BytesConverted** (*integer*) –

The number of bytes converted so far.

- **AvailabilityZone** (*string*) –

The Availability Zone where the resulting instance will reside.

- **Image** (*dict*) –

The image.

- **Format** (*string*) –

The disk image format.

- **Size** (*integer*) –

The size of the disk image, in GiB.

- **ImportManifestUrl** (*string*) –

A presigned URL for the import manifest stored in Amazon S3. For information about creating a presigned URL for an Amazon S3 object, read the “Query String Request Authentication Alternative” section of the [Authenticating REST Requests](#) topic in the *Amazon Simple Storage Service Developer Guide*.

- **Checksum** (*string*) –

The checksum computed for the disk image.

- **Volume** (*dict*) –

The volume.

- **Size** (*integer*) –
The size of the volume, in GiB.
- **Id** (*string*) –
The volume identifier.
- **Status** (*string*) –
The status of the import of this particular disk image.
- **StatusMessage** (*string*) –
The status information or errors related to the disk image.
- **Description** (*string*) –
A description of the task.
- **InstanceId** (*string*) –
The ID of the instance.
- **Platform** (*string*) –
The instance operating system.
- **Description** (*string*) –
A description of the task.

* **ImportVolume** (*dict*) –

If the task is for importing a volume, this contains information about the import volume task.

- **BytesConverted** (*integer*) –
The number of bytes converted so far.
- **AvailabilityZone** (*string*) –
The Availability Zone where the resulting volume will reside.
- **Description** (*string*) –
The description you provided when starting the import volume task.
- **Image** (*dict*) –
The image.
- **Format** (*string*) –
The disk image format.
- **Size** (*integer*) –
The size of the disk image, in GiB.
- **ImportManifestUrl** (*string*) –
A presigned URL for the import manifest stored in Amazon S3. For information about creating a presigned URL for an Amazon S3 object, read the “Query String Request Authentication Alternative” section of the [Authenticating REST Requests](#) topic in the *Amazon Simple Storage Service Developer Guide*.
- **Checksum** (*string*) –
The checksum computed for the disk image.
- **Volume** (*dict*) –
The volume.

- **Size** (*integer*) –
The size of the volume, in GiB.
- **Id** (*string*) –
The volume identifier.
- * **State** (*string*) –
The state of the conversion task.
- * **StatusMessage** (*string*) –
The status message related to the conversion task.
- * **Tags** (*list*) –
Any tags assigned to the task.
 - (*dict*) –
Describes a tag.
 - **Key** (*string*) –
The key of the tag.
Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`
 - **Value** (*string*) –
The value of the tag.
Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

import_key_pair (***kwargs*)

Imports the public key from an RSA key pair that you created with a third-party tool. Compare this with `CreateKeyPair`, in which AWS creates the key pair and gives the keys to you (AWS keeps a copy of the public key). With `ImportKeyPair`, you create the key pair and give AWS just the public key. The private key is never transferred between you and AWS.

For more information about key pairs, see [Key Pairs](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.import_key_pair(
    DryRun=True|False,
    KeyName='string',
    PublicKeyMaterial=b'bytes'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **KeyName** (*string*) – **[REQUIRED]**
A unique name for the key pair.
- **PublicKeyMaterial** (*bytes*) – **[REQUIRED]**
The public key. You must base64 encode the public key material before sending it to AWS.

Return type dict

Returns

Response Syntax

```
{
    'KeyName': 'string',
    'KeyFingerprint': 'string'
}
```

Response Structure

- *(dict)* –
 - **KeyName** (*string*) –
The key pair name you provided.
 - **KeyFingerprint** (*string*) –
The MD5 public key fingerprint as specified in section 4 of RFC 4716.

```
import_snapshot (**kwargs)
```

Imports a disk into an EBS snapshot.

Request Syntax

```
response = client.import_snapshot(
    DryRun=True|False,
    Description='string',
    DiskContainer={
        'Description': 'string',
        'Format': 'string',
        'Url': 'string',
        'UserBucket': {
            'S3Bucket': 'string',
            'S3Key': 'string'
        }
    },
    ClientData={
        'UploadStart': datetime(2015, 1, 1),
        'UploadEnd': datetime(2015, 1, 1),
        'UploadSize': 123.0,
        'Comment': 'string'
    },
    ClientToken='string',
    RoleName='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Description** (*string*) – The description string for the import snapshot task.
- **DiskContainer** (*dict*) – Information about the disk container.
 - **Description** (*string*) –
The description of the disk image being imported.
 - **Format** (*string*) –
The format of the disk image being imported.
Valid values: RAW | VHD | VMDK | OVA

- **Url** (*string*) –
The URL to the Amazon S3-based disk image being imported. It can either be a https URL (<https://..>) or an Amazon S3 URL (<s3://..>).
- **UserBucket** (*dict*) –
Describes the S3 bucket for the disk image.
 - * **S3Bucket** (*string*) –
The name of the S3 bucket where the disk image is located.
 - * **S3Key** (*string*) –
The key for the disk image.
- **ClientData** (*dict*) – The client-specific data.
 - **UploadStart** (*datetime*) –
The time that the disk upload starts.
 - **UploadEnd** (*datetime*) –
The time that the disk upload ends.
 - **UploadSize** (*float*) –
The size of the uploaded disk image, in GiB.
 - **Comment** (*string*) –
A user-defined comment about the disk upload.
- **ClientToken** (*string*) – Token to enable idempotency for VM import requests.
- **RoleName** (*string*) – The name of the role to use when not using the default role, 'vmimport'.

Return type dict

Returns

Response Syntax

```
{
  'ImportTaskId': 'string',
  'SnapshotTaskDetail': {
    'DiskImageSize': 123.0,
    'Description': 'string',
    'Format': 'string',
    'Url': 'string',
    'UserBucket': {
      'S3Bucket': 'string',
      'S3Key': 'string'
    },
    'SnapshotId': 'string',
    'Progress': 'string',
    'StatusMessage': 'string',
    'Status': 'string'
  },
  'Description': 'string'
}
```

Response Structure

- (*dict*) –
 - **ImportTaskId** (*string*) –
The ID of the import snapshot task.
 - **SnapshotTaskDetail** (*dict*) –
Information about the import snapshot task.

- * **DiskImageSize** (*float*) –
The size of the disk in the snapshot, in GiB.
- * **Description** (*string*) –
The description of the snapshot.
- * **Format** (*string*) –
The format of the disk image from which the snapshot is created.
- * **Url** (*string*) –
The URL of the disk image from which the snapshot is created.
- * **UserBucket** (*dict*) –
The S3 bucket for the disk image.
 - **S3Bucket** (*string*) –
The S3 bucket from which the disk image was created.
 - **S3Key** (*string*) –
The key from which the disk image was created.
- * **SnapshotId** (*string*) –
The snapshot ID of the disk being imported.
- * **Progress** (*string*) –
The percentage of completion for the import snapshot task.
- * **StatusMessage** (*string*) –
A detailed status message for the import snapshot task.
- * **Status** (*string*) –
A brief status for the import snapshot task.
- **Description** (*string*) –
A description of the import snapshot task.

import_volume (***kwargs*)

Creates an import volume task using metadata from the specified disk image. After importing the image, you then upload it using the `ec2-import-volume` command in the Amazon EC2 command-line interface (CLI) tools. For more information, see [Using the Command Line Tools to Import Your Virtual Machine to Amazon EC2](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.import_volume(  
    DryRun=True|False,  
    AvailabilityZone='string',  
    Image={  
        'Format': 'VMDK'|'RAW'|'VHD',  
        'Bytes': 123,  
        'ImportManifestUrl': 'string'  
    },  
    Description='string',  
    Volume={  
        'Size': 123  
    }  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If

you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **AvailabilityZone** (*string*) – [REQUIRED]

The Availability Zone for the resulting EBS volume.

- **Image** (*dict*) – [REQUIRED]

The disk image.

- **Format** (*string*) – [REQUIRED]

The disk image format.

- **Bytes** (*integer*) – [REQUIRED]

The size of the disk image, in GiB.

- **ImportManifestUrl** (*string*) – [REQUIRED]

A presigned URL for the import manifest stored in Amazon S3 and presented here as an Amazon S3 presigned URL. For information about creating a presigned URL for an Amazon S3 object, read the “Query String Request Authentication Alternative” section of the [Authenticating REST Requests](#) topic in the *Amazon Simple Storage Service Developer Guide*.

- **Description** (*string*) – A description of the volume.
- **Volume** (*dict*) – [REQUIRED]

The volume size.

- **Size** (*integer*) – [REQUIRED]

The size of the volume, in GiB.

Return type dict

Returns

Response Syntax

```
{
  'ConversionTask': {
    'ConversionTaskId': 'string',
    'ExpirationTime': 'string',
    'ImportInstance': {
      'Volumes': [
        {
          'BytesConverted': 123,
          'AvailabilityZone': 'string',
          'Image': {
            'Format': 'VMDK'|'RAW'|'VHD',
            'Size': 123,
            'ImportManifestUrl': 'string',
            'Checksum': 'string'
          },
          'Volume': {
            'Size': 123,
            'Id': 'string'
          },
          'Status': 'string',
          'StatusMessage': 'string',
          'Description': 'string'
        },
        ...
      ],
      'InstanceId': 'string',
      'Platform': 'Windows',
      'Description': 'string'
    }
  }
}
```

```
    },
    'ImportVolume': {
        'BytesConverted': 123,
        'AvailabilityZone': 'string',
        'Description': 'string',
        'Image': {
            'Format': 'VMDK'|'RAW'|'VHD',
            'Size': 123,
            'ImportManifestUrl': 'string',
            'Checksum': 'string'
        },
        'Volume': {
            'Size': 123,
            'Id': 'string'
        }
    },
    'State': 'active'|'cancelling'|'cancelled'|'completed',
    'StatusMessage': 'string',
    'Tags': [
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
]
}
```

Response Structure

- (*dict*) –

- **ConversionTask** (*dict*) –

Information about the conversion task.

- * **ConversionTaskId** (*string*) –

The ID of the conversion task.

- * **ExpirationTime** (*string*) –

The time when the task expires. If the upload isn't complete before the expiration time, we automatically cancel the task.

- * **ImportInstance** (*dict*) –

If the task is for importing an instance, this contains information about the import instance task.

- **Volumes** (*list*) –

One or more volumes.

- (*dict*) –

Describes an import volume task.

- **BytesConverted** (*integer*) –

The number of bytes converted so far.

- **AvailabilityZone** (*string*) –

The Availability Zone where the resulting instance will reside.

- **Image** (*dict*) –

The image.

- **Format** (*string*) –

The disk image format.

- **Size** (*integer*) –

The size of the disk image, in GiB.

- **ImportManifestUrl** (*string*) –

A presigned URL for the import manifest stored in Amazon S3. For information about creating a presigned URL for an Amazon S3 object, read the “Query String Request Authentication Alternative” section of the [Authenticating REST Requests](#) topic in the *Amazon Simple Storage Service Developer Guide*.

- **Checksum** (*string*) –

The checksum computed for the disk image.

- **Volume** (*dict*) –

The volume.

- **Size** (*integer*) –

The size of the volume, in GiB.

- **Id** (*string*) –

The volume identifier.

- **Status** (*string*) –

The status of the import of this particular disk image.

- **StatusMessage** (*string*) –

The status information or errors related to the disk image.

- **Description** (*string*) –

A description of the task.

- **InstanceId** (*string*) –

The ID of the instance.

- **Platform** (*string*) –

The instance operating system.

- **Description** (*string*) –

A description of the task.

- * **ImportVolume** (*dict*) –

If the task is for importing a volume, this contains information about the import volume task.

- **BytesConverted** (*integer*) –

The number of bytes converted so far.

- **AvailabilityZone** (*string*) –

The Availability Zone where the resulting volume will reside.

- **Description** (*string*) –

The description you provided when starting the import volume task.

- **Image** (*dict*) –

The image.

- **Format** (*string*) –

The disk image format.

- **Size** (*integer*) –
The size of the disk image, in GiB.
- **ImportManifestUrl** (*string*) –
A presigned URL for the import manifest stored in Amazon S3. For information about creating a presigned URL for an Amazon S3 object, read the “Query String Request Authentication Alternative” section of the [Authenticating REST Requests](#) topic in the *Amazon Simple Storage Service Developer Guide*.
- **Checksum** (*string*) –
The checksum computed for the disk image.
- **Volume** (*dict*) –
The volume.
- **Size** (*integer*) –
The size of the volume, in GiB.
- **Id** (*string*) –
The volume identifier.
- * **State** (*string*) –
The state of the conversion task.
- * **StatusMessage** (*string*) –
The status message related to the conversion task.
- * **Tags** (*list*) –
Any tags assigned to the task.
 - (*dict*) –
Describes a tag.
 - **Key** (*string*) –
The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`
 - **Value** (*string*) –
The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

modify_image_attribute (***kwargs*)

Modifies the specified attribute of the specified AMI. You can specify only one attribute at a time.

Note: AWS Marketplace product codes cannot be modified. Images with an AWS Marketplace product code cannot be made public.

Request Syntax

```
response = client.modify_image_attribute(  
    DryRun=True|False,  
    ImageId='string',  
    Attribute='string',  
    OperationType='add'|'remove',
```

```

    UserIds=[
        'string',
    ],
    UserGroups=[
        'string',
    ],
    ProductCodes=[
        'string',
    ],
    Value='string',
    LaunchPermission={
        'Add': [
            {
                'UserId': 'string',
                'Group': 'all'
            },
        ],
        'Remove': [
            {
                'UserId': 'string',
                'Group': 'all'
            },
        ]
    },
    Description={
        'Value': 'string'
    }
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **ImageId** (*string*) – **[REQUIRED]**
The ID of the AMI.
- **Attribute** (*string*) – The name of the attribute to modify.
- **OperationType** (*string*) – The operation type.
- **UserIds** (*list*) – One or more AWS account IDs. This is only valid when modifying the `launchPermission` attribute.
– (*string*) –
- **UserGroups** (*list*) – One or more user groups. This is only valid when modifying the `launchPermission` attribute.
– (*string*) –
- **ProductCodes** (*list*) – One or more product codes. After you add a product code to an AMI, it can't be removed. This is only valid when modifying the `productCodes` attribute.
– (*string*) –
- **Value** (*string*) – The value of the attribute being modified. This is only valid when modifying the `description` attribute.
- **LaunchPermission** (*dict*) – A launch permission modification.
– **Add** (*list*) –
The AWS account ID to add to the list of launch permissions for the AMI.
* (*dict*) –

- Describes a launch permission.
 - **UserId** (*string*) –
The AWS account ID.
 - **Group** (*string*) –
The name of the group.
- **Remove** (*list*) –
The AWS account ID to remove from the list of launch permissions for the AMI.
 - * (*dict*) –
Describes a launch permission.
 - **UserId** (*string*) –
The AWS account ID.
 - **Group** (*string*) –
The name of the group.
- **Description** (*dict*) – A description for the AMI.
 - **Value** (*string*) –
Valid values are case-sensitive and vary by action.

Returns None

modify_instance_attribute (***kwargs*)

Modifies the specified attribute of the specified instance. You can specify only one attribute at a time.

To modify some attributes, the instance must be stopped. For more information, see [Modifying Attributes of a Stopped Instance](#) in the *Amazon Elastic Compute Cloud User Guide* .

Request Syntax

```
response = client.modify_instance_attribute(
    DryRun=True|False,
    InstanceId='string',
    Attribute='instanceType'|'kernel'|'ramdisk'|'userData'|'disableApiTermination'|'instanceProfileArn'|'sourceDestCheck'|'blockDeviceMappings',
    Value='string',
    BlockDeviceMappings=[
        {
            'DeviceName': 'string',
            'Ebs': {
                'VolumeId': 'string',
                'DeleteOnTermination': True|False
            },
            'VirtualName': 'string',
            'NoDevice': 'string'
        },
    ],
    SourceDestCheck={
        'Value': True|False
    },
    DisableApiTermination={
        'Value': True|False
    },
    InstanceType={
        'Value': 'string'
    },
    Kernel={
        'Value': 'string'
```

```

    },
    Ramdisk={
        'Value': 'string'
    },
    UserData={
        'Value': b'bytes'
    },
    InstanceInitiatedShutdownBehavior={
        'Value': 'string'
    },
    Groups=[
        'string',
    ],
    EbsOptimized={
        'Value': True|False
    },
    SrioNetSupport={
        'Value': 'string'
    }
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceId** (*string*) – **[REQUIRED]**

The ID of the instance.

- **Attribute** (*string*) – The name of the attribute.
- **Value** (*string*) – A new value for the attribute. Use only with the `kernel`, `ramdisk`, `userData`, `disableApiTermination`, or `instanceInitiatedShutdownBehavior` attribute.
- **BlockDeviceMappings** (*list*) – Modifies the `DeleteOnTermination` attribute for volumes that are currently attached. The volume must be owned by the caller. If no value is specified for `DeleteOnTermination`, the default is `true` and the volume is deleted when the instance is terminated.

To add instance store volumes to an Amazon EBS-backed instance, you must add them when you launch the instance. For more information, see [Updating the Block Device Mapping when Launching an Instance](#) in the *Amazon Elastic Compute Cloud User Guide*.

– (*dict*) –

Describes a block device mapping entry.

- * **DeviceName** (*string*) –

The device name exposed to the instance (for example, `/dev/sdh` or `xvdh`).

- * **Ebs** (*dict*) –

Parameters used to automatically set up EBS volumes when the instance is launched.

- **VolumeId** (*string*) –

The ID of the EBS volume.

- **DeleteOnTermination** (*boolean*) –

Indicates whether the volume is deleted on instance termination.

* **VirtualName** (*string*) –

The virtual device name.

* **NoDevice** (*string*) –

suppress the specified device included in the block device mapping.

- **SourceDestCheck** (*dict*) – Specifies whether source/destination checking is enabled. A value of `true` means that checking is enabled, and `false` means checking is disabled. This value must be `false` for a NAT instance to perform NAT.

– **Value** (*boolean*) –

Valid values are `true` or `false`.

- **DisableApiTermination** (*dict*) – If the value is `true`, you can't terminate the instance using the Amazon EC2 console, CLI, or API; otherwise, you can. You cannot use this parameter for Spot Instances.

– **Value** (*boolean*) –

Valid values are `true` or `false`.

- **InstanceType** (*dict*) – Changes the instance type to the specified value. For more information, see [Instance Types](#). If the instance type is not valid, the error returned is `InvalidInstanceAttributeValue`.

– **Value** (*string*) –

Valid values are case-sensitive and vary by action.

- **Kernel** (*dict*) – Changes the instance's kernel to the specified value. We recommend that you use PV-GRUB instead of kernels and RAM disks. For more information, see [PV-GRUB](#).

– **Value** (*string*) –

Valid values are case-sensitive and vary by action.

- **Ramdisk** (*dict*) – Changes the instance's RAM disk to the specified value. We recommend that you use PV-GRUB instead of kernels and RAM disks. For more information, see [PV-GRUB](#).

– **Value** (*string*) –

Valid values are case-sensitive and vary by action.

- **UserData** (*dict*) – Changes the instance's user data to the specified value.

– **Value** (*bytes*) –

- **InstanceInitiatedShutdownBehavior** (*dict*) – Specifies whether an instance stops or terminates when you initiate shutdown from the instance (using the operating system command for system shutdown).

– **Value** (*string*) –

Valid values are case-sensitive and vary by action.

- **Groups** (*list*) – [EC2-VPC] Changes the security groups of the instance. You must specify at least one security group, even if it's just the default security group for the VPC. You must specify the security group ID, not the security group name.

– (*string*) –

- **EbsOptimized** (*dict*) – Specifies whether the instance is optimized for EBS I/O. This optimization provides dedicated throughput to Amazon EBS and an optimized configuration stack to provide optimal EBS I/O performance. This optimization isn't available with all instance types. Additional usage charges apply when using an EBS Optimized instance.

– **Value** (*boolean*) –

Valid values are `true` or `false`.

- **SriovNetSupport** (*dict*) – Set to `simple` to enable enhanced networking for the instance.

There is no way to disable enhanced networking at this time.

This option is supported only for HVM instances. Specifying this option with a PV instance can make it unreachable.

- **Value** (*string*) –

Valid values are case-sensitive and vary by action.

Returns None

modify_network_interface_attribute (***kwargs*)

Modifies the specified network interface attribute. You can specify only one attribute at a time.

Request Syntax

```
response = client.modify_network_interface_attribute(
    DryRun=True|False,
    NetworkInterfaceId='string',
    Description={
        'Value': 'string'
    },
    SourceDestCheck={
        'Value': True|False
    },
    Groups=[
        'string',
    ],
    Attachment={
        'AttachmentId': 'string',
        'DeleteOnTermination': True|False
    }
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **NetworkInterfaceId** (*string*) – **[REQUIRED]**

The ID of the network interface.

- **Description** (*dict*) – A description for the network interface.
 - **Value** (*string*) –

Valid values are case-sensitive and vary by action.

- **SourceDestCheck** (*dict*) – Indicates whether source/destination checking is enabled. A value of `true` means checking is enabled, and `false` means checking is disabled. This value must be `false` for a NAT instance to perform NAT. For more information, see [NAT Instances](#) in the *Amazon Virtual Private Cloud User Guide*.
 - **Value** (*boolean*) –

Valid values are `true` or `false`.

- **Groups** (*list*) – Changes the security groups for the network interface. The new set of groups you specify replaces the current set. You must specify at least one

group, even if it's just the default security group in the VPC. You must specify the ID of the security group, not the name.

- (*string*) –
- **Attachment** (*dict*) – Information about the interface attachment. If modifying the 'delete on termination' attribute, you must specify the ID of the interface attachment.
 - **AttachmentId** (*string*) –
The ID of the network interface attachment.
 - **DeleteOnTermination** (*boolean*) –
Indicates whether the network interface is deleted when the instance is terminated.

Returns None

modify_reserved_instances (***kwargs*)

Modifies the Availability Zone, instance count, instance type, or network platform (EC2-Classic or EC2-VPC) of your Reserved Instances. The Reserved Instances to be modified must be identical, except for Availability Zone, network platform, and instance type.

For more information, see [Modifying Reserved Instances](#) in the Amazon Elastic Compute Cloud User Guide.

Request Syntax

```
response = client.modify_reserved_instances(
    ClientToken='string',
    ReservedInstancesIds=[
        'string',
    ],
    TargetConfigurations=[
        {
            'AvailabilityZone': 'string',
            'Platform': 'string',
            'InstanceCount': 123,
            'InstanceType': 't1.micro'|'m1.small'|'m1.medium'|'m1.large'|'m1.xlarge'|'m3.me
        },
    ]
)
```

Parameters

- **ClientToken** (*string*) – A unique, case-sensitive token you provide to ensure idempotency of your modification request. For more information, see [Ensuring Idempotency](#).
- **ReservedInstancesIds** (*list*) – **[REQUIRED]**
The IDs of the Reserved Instances to modify.
 - (*string*) –
- **TargetConfigurations** (*list*) – **[REQUIRED]**
The configuration settings for the Reserved Instances to modify.
 - (*dict*) –
Describes the configuration settings for the modified Reserved Instances.
 - * **AvailabilityZone** (*string*) –
The Availability Zone for the modified Reserved Instances.
 - * **Platform** (*string*) –

The network platform of the modified Reserved Instances, which is either EC2-Classic or EC2-VPC.

* **InstanceCount** (*integer*) –

The number of modified Reserved Instances.

* **InstanceType** (*string*) –

The instance type for the modified Reserved Instances.

Return type dict

Returns

Response Syntax

```
{
    'ReservedInstancesModificationId': 'string'
}
```

Response Structure

- (*dict*) –
 - **ReservedInstancesModificationId** (*string*) –
The ID for the modification.

modify_snapshot_attribute (***kwargs*)

Adds or removes permission settings for the specified snapshot. You may add or remove specified AWS account IDs from a snapshot's list of create volume permissions, but you cannot do both in a single API call. If you need to both add and remove account IDs for a snapshot, you must use multiple API calls.

For more information on modifying snapshot permissions, see [Sharing Snapshots](#) in the *Amazon Elastic Compute Cloud User Guide*.

Note: Snapshots with AWS Marketplace product codes cannot be made public.

Request Syntax

```
response = client.modify_snapshot_attribute(
    DryRun=True|False,
    SnapshotId='string',
    Attribute='productCodes'|'createVolumePermission',
    OperationType='add'|'remove',
    UserIds=[
        'string',
    ],
    GroupNames=[
        'string',
    ],
    CreateVolumePermission={
        'Add': [
            {
                'UserId': 'string',
                'Group': 'all'
            },
        ],
        'Remove': [
            {
                'UserId': 'string',
                'Group': 'all'
            },
        ],
    }
)
```

```
}  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SnapshotId** (*string*) – **[REQUIRED]**
The ID of the snapshot.
- **Attribute** (*string*) – The snapshot attribute to modify.

Note: Only volume creation permissions may be modified at the customer level.

- **OperationType** (*string*) – The type of operation to perform to the attribute.
- **UserIds** (*list*) – The account ID to modify for the snapshot.
 - (*string*) –
- **GroupNames** (*list*) – The group to modify for the snapshot.
 - (*string*) –
- **CreateVolumePermission** (*dict*) – A JSON representation of the snapshot attribute modification.
 - **Add** (*list*) –

Adds a specific AWS account ID or group to a volume’s list of create volume permissions.

* (*dict*) –

Describes the user or group to be added or removed from the permissions for a volume.

· **UserId** (*string*) –

The specific AWS account ID that is to be added or removed from a volume’s list of create volume permissions.

· **Group** (*string*) –

The specific group that is to be added or removed from a volume’s list of create volume permissions.

– **Remove** (*list*) –

Removes a specific AWS account ID or group from a volume’s list of create volume permissions.

* (*dict*) –

Describes the user or group to be added or removed from the permissions for a volume.

· **UserId** (*string*) –

The specific AWS account ID that is to be added or removed from a volume’s list of create volume permissions.

· **Group** (*string*) –

The specific group that is to be added or removed from a volume’s list of create volume permissions.

Returns None

modify_subnet_attribute (**kwargs)

Modifies a subnet attribute.

Request Syntax

```
response = client.modify_subnet_attribute(
    SubnetId='string',
    MapPublicIpOnLaunch={
        'Value': True|False
    }
)
```

Parameters

- **SubnetId** (*string*) – [REQUIRED]
The ID of the subnet.
- **MapPublicIpOnLaunch** (*dict*) – Specify `true` to indicate that instances launched into the specified subnet should be assigned public IP address.
 - **Value** (*boolean*) –
Valid values are `true` or `false`.

Returns None

modify_volume_attribute (**kwargs)

Modifies a volume attribute.

By default, all I/O operations for the volume are suspended when the data on the volume is determined to be potentially inconsistent, to prevent undetectable, latent data corruption. The I/O access to the volume can be resumed by first enabling I/O access and then checking the data consistency on your volume.

You can change the default behavior to resume I/O operations. We recommend that you change this only for boot volumes or for volumes that are stateless or disposable.

Request Syntax

```
response = client.modify_volume_attribute(
    DryRun=True|False,
    VolumeId='string',
    AutoEnableIO={
        'Value': True|False
    }
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VolumeId** (*string*) – [REQUIRED]
The ID of the volume.
- **AutoEnableIO** (*dict*) – Indicates whether the volume should be auto-enabled for I/O operations.
 - **Value** (*boolean*) –
Valid values are `true` or `false`.

Returns None

modify_vpc_attribute (**kwargs)

Modifies the specified attribute of the specified VPC.

Request Syntax

```
response = client.modify_vpc_attribute(
    VpcId='string',
    EnableDnsSupport={
        'Value': True|False
    },
    EnableDnsHostnames={
        'Value': True|False
    }
)
```

Parameters

- **VpcId** (*string*) – [REQUIRED]

The ID of the VPC.

- **EnableDnsSupport** (*dict*) – Indicates whether the DNS resolution is supported for the VPC. If enabled, queries to the Amazon provided DNS server at the 169.254.169.253 IP address, or the reserved IP address at the base of the VPC network range “plus two” will succeed. If disabled, the Amazon provided DNS service in the VPC that resolves public DNS hostnames to IP addresses is not enabled.

– **Value** (*boolean*) –

Valid values are `true` or `false`.

- **EnableDnsHostnames** (*dict*) – Indicates whether the instances launched in the VPC get DNS hostnames. If enabled, instances in the VPC get DNS hostnames; otherwise, they do not.

You can only enable DNS hostnames if you also enable DNS support.

– **Value** (*boolean*) –

Valid values are `true` or `false`.

Returns None

modify_vpc_endpoint (***kwargs*)

Modifies attributes of a specified VPC endpoint. You can modify the policy associated with the endpoint, and you can add and remove route tables associated with the endpoint.

Request Syntax

```
response = client.modify_vpc_endpoint(
    DryRun=True|False,
    VpcEndpointId='string',
    ResetPolicy=True|False,
    PolicyDocument='string',
    AddRouteTableIds=[
        'string',
    ],
    RemoveRouteTableIds=[
        'string',
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **VpcEndpointId** (*string*) – [REQUIRED]
The ID of the endpoint.
- **ResetPolicy** (*boolean*) – Specify `true` to reset the policy document to the default policy. The default policy allows access to the service.
- **PolicyDocument** (*string*) – A policy document to attach to the endpoint. The policy must be in valid JSON format.
- **AddRouteTableIds** (*list*) – One or more route tables IDs to associate with the endpoint.
– (*string*) –
- **RemoveRouteTableIds** (*list*) – One or more route table IDs to disassociate from the endpoint.
– (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'Return': True|False
}
```

Response Structure

- (*dict*) –
– **Return** (*boolean*) –

Returns `true` if the request succeeds; otherwise, it returns an error.

monitor_instances (***kwargs*)

Enables monitoring for a running instance. For more information about monitoring instances, see [Monitoring Your Instances and Volumes in the Amazon Elastic Compute Cloud User Guide](#).

Request Syntax

```
response = client.monitor_instances(
    DryRun=True|False,
    InstanceIds=[
        'string',
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceIds** (*list*) – [REQUIRED]
One or more instance IDs.
– (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'InstanceMonitorings': [
        {
```

```
        'InstanceId': 'string',
        'Monitoring': {
            'State': 'disabled'|'disabling'|'enabled'|'pending'
        }
    },
]
}
```

Response Structure

- (*dict*) –
 - **InstanceMonitorings** (*list*) –
Monitoring information for one or more instances.
 - * (*dict*) –
Describes the monitoring information of the instance.
 - **InstanceId** (*string*) –
The ID of the instance.
 - **Monitoring** (*dict*) –
The monitoring information.
 - **State** (*string*) –
Indicates whether monitoring is enabled for the instance.

`move_address_to_vpc` (**kwargs)

Moves an Elastic IP address from the EC2-Classic platform to the EC2-VPC platform. The Elastic IP address must be allocated to your account, and it must not be associated with an instance. After the Elastic IP address is moved, it is no longer available for use in the EC2-Classic platform, unless you move it back using the `RestoreAddressToClassic` request. You cannot move an Elastic IP address that's allocated for use in the EC2-VPC platform to the EC2-Classic platform.

Request Syntax

```
response = client.move_address_to_vpc(
    DryRun=True|False,
    PublicIp='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **PublicIp** (*string*) – **[REQUIRED]**
The Elastic IP address.

Return type dict

Returns

Response Syntax

```
{
    'AllocationId': 'string',
    'Status': 'MoveInProgress'|'InVpc'|'InClassic'
}
```

Response Structure

- (*dict*) –

- **AllocationId** (*string*) –
The allocation ID for the Elastic IP address.
- **Status** (*string*) –
The status of the move of the IP address.

purchase_reserved_instances_offering (**kwargs)

Purchases a Reserved Instance for use with your account. With Amazon EC2 Reserved Instances, you obtain a capacity reservation for a certain instance configuration over a specified period of time and pay a lower hourly rate compared to on-Demand Instance pricing.

Use `DescribeReservedInstancesOfferings` to get a list of Reserved Instance offerings that match your specifications. After you've purchased a Reserved Instance, you can check for your new Reserved Instance with `DescribeReservedInstances`.

For more information, see [Reserved Instances](#) and [Reserved Instance Marketplace](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.purchase_reserved_instances_offering(
    DryRun=True|False,
    ReservedInstancesOfferingId='string',
    InstanceCount=123,
    LimitPrice={
        'Amount': 123.0,
        'CurrencyCode': 'USD'
    }
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **ReservedInstancesOfferingId** (*string*) – **[REQUIRED]**
The ID of the Reserved Instance offering to purchase.
- **InstanceCount** (*integer*) – **[REQUIRED]**
The number of Reserved Instances to purchase.
- **LimitPrice** (*dict*) – Specified for Reserved Instance Marketplace offerings to limit the total order and ensure that the Reserved Instances are not purchased at unexpected prices.
 - **Amount** (*float*) –
Used for Reserved Instance Marketplace offerings. Specifies the limit price on the total order (instanceCount * price).
 - **CurrencyCode** (*string*) –
The currency in which the `limitPrice` amount is specified. At this time, the only supported currency is USD.

Return type dict

Returns

Response Syntax

```
{
    'ReservedInstancesId': 'string'
}
```

Response Structure

- (*dict*) –
 - **ReservedInstancesId** (*string*) –

The IDs of the purchased Reserved Instances.

reboot_instances (***kwargs*)

Requests a reboot of one or more instances. This operation is asynchronous; it only queues a request to reboot the specified instances. The operation succeeds if the instances are valid and belong to you. Requests to reboot terminated instances are ignored.

If a Linux/Unix instance does not cleanly shut down within four minutes, Amazon EC2 performs a hard reboot.

For more information about troubleshooting, see [Getting Console Output and Rebooting Instances](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.reboot_instances(
    DryRun=True|False,
    InstanceIds=[
        'string',
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceIds** (*list*) – **[REQUIRED]**

One or more instance IDs.

– (*string*) –

Returns None

register_image (***kwargs*)

Registers an AMI. When you're creating an AMI, this is the final step you must complete before you can launch an instance from the AMI. For more information about creating AMIs, see [Creating Your Own AMIs](#) in the *Amazon Elastic Compute Cloud User Guide*.

Note: For Amazon EBS-backed instances, `CreateImage` creates and registers the AMI in a single request, so you don't have to register the AMI yourself.

You can also use `RegisterImage` to create an Amazon EBS-backed Linux AMI from a snapshot of a root device volume. For more information, see [Launching an Instance from a Snapshot](#) in the *Amazon Elastic Compute Cloud User Guide*.

Warning: Some Linux distributions, such as Red Hat Enterprise Linux (RHEL) and SUSE Linux Enterprise Server (SLES), use the EC2 `billingProduct` code associated with an AMI to verify subscription status for package updates. Creating an AMI from an EBS snapshot does not maintain this billing code, and subsequent instances launched from such an AMI will not be able to connect to package update infrastructure.

Similarly, although you can create a Windows AMI from a snapshot, you can't successfully launch an instance from the AMI.

To create Windows AMIs or to create AMIs for Linux operating systems that must retain AMI billing codes to work properly, see `CreateImage`.

If needed, you can deregister an AMI at any time. Any modifications you make to an AMI backed by an instance store volume invalidates its registration. If you make changes to an image, deregister the previous image and register the new image.

Note: You can't register an image where a secondary (non-root) snapshot has AWS Marketplace product codes.

Request Syntax

```
response = client.register_image(
    DryRun=True|False,
    ImageLocation='string',
    Name='string',
    Description='string',
    Architecture='i386'|'x86_64',
    KernelId='string',
    RamdiskId='string',
    RootDeviceName='string',
    BlockDeviceMappings=[
        {
            'VirtualName': 'string',
            'DeviceName': 'string',
            'Ebs': {
                'SnapshotId': 'string',
                'VolumeSize': 123,
                'DeleteOnTermination': True|False,
                'VolumeType': 'standard'|"io1"|"gp2",
                'Iops': 123,
                'Encrypted': True|False
            },
            'NoDevice': 'string'
        },
    ],
    VirtualizationType='string',
    SriovNetSupport='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **ImageLocation** (*string*) – The full path to your AMI manifest in Amazon S3 storage.
- **Name** (*string*) – **[REQUIRED]**

A name for your AMI.

Constraints: 3-128 alphanumeric characters, parentheses (`()`), square brackets (`[]`), spaces (), periods (`.`), slashes (`/`), dashes (`-`), single quotes (`'`), at-signs (`@`), or underscores (`_`)

- **Description** (*string*) – A description for your AMI.
- **Architecture** (*string*) – The architecture of the AMI.

Default: For Amazon EBS-backed AMIs, `i386`. For instance store-backed AMIs, the architecture specified in the manifest file.

- **KernelId** (*string*) – The ID of the kernel.
- **RamdiskId** (*string*) – The ID of the RAM disk.
- **RootDeviceName** (*string*) – The name of the root device (for example, `/dev/sda1`, or `/dev/xvda`).
- **BlockDeviceMappings** (*list*) – One or more block device mapping entries.
 - (*dict*) –

Describes a block device mapping.

- * **VirtualName** (*string*) –

The virtual device name (`ephemeralN`). Instance store volumes are numbered starting from 0. An instance type with 2 available instance store volumes can specify mappings for `ephemeral0` and `ephemeral1`. The number of available instance store volumes depends on the instance type. After you connect to the instance, you must mount the volume.

Constraints: For M3 instances, you must specify instance store volumes in the block device mapping for the instance. When you launch an M3 instance, we ignore any instance store volumes specified in the block device mapping for the AMI.

- * **DeviceName** (*string*) –

The device name exposed to the instance (for example, `/dev/sdh` or `xvdh`).

- * **Ebs** (*dict*) –

Parameters used to automatically set up EBS volumes when the instance is launched.

- **SnapshotId** (*string*) –

The ID of the snapshot.

- **VolumeSize** (*integer*) –

The size of the volume, in GiB.

Constraints: 1-1024 for standard volumes, 1-16384 for `gp2` volumes, and 4-16384 for `io1` volumes. If you specify a snapshot, the volume size must be equal to or larger than the snapshot size.

Default: If you're creating the volume from a snapshot and don't specify a volume size, the default is the snapshot size.

- **DeleteOnTermination** (*boolean*) –

Indicates whether the EBS volume is deleted on instance termination.

- **VolumeType** (*string*) –

The volume type. `gp2` for General Purpose (SSD) volumes,

`io1` for Provisioned IOPS (SSD) volumes, and `standard` for Magnetic volumes.

Default: `standard`

- **Iops** (*integer*) –

The number of I/O operations per second (IOPS) that the volume supports. For Provisioned IOPS (SSD) volumes, this represents the number of IOPS that are provisioned for the volume. For General Purpose (SSD) volumes, this represents the baseline performance of the volume and the rate at which the volume accumulates I/O credits for bursting. For more information on General Purpose (SSD) baseline performance, I/O credits, and bursting, see [Amazon EBS Volume Types](#) in the *Amazon Elastic Compute Cloud User Guide*.

Constraint: Range is 100 to 20000 for Provisioned IOPS (SSD) volumes and 3 to 10000 for General Purpose (SSD) volumes.

Condition: This parameter is required for requests to create `io1` volumes; it is not used in requests to create `standard` or `gp2` volumes.

- **Encrypted** (*boolean*) –

Indicates whether the EBS volume is encrypted. Encrypted Amazon EBS volumes may only be attached to instances that support Amazon EBS encryption.

- * **NoDevice** (*string*) –

Suppresses the specified device included in the block device mapping of the AMI.

- **VirtualizationType** (*string*) – The type of virtualization.

Default: `paravirtual`

- **SriovNetSupport** (*string*) – Set to `simple` to enable enhanced networking for the AMI and any instances that you launch from the AMI.

There is no way to disable enhanced networking at this time.

This option is supported only for HVM AMIs. Specifying this option with a PV AMI can make instances launched from the AMI unreachable.

Return type `dict`

Returns

Response Syntax

```
{
    'ImageId': 'string'
}
```

Response Structure

- (*dict*) –
 - **ImageId** (*string*) –

The ID of the newly registered AMI.

reject_vpc_peering_connection (**kwargs)

Rejects a VPC peering connection request. The VPC peering connection must be in the

pending-acceptance state. Use the `DescribeVpcPeeringConnections` request to view your outstanding VPC peering connection requests. To delete an active VPC peering connection, or to delete a VPC peering connection request that you initiated, use `DeleteVpcPeeringConnection`.

Request Syntax

```
response = client.reject_vpc_peering_connection(
    DryRun=True|False,
    VpcPeeringConnectionId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcPeeringConnectionId** (*string*) – **[REQUIRED]**

The ID of the VPC peering connection.

Return type dict

Returns

Response Syntax

```
{
    'Return': True|False
}
```

Response Structure

- (*dict*) –
 - **Return** (*boolean*) –

Returns `true` if the request succeeds; otherwise, it returns an error.

release_address (***kwargs*)

Releases the specified Elastic IP address.

After releasing an Elastic IP address, it is released to the IP address pool and might be unavailable to you. Be sure to update your DNS records and any servers or devices that communicate with the address. If you attempt to release an Elastic IP address that you already released, you'll get an `AuthFailure` error if the address is already allocated to another AWS account.

[EC2-Classic, default VPC] Releasing an Elastic IP address automatically disassociates it from any instance that it's associated with. To disassociate an Elastic IP address without releasing it, use `DisassociateAddress`.

[Nondefault VPC] You must use `DisassociateAddress` to disassociate the Elastic IP address before you try to release it. Otherwise, Amazon EC2 returns an error (`InvalidIpAddress.InUse`).

Request Syntax

```
response = client.release_address(
    DryRun=True|False,
    PublicIp='string',
    AllocationId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **PublicIp** (*string*) – [EC2-Classic] The Elastic IP address. Required for EC2-Classic.
- **AllocationId** (*string*) – [EC2-VPC] The allocation ID. Required for EC2-VPC.

Returns None

replace_network_acl_association (***kwargs*)

Changes which network ACL a subnet is associated with. By default when you create a subnet, it's automatically associated with the default network ACL. For more information about network ACLs, see [Network ACLs](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.replace_network_acl_association(
    DryRun=True|False,
    AssociationId='string',
    NetworkAclId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **AssociationId** (*string*) – **[REQUIRED]**
The ID of the current association between the original network ACL and the subnet.
- **NetworkAclId** (*string*) – **[REQUIRED]**
The ID of the new network ACL to associate with the subnet.

Return type dict

Returns

Response Syntax

```
{
    'NewAssociationId': 'string'
}
```

Response Structure

- (*dict*) –
 - **NewAssociationId** (*string*) –
The ID of the new association.

replace_network_acl_entry (***kwargs*)

Replaces an entry (rule) in a network ACL. For more information about network ACLs, see [Network ACLs](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.replace_network_acl_entry(
    DryRun=True|False,
    NetworkAclId='string',
    RuleNumber=123,
```

```
Protocol='string',
RuleAction='allow'|'deny',
Egress=True|False,
CidrBlock='string',
IcmpTypeCode={
    'Type': 123,
    'Code': 123
},
PortRange={
    'From': 123,
    'To': 123
}
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **NetworkAclId** (*string*) – **[REQUIRED]**
The ID of the ACL.
- **RuleNumber** (*integer*) – **[REQUIRED]**
The rule number of the entry to replace.
- **Protocol** (*string*) – **[REQUIRED]**
The IP protocol. You can specify `all` or `-1` to mean all protocols.
- **RuleAction** (*string*) – **[REQUIRED]**
Indicates whether to allow or deny the traffic that matches the rule.
- **Egress** (*boolean*) – **[REQUIRED]**
Indicates whether to replace the egress rule.
Default: If no value is specified, we replace the ingress rule.
- **CidrBlock** (*string*) – **[REQUIRED]**
The network range to allow or deny, in CIDR notation.
- **IcmpTypeCode** (*dict*) – ICMP protocol: The ICMP type and code. Required if specifying 1 (ICMP) for the protocol.
 - **Type** (*integer*) –
The ICMP code. A value of `-1` means all codes for the specified ICMP type.
 - **Code** (*integer*) –
The ICMP type. A value of `-1` means all types.
- **PortRange** (*dict*) – TCP or UDP protocols: The range of ports the rule applies to. Required if specifying 6 (TCP) or 17 (UDP) for the protocol.
 - **From** (*integer*) –
The first port in the range.
 - **To** (*integer*) –
The last port in the range.

Returns None

replace_route (***kwargs*)

Replaces an existing route within a route table in a VPC. You must provide only one of the following: Internet gateway or virtual private gateway, NAT instance, VPC peering connection, or network interface.

For more information about route tables, see [Route Tables](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.replace_route(
    DryRun=True|False,
    RouteTableId='string',
    DestinationCidrBlock='string',
    GatewayId='string',
    InstanceId='string',
    NetworkInterfaceId='string',
    VpcPeeringConnectionId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **RouteTableId** (*string*) – **[REQUIRED]**
The ID of the route table.
- **DestinationCidrBlock** (*string*) – **[REQUIRED]**
The CIDR address block used for the destination match. The value you provide must match the CIDR of an existing route in the table.
- **GatewayId** (*string*) – The ID of an Internet gateway or virtual private gateway.
- **InstanceId** (*string*) – The ID of a NAT instance in your VPC.
- **NetworkInterfaceId** (*string*) – The ID of a network interface.
- **VpcPeeringConnectionId** (*string*) – The ID of a VPC peering connection.

Returns None

replace_route_table_association (***kwargs*)

Changes the route table associated with a given subnet in a VPC. After the operation completes, the subnet uses the routes in the new route table it's associated with. For more information about route tables, see [Route Tables](#) in the *Amazon Virtual Private Cloud User Guide*.

You can also use `ReplaceRouteTableAssociation` to change which table is the main route table in the VPC. You just specify the main route table's association ID and the route table to be the new main route table.

Request Syntax

```
response = client.replace_route_table_association(
    DryRun=True|False,
    AssociationId='string',
    RouteTableId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **AssociationId** (*string*) – **[REQUIRED]**
The association ID.

- **RouteTableId** (*string*) – [REQUIRED]

The ID of the new route table to associate with the subnet.

Return type dict

Returns

Response Syntax

```
{
    'NewAssociationId': 'string'
}
```

Response Structure

- (*dict*) –
 - **NewAssociationId** (*string*) –
The ID of the new association.

report_instance_status (***kwargs*)

Submits feedback about the status of an instance. The instance must be in the `running` state. If your experience with the instance differs from the instance status returned by `DescribeInstanceState`, use `ReportInstanceState` to report your experience with the instance. Amazon EC2 collects this information to improve the accuracy of status checks.

Use of this action does not change the value returned by `DescribeInstanceState`.

Request Syntax

```
response = client.report_instance_status(
    DryRun=True|False,
    Instances=[
        'string',
    ],
    Status='ok'|'impaired',
    StartTime=datetime(2015, 1, 1),
    EndTime=datetime(2015, 1, 1),
    ReasonCodes=[
        'instance-stuck-in-state'|'unresponsive'|'not-accepting-credentials'|'password-not-
    ],
    Description='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Instances** (*list*) – [REQUIRED]
One or more instances.
 - (*string*) –
- **Status** (*string*) – [REQUIRED]
The status of all instances listed.
- **StartTime** (*datetime*) – The time at which the reported instance health state began.
- **EndTime** (*datetime*) – The time at which the reported instance health state ended.

- **ReasonCodes** (*list*) – [REQUIRED]

One or more reason codes that describes the health state of your instance.

- `instance-stuck-in-state` : My instance is stuck in a state.
- `unresponsive` : My instance is unresponsive.
- `not-accepting-credentials` : My instance is not accepting my credentials.
- `password-not-available` : A password is not available for my instance.
- `performance-network` : My instance is experiencing performance problems which I believe are network related.
- `performance-instance-store` : My instance is experiencing performance problems which I believe are related to the instance stores.
- `performance-ebs-volume` : My instance is experiencing performance problems which I believe are related to an EBS volume.
- `performance-other` : My instance is experiencing performance problems.
- `other` : [explain using the description parameter]
- (*string*) –

- **Description** (*string*) – Descriptive text about the health state of your instance.

Returns None

request_spot_fleet (***kwargs*)

Creates a Spot fleet request.

You can submit a single request that includes multiple launch specifications that vary by instance type, AMI, Availability Zone, or subnet.

By default, the Spot fleet requests Spot instances in the Spot pool where the price per unit is the lowest. Each launch specification can include its own instance weighting that reflects the value of the instance type to your application workload.

Alternatively, you can specify that the Spot fleet distribute the target capacity across the Spot pools included in its launch specifications. By ensuring that the Spot instances in your Spot fleet are in different Spot pools, you can improve the availability of your fleet.

For more information, see [Spot Fleet Requests](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.request_spot_fleet(
    DryRun=True|False,
    SpotFleetRequestConfig={
        'ClientToken': 'string',
        'SpotPrice': 'string',
        'TargetCapacity': 123,
        'ValidFrom': datetime(2015, 1, 1),
        'ValidUntil': datetime(2015, 1, 1),
        'TerminateInstancesWithExpiration': True|False,
        'IamFleetRole': 'string',
        'LaunchSpecifications': [
            {
                'ImageId': 'string',
                'KeyName': 'string',
                'SecurityGroups': [
                    {
                        'GroupName': 'string',
                        'GroupId': 'string'
```

```
    },
],
'UserData': 'string',
'AddressingType': 'string',
'InstanceType': 't1.micro'|'m1.small'|'m1.medium'|'m1.large'|'m1.xlarge'|'m
'Placement': {
    'AvailabilityZone': 'string',
    'GroupName': 'string'
},
'KernelId': 'string',
'RamdiskId': 'string',
'BlockDeviceMappings': [
    {
        'VirtualName': 'string',
        'DeviceName': 'string',
        'Ebs': {
            'SnapshotId': 'string',
            'VolumeSize': 123,
            'DeleteOnTermination': True|False,
            'VolumeType': 'standard'|'io1'|'gp2',
            'Iops': 123,
            'Encrypted': True|False
        },
        'NoDevice': 'string'
    },
],
'Monitoring': {
    'Enabled': True|False
},
'SubnetId': 'string',
'NetworkInterfaces': [
    {
        'NetworkInterfaceId': 'string',
        'DeviceIndex': 123,
        'SubnetId': 'string',
        'Description': 'string',
        'PrivateIpAddress': 'string',
        'Groups': [
            'string',
        ],
        'DeleteOnTermination': True|False,
        'PrivateIpAddresses': [
            {
                'PrivateIpAddress': 'string',
                'Primary': True|False
            },
        ],
        'SecondaryPrivateIpAddressCount': 123,
        'AssociatePublicIpAddress': True|False
    },
],
'IamInstanceProfile': {
    'Arn': 'string',
    'Name': 'string'
},
'EbsOptimized': True|False,
'WeightedCapacity': 123.0,
'SpotPrice': 'string'
```

```

    },
    ],
    'AllocationStrategy': 'lowestPrice'|'diversified'
}
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SpotFleetRequestConfig** (*dict*) – **[REQUIRED]**

The configuration for the Spot fleet request.

- **ClientToken** (*string*) –

A unique, case-sensitive identifier you provide to ensure idempotency of your listings. This helps avoid duplicate listings. For more information, see [Ensuring Idempotency](#).

- **SpotPrice** (*string*) – **[REQUIRED]**

The bid price per unit hour.

- **TargetCapacity** (*integer*) – **[REQUIRED]**

The number of units to request. You can choose to set the target capacity in terms of instances or a performance characteristic that is important to your application workload, such as vCPUs, memory, or I/O.

- **ValidFrom** (*datetime*) –

The start date and time of the request, in UTC format (for example, `YYYY-MM-DD T*HH* :MM :SS Z`). The default is to start fulfilling the request immediately.

- **ValidUntil** (*datetime*) –

The end date and time of the request, in UTC format (for example, `YYYY-MM-DD T*HH* :MM :SS Z`). At this point, no new Spot instance requests are placed or enabled to fulfill the request.

- **TerminateInstancesWithExpiration** (*boolean*) –

Indicates whether running Spot instances should be terminated when the Spot fleet request expires.

- **IamFleetRole** (*string*) – **[REQUIRED]**

Grants the Spot fleet permission to terminate Spot instances on your behalf when you cancel its Spot fleet request using `CancelSpotFleetRequests` or when the Spot fleet request expires, if you set `terminateInstancesWithExpiration`.

- **LaunchSpecifications** (*list*) – **[REQUIRED]**

Information about the launch specifications for the Spot fleet request.

* (*dict*) –

Describes the launch specification for one or more Spot instances.

- **ImageId** (*string*) –

The ID of the AMI.

- **KeyName** (*string*) –

The name of the key pair.

- **SecurityGroups** (*list*) –
One or more security groups. To request an instance in a non-default VPC, you must specify the ID of the security group. To request an instance in EC2-Classic or a default VPC, you can specify the name or the ID of the security group.
- (*dict*) –
Describes a security group.
- **GroupName** (*string*) –
The name of the security group.
- **GroupId** (*string*) –
The ID of the security group.
- **UserData** (*string*) –
The Base64-encoded MIME user data to make available to the instances.
- **AddressingType** (*string*) –
Deprecated.
- **InstanceType** (*string*) –
The instance type.
- **Placement** (*dict*) –
The placement information.
- **AvailabilityZone** (*string*) –
The Availability Zone.
- **GroupName** (*string*) –
The name of the placement group (for cluster instances).
- **KernelId** (*string*) –
The ID of the kernel.
- **RamdiskId** (*string*) –
The ID of the RAM disk.
- **BlockDeviceMappings** (*list*) –
One or more block device mapping entries.
- (*dict*) –
Describes a block device mapping.
- **VirtualName** (*string*) –
The virtual device name (ephemeral N). Instance store volumes are numbered starting from 0. An instance type with 2 available instance store volumes can specify mappings for `ephemeral0` and `ephemeral1`. The number of available instance store volumes depends on the instance type. After you connect to the instance, you must mount the volume.

Constraints: For M3 instances, you must specify instance store volumes in the block device mapping for the instance. When you launch an M3 instance, we ignore any instance store volumes specified in the block device mapping for the AMI.
- **DeviceName** (*string*) –

The device name exposed to the instance (for example, `/dev/sdh` or `xvdh`).

- **Ebs** (*dict*) –

Parameters used to automatically set up EBS volumes when the instance is launched.

- **SnapshotId** (*string*) –

The ID of the snapshot.

- **VolumeSize** (*integer*) –

The size of the volume, in GiB.

Constraints: 1-1024 for standard volumes, 1-16384 for gp2 volumes, and 4-16384 for io1 volumes. If you specify a snapshot, the volume size must be equal to or larger than the snapshot size.

Default: If you're creating the volume from a snapshot and don't specify a volume size, the default is the snapshot size.

- **DeleteOnTermination** (*boolean*) –

Indicates whether the EBS volume is deleted on instance termination.

- **VolumeType** (*string*) –

The volume type. `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, and `standard` for Magnetic volumes.

Default: `standard`

- **Iops** (*integer*) –

The number of I/O operations per second (IOPS) that the volume supports. For Provisioned IOPS (SSD) volumes, this represents the number of IOPS that are provisioned for the volume. For General Purpose (SSD) volumes, this represents the baseline performance of the volume and the rate at which the volume accumulates I/O credits for bursting. For more information on General Purpose (SSD) baseline performance, I/O credits, and bursting, see [Amazon EBS Volume Types](#) in the *Amazon Elastic Compute Cloud User Guide*.

Constraint: Range is 100 to 20000 for Provisioned IOPS (SSD) volumes and 3 to 10000 for General Purpose (SSD) volumes.

Condition: This parameter is required for requests to create `io1` volumes; it is not used in requests to create `standard` or `gp2` volumes.

- **Encrypted** (*boolean*) –

Indicates whether the EBS volume is encrypted. Encrypted Amazon EBS volumes may only be attached to instances that support Amazon EBS encryption.

- **NoDevice** (*string*) –

Suppresses the specified device included in the block device mapping of the AMI.

- **Monitoring** (*dict*) –
Enable or disable monitoring for the instances.
- **Enabled** (*boolean*) –
Enables monitoring for the instance.
Default: `false`
- **SubnetId** (*string*) –
The ID of the subnet in which to launch the instances.
- **NetworkInterfaces** (*list*) –
One or more network interfaces.
- (*dict*) –
Describes a network interface.
- **NetworkInterfaceId** (*string*) –
The ID of the network interface.
- **DeviceIndex** (*integer*) –
The index of the device on the instance for the network interface attachment. If you are specifying a network interface in a `RunInstances` request, you must provide the device index.
- **SubnetId** (*string*) –
The ID of the subnet associated with the network string. Applies only if creating a network interface when launching an instance.
- **Description** (*string*) –
The description of the network interface. Applies only if creating a network interface when launching an instance.
- **PrivateIpAddress** (*string*) –
The private IP address of the network interface. Applies only if creating a network interface when launching an instance.
- **Groups** (*list*) –
The IDs of the security groups for the network interface. Applies only if creating a network interface when launching an instance.
- (*string*) –
- **DeleteOnTermination** (*boolean*) –
If set to `true`, the interface is deleted when the instance is terminated. You can specify `true` only if creating a new network interface when launching an instance.
- **PrivateIpAddresses** (*list*) –
One or more private IP addresses to assign to the network interface. Only one private IP address can be designated as primary.
- (*dict*) –
Describes a secondary private IP address for a network interface.
- **PrivateIpAddress** (*string*) – **[REQUIRED]**
The private IP addresses.

- **Primary** (*boolean*) –

Indicates whether the private IP address is the primary private IP address. Only one IP address can be designated as primary.

- **SecondaryPrivateIpAddressCount** (*integer*) –

The number of secondary private IP addresses. You can't specify this option and specify more than one private IP address using the private IP addresses option.

- **AssociatePublicIpAddress** (*boolean*) –

Indicates whether to assign a public IP address to an instance you launch in a VPC. The public IP address can only be assigned to a network interface for eth0, and can only be assigned to a new network interface, not an existing one. You cannot specify more than one network interface in the request. If launching into a default subnet, the default value is `true`.

- **IamInstanceProfile** (*dict*) –

The IAM instance profile.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) of the instance profile.

- **Name** (*string*) –

The name of the instance profile.

- **EbsOptimized** (*boolean*) –

Indicates whether the instances are optimized for EBS I/O. This optimization provides dedicated throughput to Amazon EBS and an optimized configuration stack to provide optimal EBS I/O performance. This optimization isn't available with all instance types. Additional usage charges apply when using an EBS Optimized instance.

Default: `false`

- **WeightedCapacity** (*float*) –

The number of units provided by the specified instance type. These are the same units that you chose to set the target capacity in terms (instances or a performance characteristic such as vCPUs, memory, or I/O).

If the target capacity divided by this value is not a whole number, we round the number of instances to the next whole number. If this value is not specified, the default is 1.

- **SpotPrice** (*string*) –

The bid price per unit hour for the specified instance type. If this value is not specified, the default is the Spot bid price specified for the fleet. To determine the bid price per unit hour, divide the Spot bid price by the value of `WeightedCapacity`.

- **AllocationStrategy** (*string*) –

Determines how to allocate the target capacity across the Spot pools specified by the Spot fleet request. The default is `lowestPrice`.

Return type dict

Returns

Response Syntax

```
{
    'SpotFleetRequestId': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output of RequestSpotFleet.

- **SpotFleetRequestId** (*string*) –

The ID of the Spot fleet request.

request_spot_instances (***kwargs*)

Creates a Spot instance request. Spot instances are instances that Amazon EC2 launches when the bid price that you specify exceeds the current Spot price. Amazon EC2 periodically sets the Spot price based on available Spot Instance capacity and current Spot instance requests. For more information, see [Spot Instance Requests](#) in the *Amazon Elastic Compute Cloud User Guide* .

Request Syntax

```
response = client.request_spot_instances(
    DryRun=True|False,
    SpotPrice='string',
    ClientToken='string',
    InstanceCount=123,
    Type='one-time'|'persistent',
    ValidFrom=datetime(2015, 1, 1),
    ValidUntil=datetime(2015, 1, 1),
    LaunchGroup='string',
    AvailabilityZoneGroup='string',
    LaunchSpecification={
        'ImageId': 'string',
        'KeyName': 'string',
        'SecurityGroups': [
            'string',
        ],
        'UserData': 'string',
        'AddressingType': 'string',
        'InstanceType': 't1.micro'|'m1.small'|'m1.medium'|'m1.large'|'m1.xlarge'|'m3.medium',
        'Placement': {
            'AvailabilityZone': 'string',
            'GroupName': 'string'
        },
        'KernelId': 'string',
        'RamdiskId': 'string',
        'BlockDeviceMappings': [
            {
                'VirtualName': 'string',
                'DeviceName': 'string',
                'Ebs': {
                    'SnapshotId': 'string',
                    'VolumeSize': 123,
                    'DeleteOnTermination': True|False,
                    'VolumeType': 'standard'|'io1'|'gp2',
                    'Iops': 123,
                    'Encrypted': True|False
                }
            }
        ]
    },
    'KernelId': 'string',
    'RamdiskId': 'string',
    'BlockDeviceMappings': [
        {
            'VirtualName': 'string',
            'DeviceName': 'string',
            'Ebs': {
                'SnapshotId': 'string',
                'VolumeSize': 123,
                'DeleteOnTermination': True|False,
                'VolumeType': 'standard'|'io1'|'gp2',
                'Iops': 123,
                'Encrypted': True|False
            }
        }
    ]
}
```

```

        'NoDevice': 'string'
    },
],
'SubnetId': 'string',
'NetworkInterfaces': [
    {
        'NetworkInterfaceId': 'string',
        'DeviceIndex': 123,
        'SubnetId': 'string',
        'Description': 'string',
        'PrivateIpAddress': 'string',
        'Groups': [
            'string',
        ],
        'DeleteOnTermination': True|False,
        'PrivateIpAddresses': [
            {
                'PrivateIpAddress': 'string',
                'Primary': True|False
            },
        ],
        'SecondaryPrivateIpAddressCount': 123,
        'AssociatePublicIpAddress': True|False
    },
],
'IamInstanceProfile': {
    'Arn': 'string',
    'Name': 'string'
},
'EbsOptimized': True|False,
'Monitoring': {
    'Enabled': True|False
},
'SecurityGroupIds': [
    'string',
]
}
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SpotPrice** (*string*) – **[REQUIRED]**
The maximum hourly price (bid) for any Spot instance launched to fulfill the request.
- **ClientToken** (*string*) – Unique, case-sensitive identifier that you provide to ensure the idempotency of the request. For more information, see [How to Ensure Idempotency in the Amazon Elastic Compute Cloud User Guide](#).
- **InstanceCount** (*integer*) – The maximum number of Spot instances to launch.
Default: 1
- **Type** (*string*) – The Spot instance request type.
Default: one-time
- **ValidFrom** (*datetime*) – The start date of the request. If this is a one-time re-

quest, the request becomes active at this date and time and remains active until all instances launch, the request expires, or the request is canceled. If the request is persistent, the request becomes active at this date and time and remains active until it expires or is canceled.

Default: The request is effective indefinitely.

- **ValidUntil** (*datetime*) – The end date of the request. If this is a one-time request, the request remains active until all instances launch, the request is canceled, or this date is reached. If the request is persistent, it remains active until it is canceled or this date and time is reached.

Default: The request is effective indefinitely.

- **LaunchGroup** (*string*) – The instance launch group. Launch groups are Spot instances that launch together and terminate together.

Default: Instances are launched and terminated individually

- **AvailabilityZoneGroup** (*string*) – The user-specified name for a logical grouping of bids.

When you specify an Availability Zone group in a Spot Instance request, all Spot instances in the request are launched in the same Availability Zone. Instance proximity is maintained with this parameter, but the choice of Availability Zone is not. The group applies only to bids for Spot Instances of the same instance type. Any additional Spot instance requests that are specified with the same Availability Zone group name are launched in that same Availability Zone, as long as at least one instance from the group is still active.

If there is no active instance running in the Availability Zone group that you specify for a new Spot instance request (all instances are terminated, the bid is expired, or the bid falls below current market), then Amazon EC2 launches the instance in any Availability Zone where the constraint can be met. Consequently, the subsequent set of Spot instances could be placed in a different zone from the original request, even if you specified the same Availability Zone group.

Default: Instances are launched in any available Availability Zone.

- **LaunchSpecification** (*dict*) – Describes the launch specification for an instance.
 - **ImageId** (*string*) –
The ID of the AMI.
 - **KeyName** (*string*) –
The name of the key pair.
 - **SecurityGroups** (*list*) –
* (*string*) –
 - **UserData** (*string*) –
The Base64-encoded MIME user data to make available to the instances.
 - **AddressingType** (*string*) –
Deprecated.
 - **InstanceType** (*string*) –
The instance type.
 - **Placement** (*dict*) –
The placement information for the instance.
 - * **AvailabilityZone** (*string*) –
The Availability Zone.
 - * **GroupName** (*string*) –

- The name of the placement group (for cluster instances).
- **KernelId** (*string*) –
 - The ID of the kernel.
- **RamdiskId** (*string*) –
 - The ID of the RAM disk.
- **BlockDeviceMappings** (*list*) –
 - One or more block device mapping entries.
 - * (*dict*) –
 - Describes a block device mapping.
 - **VirtualName** (*string*) –
 - The virtual device name (ephemeral N). Instance store volumes are numbered starting from 0. An instance type with 2 available instance store volumes can specify mappings for `ephemeral0` and `ephemeral1`. The number of available instance store volumes depends on the instance type. After you connect to the instance, you must mount the volume.
 - Constraints: For M3 instances, you must specify instance store volumes in the block device mapping for the instance. When you launch an M3 instance, we ignore any instance store volumes specified in the block device mapping for the AMI.
 - **DeviceName** (*string*) –
 - The device name exposed to the instance (for example, `/dev/sdh` or `xvdh`).
 - **Ebs** (*dict*) –
 - Parameters used to automatically set up EBS volumes when the instance is launched.
 - **SnapshotId** (*string*) –
 - The ID of the snapshot.
 - **VolumeSize** (*integer*) –
 - The size of the volume, in GiB.
 - Constraints: 1-1024 for standard volumes, 1-16384 for `gp2` volumes, and 4-16384 for `io1` volumes. If you specify a snapshot, the volume size must be equal to or larger than the snapshot size.
 - Default: If you're creating the volume from a snapshot and don't specify a volume size, the default is the snapshot size.
 - **DeleteOnTermination** (*boolean*) –
 - Indicates whether the EBS volume is deleted on instance termination.
 - **VolumeType** (*string*) –
 - The volume type. `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, and `standard` for Magnetic volumes.
 - Default: `standard`

- **Iops** (*integer*) –

The number of I/O operations per second (IOPS) that the volume supports. For Provisioned IOPS (SSD) volumes, this represents the number of IOPS that are provisioned for the volume. For General Purpose (SSD) volumes, this represents the baseline performance of the volume and the rate at which the volume accumulates I/O credits for bursting. For more information on General Purpose (SSD) baseline performance, I/O credits, and bursting, see [Amazon EBS Volume Types](#) in the *Amazon Elastic Compute Cloud User Guide*.

Constraint: Range is 100 to 20000 for Provisioned IOPS (SSD) volumes and 3 to 10000 for General Purpose (SSD) volumes.

Condition: This parameter is required for requests to create `io1` volumes; it is not used in requests to create `standard` or `gp2` volumes.

- **Encrypted** (*boolean*) –

Indicates whether the EBS volume is encrypted. Encrypted Amazon EBS volumes may only be attached to instances that support Amazon EBS encryption.

- **NoDevice** (*string*) –

Suppresses the specified device included in the block device mapping of the AMI.

- **SubnetId** (*string*) –

The ID of the subnet in which to launch the instance.

- **NetworkInterfaces** (*list*) –

One or more network interfaces.

- * (*dict*) –

Describes a network interface.

- **NetworkInterfaceId** (*string*) –

The ID of the network interface.

- **DeviceIndex** (*integer*) –

The index of the device on the instance for the network interface attachment. If you are specifying a network interface in a `RunInstances` request, you must provide the device index.

- **SubnetId** (*string*) –

The ID of the subnet associated with the network string. Applies only if creating a network interface when launching an instance.

- **Description** (*string*) –

The description of the network interface. Applies only if creating a network interface when launching an instance.

- **PrivateIpAddress** (*string*) –

The private IP address of the network interface. Applies only if creating a network interface when launching an instance.

- **Groups** (*list*) –

The IDs of the security groups for the network interface. Applies only if creating a network interface when launching an instance.

- *(string)* –
- **DeleteOnTermination** (*boolean*) –

If set to `true`, the interface is deleted when the instance is terminated. You can specify `true` only if creating a new network interface when launching an instance.

- **PrivateIpAddresses** (*list*) –

One or more private IP addresses to assign to the network interface. Only one private IP address can be designated as primary.

- *(dict)* –

Describes a secondary private IP address for a network interface.

- **PrivateIpAddress** (*string*) – [REQUIRED]

The private IP addresses.

- **Primary** (*boolean*) –

Indicates whether the private IP address is the primary private IP address. Only one IP address can be designated as primary.

- **SecondaryPrivateIpAddressCount** (*integer*) –

The number of secondary private IP addresses. You can't specify this option and specify more than one private IP address using the private IP addresses option.

- **AssociatePublicIpAddress** (*boolean*) –

Indicates whether to assign a public IP address to an instance you launch in a VPC. The public IP address can only be assigned to a network interface for `eth0`, and can only be assigned to a new network interface, not an existing one. You cannot specify more than one network interface in the request. If launching into a default subnet, the default value is `true`.

- **IamInstanceProfile** (*dict*) –

The IAM instance profile.

- * **Arn** (*string*) –

The Amazon Resource Name (ARN) of the instance profile.

- * **Name** (*string*) –

The name of the instance profile.

- **EbsOptimized** (*boolean*) –

Indicates whether the instance is optimized for EBS I/O. This optimization provides dedicated throughput to Amazon EBS and an optimized configuration stack to provide optimal EBS I/O performance. This optimization isn't available with all instance types. Additional usage charges apply when using an EBS Optimized instance.

Default: `false`

- **Monitoring** (*dict*) –

Describes the monitoring for the instance.

- * **Enabled** (*boolean*) – [REQUIRED]

Indicates whether monitoring is enabled for the instance.

- **SecurityGroupIds** (*list*) –
* (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'SpotInstanceRequests': [
    {
      'SpotInstanceRequestId': 'string',
      'SpotPrice': 'string',
      'Type': 'one-time'|'persistent',
      'State': 'open'|'active'|'closed'|'cancelled'|'failed',
      'Fault': {
        'Code': 'string',
        'Message': 'string'
      },
      'Status': {
        'Code': 'string',
        'UpdateTime': datetime(2015, 1, 1),
        'Message': 'string'
      },
      'ValidFrom': datetime(2015, 1, 1),
      'ValidUntil': datetime(2015, 1, 1),
      'LaunchGroup': 'string',
      'AvailabilityZoneGroup': 'string',
      'LaunchSpecification': {
        'ImageId': 'string',
        'KeyName': 'string',
        'SecurityGroups': [
          {
            'GroupName': 'string',
            'GroupId': 'string'
          }
        ],
        'UserData': 'string',
        'AddressingType': 'string',
        'InstanceType': 't1.micro'|'m1.small'|'m1.medium'|'m1.large'|'m1.x',
        'Placement': {
          'AvailabilityZone': 'string',
          'GroupName': 'string'
        },
        'KernelId': 'string',
        'RamdiskId': 'string',
        'BlockDeviceMappings': [
          {
            'VirtualName': 'string',
            'DeviceName': 'string',
            'Ebs': {
              'SnapshotId': 'string',
              'VolumeSize': 123,
              'DeleteOnTermination': True|False,
              'VolumeType': 'standard'|'io1'|'gp2',
              'Iops': 123,
              'Encrypted': True|False
            }
          }
        ]
      }
    }
  ]
}
```



```

        'NoDevice': 'string'
    },
],
'SubnetId': 'string',
'NetworkInterfaces': [
    {
        'NetworkInterfaceId': 'string',
        'DeviceIndex': 123,
        'SubnetId': 'string',
        'Description': 'string',
        'PrivateIpAddress': 'string',
        'Groups': [
            'string',
        ],
        'DeleteOnTermination': True|False,
        'PrivateIpAddresses': [
            {
                'PrivateIpAddress': 'string',
                'Primary': True|False
            },
        ],
        'SecondaryPrivateIpAddressCount': 123,
        'AssociatePublicIpAddress': True|False
    },
],
'IamInstanceProfile': {
    'Arn': 'string',
    'Name': 'string'
},
'EbsOptimized': True|False,
'Monitoring': {
    'Enabled': True|False
}
},
'InstanceId': 'string',
'CreateTime': datetime(2015, 1, 1),
'ProductDescription': 'Linux/UNIX'|'Linux/UNIX (Amazon VPC)'|'Windows'
'Tags': [
    {
        'Key': 'string',
        'Value': 'string'
    },
],
'LaunchedAvailabilityZone': 'string'
},
]
}

```

Response Structure

- (dict) –

Contains the output of RequestSpotInstances.

- **SpotInstanceRequests** (list) –

One or more Spot instance requests.

- * (dict) –

Describe a Spot instance request.

- **SpotInstanceRequestId** (string) –

The ID of the Spot instance request.

- **SpotPrice** (*string*) –

The maximum hourly price (bid) for any Spot instance launched to fulfill the request.

- **Type** (*string*) –

The Spot instance request type.

- **State** (*string*) –

The state of the Spot instance request. Spot bid status information can help you track your Spot instance requests. For more information, see [Spot Bid Status](#) in the *Amazon Elastic Compute Cloud User Guide*.

- **Fault** (*dict*) –

The fault codes for the Spot instance request, if any.

- **Code** (*string*) –

The reason code for the Spot instance state change.

- **Message** (*string*) –

The message for the Spot instance state change.

- **Status** (*dict*) –

The status code and status message describing the Spot instance request.

- **Code** (*string*) –

The status code.

- **UpdateTime** (*datetime*) –

The date and time of the most recent status update, in UTC format (for example, *YYYY-MM-DD T*HH* :MM :SS Z*).

- **Message** (*string*) –

The description for the status code.

- **ValidFrom** (*datetime*) –

The start date of the request, in UTC format (for example, *YYYY-MM-DD T*HH* :MM :SS Z*). If this is a one-time request, the request becomes active at this date and time and remains active until all instances launch, the request expires, or the request is canceled. If the request is persistent, the request becomes active at this date and time and remains active until it expires or is canceled.

- **ValidUntil** (*datetime*) –

The end date of the request, in UTC format (for example, *YYYY-MM-DD T*HH* :MM :SS Z*). If this is a one-time request, the request remains active until all instances launch, the request is canceled, or this date is reached. If the request is persistent, it remains active until it is canceled or this date is reached.

- **LaunchGroup** (*string*) –

The instance launch group. Launch groups are Spot instances that launch together and terminate together.

- **AvailabilityZoneGroup** (*string*) –

The Availability Zone group. If you specify the same Availability Zone group for all Spot instance requests, all Spot instances are launched in the same Availability Zone.

- **LaunchSpecification** (*dict*) –

Additional information for launching instances.

- **ImageId** (*string*) –

The ID of the AMI.

- **KeyName** (*string*) –

The name of the key pair.

- **SecurityGroups** (*list*) –

One or more security groups. To request an instance in a non-default VPC, you must specify the ID of the security group. To request an instance in EC2-Classic or a default VPC, you can specify the name or the ID of the security group.

- (*dict*) –

Describes a security group.

- **GroupName** (*string*) –

The name of the security group.

- **GroupId** (*string*) –

The ID of the security group.

- **UserData** (*string*) –

The Base64-encoded MIME user data to make available to the instances.

- **AddressingType** (*string*) –

Deprecated.

- **InstanceType** (*string*) –

The instance type.

- **Placement** (*dict*) –

The placement information for the instance.

- **AvailabilityZone** (*string*) –

The Availability Zone.

- **GroupName** (*string*) –

The name of the placement group (for cluster instances).

- **KernelId** (*string*) –

The ID of the kernel.

- **RamdiskId** (*string*) –

The ID of the RAM disk.

- **BlockDeviceMappings** (*list*) –

One or more block device mapping entries.

- (*dict*) –

Describes a block device mapping.

- **VirtualName** (*string*) –

The virtual device name (ephemeral N). Instance store volumes are numbered starting from 0. An instance type with 2 available instance store volumes can specify mappings for

`ephemeral0` and `ephemeral1`. The number of available instance store volumes depends on the instance type. After you connect to the instance, you must mount the volume.

Constraints: For M3 instances, you must specify instance store volumes in the block device mapping for the instance. When you launch an M3 instance, we ignore any instance store volumes specified in the block device mapping for the AMI.

- **DeviceName** (*string*) –

The device name exposed to the instance (for example, `/dev/sdh` or `xvdh`).

- **Ebs** (*dict*) –

Parameters used to automatically set up EBS volumes when the instance is launched.

- **SnapshotId** (*string*) –

The ID of the snapshot.

- **VolumeSize** (*integer*) –

The size of the volume, in GiB.

Constraints: 1–1024 for standard volumes, 1–16384 for gp2 volumes, and 4–16384 for io1 volumes. If you specify a snapshot, the volume size must be equal to or larger than the snapshot size.

Default: If you're creating the volume from a snapshot and don't specify a volume size, the default is the snapshot size.

- **DeleteOnTermination** (*boolean*) –

Indicates whether the EBS volume is deleted on instance termination.

- **VolumeType** (*string*) –

The volume type. `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, and `standard` for Magnetic volumes.

Default: `standard`

- **Iops** (*integer*) –

The number of I/O operations per second (IOPS) that the volume supports. For Provisioned IOPS (SSD) volumes, this represents the number of IOPS that are provisioned for the volume. For General Purpose (SSD) volumes, this represents the baseline performance of the volume and the rate at which the volume accumulates I/O credits for bursting. For more information on General Purpose (SSD) baseline performance, I/O credits, and bursting, see [Amazon EBS Volume Types](#) in the *Amazon Elastic Compute Cloud User Guide*.

Constraint: Range is 100 to 20000 for Provisioned IOPS (SSD) volumes and 3 to 10000 for General Purpose (SSD) volumes.

Condition: This parameter is required for requests to create `io1` volumes; it is not used in requests to create `standard`

or gp2 volumes.

- **Encrypted** (*boolean*) –

Indicates whether the EBS volume is encrypted. Encrypted Amazon EBS volumes may only be attached to instances that support Amazon EBS encryption.

- **NoDevice** (*string*) –

Suppresses the specified device included in the block device mapping of the AMI.

- **SubnetId** (*string*) –

The ID of the subnet in which to launch the instance.

- **NetworkInterfaces** (*list*) –

One or more network interfaces.

- (*dict*) –

Describes a network interface.

- **NetworkInterfaceId** (*string*) –

The ID of the network interface.

- **DeviceIndex** (*integer*) –

The index of the device on the instance for the network interface attachment. If you are specifying a network interface in a RunInstances request, you must provide the device index.

- **SubnetId** (*string*) –

The ID of the subnet associated with the network string. Applies only if creating a network interface when launching an instance.

- **Description** (*string*) –

The description of the network interface. Applies only if creating a network interface when launching an instance.

- **PrivateIpAddress** (*string*) –

The private IP address of the network interface. Applies only if creating a network interface when launching an instance.

- **Groups** (*list*) –

The IDs of the security groups for the network interface. Applies only if creating a network interface when launching an instance.

- (*string*) –

- **DeleteOnTermination** (*boolean*) –

If set to `true`, the interface is deleted when the instance is terminated. You can specify `true` only if creating a new network interface when launching an instance.

- **PrivateIpAddresses** (*list*) –

One or more private IP addresses to assign to the network interface. Only one private IP address can be designated as primary.

- (*dict*) –

Describes a secondary private IP address for a network interface.

- **PrivateIpAddress** (*string*) –
The private IP addresses.
- **Primary** (*boolean*) –
Indicates whether the private IP address is the primary private IP address. Only one IP address can be designated as primary.
- **SecondaryPrivateIpAddressCount** (*integer*) –
The number of secondary private IP addresses. You can't specify this option and specify more than one private IP address using the private IP addresses option.
- **AssociatePublicIpAddress** (*boolean*) –
Indicates whether to assign a public IP address to an instance you launch in a VPC. The public IP address can only be assigned to a network interface for eth0, and can only be assigned to a new network interface, not an existing one. You cannot specify more than one network interface in the request. If launching into a default subnet, the default value is `true`.
- **IamInstanceProfile** (*dict*) –
The IAM instance profile.
- **Arn** (*string*) –
The Amazon Resource Name (ARN) of the instance profile.
- **Name** (*string*) –
The name of the instance profile.
- **EbsOptimized** (*boolean*) –
Indicates whether the instance is optimized for EBS I/O. This optimization provides dedicated throughput to Amazon EBS and an optimized configuration stack to provide optimal EBS I/O performance. This optimization isn't available with all instance types. Additional usage charges apply when using an EBS Optimized instance.

Default: `false`
- **Monitoring** (*dict*) –
Describes the monitoring for the instance.
- **Enabled** (*boolean*) –
Indicates whether monitoring is enabled for the instance.
- **InstanceId** (*string*) –
The instance ID, if an instance has been launched to fulfill the Spot instance request.
- **CreateTime** (*datetime*) –
The date and time when the Spot instance request was created, in UTC format (for example, `YYYY-MM-DD T*HH* :MM :SS Z`).
- **ProductDescription** (*string*) –
The product description associated with the Spot instance.
- **Tags** (*list*) –
Any tags assigned to the resource.

- *(dict)* –
Describes a tag.
- **Key** (*string*) –
The key of the tag.
Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`
- **Value** (*string*) –
The value of the tag.
Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.
- **LaunchedAvailabilityZone** (*string*) –
The Availability Zone in which the bid is launched.

reset_image_attribute (***kwargs*)

Resets an attribute of an AMI to its default value.

Note: The `productCodes` attribute can't be reset.

Request Syntax

```
response = client.reset_image_attribute(
    DryRun=True|False,
    ImageId='string',
    Attribute='launchPermission'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **ImageId** (*string*) – **[REQUIRED]**
The ID of the AMI.
- **Attribute** (*string*) – **[REQUIRED]**
The attribute to reset (currently you can only reset the launch permission attribute).

Returns None

reset_instance_attribute (***kwargs*)

Resets an attribute of an instance to its default value. To reset the `kernel` or `ramdisk`, the instance must be in a stopped state. To reset the `SourceDestCheck`, the instance can be either running or stopped.

The `SourceDestCheck` attribute controls whether source/destination checking is enabled. The default value is `true`, which means checking is enabled. This value must be `false` for a NAT instance to perform NAT. For more information, see [NAT Instances](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = client.reset_instance_attribute(
    DryRun=True|False,
```

```
InstanceId='string',
Attribute='instanceType'|'kernel'|'ramdisk'|'userData'|'disableApiTermination'|'instanc
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceId** (*string*) – [REQUIRED]

The ID of the instance.

- **Attribute** (*string*) – [REQUIRED]

The attribute to reset.

Returns None

reset_network_interface_attribute (**kwargs)

Resets a network interface attribute. You can specify only one attribute at a time.

Request Syntax

```
response = client.reset_network_interface_attribute(
    DryRun=True|False,
    NetworkInterfaceId='string',
    SourceDestCheck='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **NetworkInterfaceId** (*string*) – [REQUIRED]

The ID of the network interface.

- **SourceDestCheck** (*string*) – The source/destination checking attribute. Resets the value to `true`.

Returns None

reset_snapshot_attribute (**kwargs)

Resets permission settings for the specified snapshot.

For more information on modifying snapshot permissions, see [Sharing Snapshots](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.reset_snapshot_attribute(
    DryRun=True|False,
    SnapshotId='string',
    Attribute='productCodes'|'createVolumePermission'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **SnapshotId** (*string*) – [REQUIRED]

The ID of the snapshot.

- **Attribute** (*string*) – [REQUIRED]

The attribute to reset. Currently, only the attribute for permission to create volumes can be reset.

Returns None

restore_address_to_classic (***kwargs*)

Restores an Elastic IP address that was previously moved to the EC2-VPC platform back to the EC2-Classic platform. You cannot move an Elastic IP address that was originally allocated for use in EC2-VPC. The Elastic IP address must not be associated with an instance or network interface.

Request Syntax

```
response = client.restore_address_to_classic(
    DryRun=True|False,
    PublicIp='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **PublicIp** (*string*) – [REQUIRED]

The Elastic IP address.

Return type dict

Returns

Response Syntax

```
{
    'Status': 'MoveInProgress'|'InVpc'|'InClassic',
    'PublicIp': 'string'
}
```

Response Structure

- (*dict*) –
 - **Status** (*string*) –

The move status for the IP address.
 - **PublicIp** (*string*) –

The Elastic IP address.

revoke_security_group_egress (***kwargs*)

Removes one or more egress rules from a security group for EC2-VPC. The values that you specify in the revoke request (for example, ports) must match the existing rule's values for the rule to be revoked.

Each rule consists of the protocol and the CIDR range or source security group. For the TCP and UDP protocols, you must also specify the destination port or range of ports. For the ICMP protocol, you must also specify the ICMP type and code.

Rule changes are propagated to instances within the security group as quickly as possible. However, a small delay might occur.

Request Syntax

```
response = client.revoke_security_group_egress(
    DryRun=True|False,
    GroupId='string',
    SourceSecurityGroupName='string',
    SourceSecurityGroupOwnerId='string',
    IpProtocol='string',
    FromPort=123,
    ToPort=123,
    CidrIp='string',
    IpPermissions=[
        {
            'IpProtocol': 'string',
            'FromPort': 123,
            'ToPort': 123,
            'UserIdGroupPairs': [
                {
                    'UserId': 'string',
                    'GroupName': 'string',
                    'GroupId': 'string'
                },
            ],
            'IpRanges': [
                {
                    'CidrIp': 'string'
                },
            ],
            'PrefixListIds': [
                {
                    'PrefixListId': 'string'
                },
            ],
        },
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **GroupId** (*string*) – **[REQUIRED]**
The ID of the security group.
- **SourceSecurityGroupName** (*string*) – The name of a destination security group. To revoke outbound access to a destination security group, we recommend that you use a set of IP permissions instead.
- **SourceSecurityGroupOwnerId** (*string*) – The AWS account number for a destination security group. To revoke outbound access to a destination security group, we recommend that you use a set of IP permissions instead.
- **IpProtocol** (*string*) – The IP protocol name (`tcp`, `udp`, `icmp`) or number (see [Protocol Numbers](#)). Use `-1` to specify all.
- **FromPort** (*integer*) – The start of port range for the TCP and UDP protocols, or an ICMP type number. For the ICMP type number, use `-1` to specify all ICMP types.
- **ToPort** (*integer*) – The end of port range for the TCP and UDP protocols, or an ICMP code number. For the ICMP code number, use `-1` to specify all ICMP

codes for the ICMP type.

- **CidrIp** (*string*) – The CIDR IP address range. You can't specify this parameter when specifying a source security group.
- **IpPermissions** (*list*) – A set of IP permissions. You can't specify a destination security group and a CIDR IP address range.
 - (*dict*) –

Describes a security group rule.

- * **IpProtocol** (*string*) –

The protocol.

When you call `DescribeSecurityGroups`, the protocol value returned is the number. Exception: For TCP, UDP, and ICMP, the value returned is the name (for example, `tcp`, `udp`, or `icmp`). For a list of protocol numbers, see [Protocol Numbers](#). (VPC only) When you call `AuthorizeSecurityGroupIngress`, you can use `-1` to specify all.

- * **FromPort** (*integer*) –

The start of port range for the TCP and UDP protocols, or an ICMP type number. A value of `-1` indicates all ICMP types.

- * **ToPort** (*integer*) –

The end of port range for the TCP and UDP protocols, or an ICMP code. A value of `-1` indicates all ICMP codes for the specified ICMP type.

- * **UserIdGroupPairs** (*list*) –

One or more security group and AWS account ID pairs.

- (*dict*) –

Describes a security group and AWS account ID pair.

- **UserId** (*string*) –

The ID of an AWS account. EC2-Classic only.

- **GroupName** (*string*) –

The name of the security group. In a request, use this parameter for a security group in EC2-Classic or a default VPC only. For a security group in a nondefault VPC, use `GroupId`.

- **GroupId** (*string*) –

The ID of the security group.

- * **IpRanges** (*list*) –

One or more IP ranges.

- (*dict*) –

Describes an IP range.

- **CidrIp** (*string*) –

The CIDR range. You can either specify a CIDR range or a source security group, not both.

- * **PrefixListIds** (*list*) –

(Valid for `AuthorizeSecurityGroupEgress`, `RevokeSecurityGroupEgress` and `DescribeSecurityGroups` only) One or more prefix list IDs for an AWS service. In an `AuthorizeSecurityGroupEgress` request, this is the AWS service that you want to access through a VPC endpoint from instances associated with the security group.

- (*dict*) –
The ID of the prefix.
- **PrefixListId** (*string*) –
The ID of the prefix.

Returns None

revoke_security_group_ingress (***kwargs*)

Removes one or more ingress rules from a security group. The values that you specify in the revoke request (for example, ports) must match the existing rule's values for the rule to be removed.

Each rule consists of the protocol and the CIDR range or source security group. For the TCP and UDP protocols, you must also specify the destination port or range of ports. For the ICMP protocol, you must also specify the ICMP type and code.

Rule changes are propagated to instances within the security group as quickly as possible. However, a small delay might occur.

Request Syntax

```
response = client.revoke_security_group_ingress(
    DryRun=True|False,
    GroupName='string',
    GroupId='string',
    SourceSecurityGroupName='string',
    SourceSecurityGroupOwnerId='string',
    IpProtocol='string',
    FromPort=123,
    ToPort=123,
    CidrIp='string',
    IpPermissions=[
        {
            'IpProtocol': 'string',
            'FromPort': 123,
            'ToPort': 123,
            'UserIdGroupPairs': [
                {
                    'UserId': 'string',
                    'GroupName': 'string',
                    'GroupId': 'string'
                },
            ],
            'IpRanges': [
                {
                    'CidrIp': 'string'
                },
            ],
            'PrefixListIds': [
                {
                    'PrefixListId': 'string'
                },
            ],
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If

you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **GroupName** (*string*) – [EC2-Classic, default VPC] The name of the security group.
- **GroupId** (*string*) – The ID of the security group. Required for a security group in a nondefault VPC.
- **SourceSecurityGroupName** (*string*) – [EC2-Classic, default VPC] The name of the source security group. You can't specify this parameter in combination with the following parameters: the CIDR IP address range, the start of the port range, the IP protocol, and the end of the port range. For EC2-VPC, the source security group must be in the same VPC.
- **SourceSecurityGroupOwnerId** (*string*) – [EC2-Classic, default VPC] The AWS account ID of the source security group. For EC2-VPC, the source security group must be in the same VPC. You can't specify this parameter in combination with the following parameters: the CIDR IP address range, the IP protocol, the start of the port range, and the end of the port range. To revoke a specific rule for an IP protocol and port range, use a set of IP permissions instead.
- **IpProtocol** (*string*) – The IP protocol name (`tcp`, `udp`, `icmp`) or number (see [Protocol Numbers](#)). Use `-1` to specify all.
- **FromPort** (*integer*) – The start of port range for the TCP and UDP protocols, or an ICMP type number. For the ICMP type number, use `-1` to specify all ICMP types.
- **ToPort** (*integer*) – The end of port range for the TCP and UDP protocols, or an ICMP code number. For the ICMP code number, use `-1` to specify all ICMP codes for the ICMP type.
- **CidrIp** (*string*) – The CIDR IP address range. You can't specify this parameter when specifying a source security group.
- **IpPermissions** (*list*) – A set of IP permissions. You can't specify a source security group and a CIDR IP address range.
 - (*dict*) –

Describes a security group rule.

- * **IpProtocol** (*string*) –

The protocol.

When you call `DescribeSecurityGroups`, the protocol value returned is the number. Exception: For TCP, UDP, and ICMP, the value returned is the name (for example, `tcp`, `udp`, or `icmp`). For a list of protocol numbers, see [Protocol Numbers](#). (VPC only) When you call `AuthorizeSecurityGroupIngress`, you can use `-1` to specify all.

- * **FromPort** (*integer*) –

The start of port range for the TCP and UDP protocols, or an ICMP type number. A value of `-1` indicates all ICMP types.

- * **ToPort** (*integer*) –

The end of port range for the TCP and UDP protocols, or an ICMP code. A value of `-1` indicates all ICMP codes for the specified ICMP type.

- * **UserIdGroupPairs** (*list*) –

One or more security group and AWS account ID pairs.

- (*dict*) –

Describes a security group and AWS account ID pair.

- **UserId** (*string*) –

The ID of an AWS account. EC2-Classic only.

- **GroupName** (*string*) –

The name of the security group. In a request, use this parameter for a security group in EC2-Classic or a default VPC only. For a security group in a nondefault VPC, use `GroupId`.

- **GroupId** (*string*) –

The ID of the security group.

- * **IpRanges** (*list*) –

One or more IP ranges.

- (*dict*) –

Describes an IP range.

- **CidrIp** (*string*) –

The CIDR range. You can either specify a CIDR range or a source security group, not both.

- * **PrefixListIds** (*list*) –

(Valid for `AuthorizeSecurityGroupEgress`, `RevokeSecurityGroupEgress` and `DescribeSecurityGroups` only) One or more prefix list IDs for an AWS service. In an `AuthorizeSecurityGroupEgress` request, this is the AWS service that you want to access through a VPC endpoint from instances associated with the security group.

- (*dict*) –

The ID of the prefix.

- **PrefixListId** (*string*) –

The ID of the prefix.

Returns None

run_instances (***kwargs*)

Launches the specified number of instances using an AMI for which you have permissions.

When you launch an instance, it enters the `pending` state. After the instance is ready for you, it enters the `running` state. To check the state of your instance, call `DescribeInstances`.

If you don't specify a security group when launching an instance, Amazon EC2 uses the default security group. For more information, see [Security Groups](#) in the *Amazon Elastic Compute Cloud User Guide*.

[EC2-VPC only accounts] If you don't specify a subnet in the request, we choose a default subnet from your default VPC for you.

[EC2-Classic accounts] If you're launching into EC2-Classic and you don't specify an Availability Zone, we choose one for you.

Linux instances have access to the public key of the key pair at boot. You can use this key to provide secure access to the instance. Amazon EC2 public images use this feature to provide secure access without passwords. For more information, see [Key Pairs](#) in the *Amazon Elastic Compute Cloud User Guide*.

You can provide optional user data when launching an instance. For more information, see [Instance Metadata](#) in the *Amazon Elastic Compute Cloud User Guide*.

If any of the AMIs have a product code attached for which the user has not subscribed, `RunInstances` fails.

T2 instance types can only be launched into a VPC. If you do not have a default VPC, or if you do not specify a subnet ID in the request, `RunInstances` fails.

For more information about troubleshooting, see [What To Do If An Instance Immediately Terminates](#) , and [Troubleshooting Connecting to Your Instance](#) in the *Amazon Elastic Compute Cloud User Guide* .

Request Syntax

```
response = client.run_instances(
    DryRun=True|False,
    ImageId='string',
    MinCount=123,
    MaxCount=123,
    KeyName='string',
    SecurityGroups=[
        'string',
    ],
    SecurityGroupIds=[
        'string',
    ],
    UserData='string',
    InstanceType='t1.micro'|'m1.small'|'m1.medium'|'m1.large'|'m1.xlarge'|'m3.medium'|'m3.1
    Placement={
        'AvailabilityZone': 'string',
        'GroupName': 'string',
        'Tenancy': 'default'|'dedicated'
    },
    KernelId='string',
    RamdiskId='string',
    BlockDeviceMappings=[
        {
            'VirtualName': 'string',
            'DeviceName': 'string',
            'Ebs': {
                'SnapshotId': 'string',
                'VolumeSize': 123,
                'DeleteOnTermination': True|False,
                'VolumeType': 'standard'|'io1'|'gp2',
                'Iops': 123,
                'Encrypted': True|False
            },
            'NoDevice': 'string'
        },
    ],
    Monitoring={
        'Enabled': True|False
    },
    SubnetId='string',
    DisableApiTermination=True|False,
    InstanceInitiatedShutdownBehavior='stop'|'terminate',
    PrivateIpAddress='string',
    ClientToken='string',
    AdditionalInfo='string',
    NetworkInterfaces=[
        {
            'NetworkInterfaceId': 'string',
            'DeviceIndex': 123,
            'SubnetId': 'string',
            'Description': 'string',
```

```
        'PrivateIpAddress': 'string',
        'Groups': [
            'string',
        ],
        'DeleteOnTermination': True|False,
        'PrivateIpAddresses': [
            {
                'PrivateIpAddress': 'string',
                'Primary': True|False
            },
        ],
        'SecondaryPrivateIpAddressCount': 123,
        'AssociatePublicIpAddress': True|False
    },
],
IamInstanceProfile={
    'Arn': 'string',
    'Name': 'string'
},
EbsOptimized=True|False
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **ImageId** (*string*) – **[REQUIRED]**
The ID of the AMI, which you can get by calling `DescribeImages`.
- **MinCount** (*integer*) – **[REQUIRED]**
The minimum number of instances to launch. If you specify a minimum that is more instances than Amazon EC2 can launch in the target Availability Zone, Amazon EC2 launches no instances.

Constraints: Between 1 and the maximum number you’re allowed for the specified instance type. For more information about the default limits, and how to request an increase, see [How many instances can I run in Amazon EC2](#) in the Amazon EC2 General FAQ.
- **MaxCount** (*integer*) – **[REQUIRED]**
The maximum number of instances to launch. If you specify more instances than Amazon EC2 can launch in the target Availability Zone, Amazon EC2 launches the largest possible number of instances above `MinCount`.

Constraints: Between 1 and the maximum number you’re allowed for the specified instance type. For more information about the default limits, and how to request an increase, see [How many instances can I run in Amazon EC2](#) in the Amazon EC2 General FAQ.
- **KeyName** (*string*) – The name of the key pair. You can create a key pair using `CreateKeyPair` or `ImportKeyPair`.

Warning: If you do not specify a key pair, you can’t connect to the instance unless you choose an AMI that is configured to allow users another way to log in.

- **SecurityGroups** (*list*) – [EC2-Classic, default VPC] One or more security group names. For a nondefault VPC, you must use security group IDs instead.

Default: Amazon EC2 uses the default security group.

– (*string*) –

- **SecurityGroupIds** (*list*) – One or more security group IDs. You can create a security group using `CreateSecurityGroup`.

Default: Amazon EC2 uses the default security group.

– (*string*) –

- **UserData** (*string*) – The Base64-encoded MIME user data for the instances.

This value will be base64 encoded automatically. Do not base64 encode this value prior to performing the operation.

- **InstanceType** (*string*) – The instance type. For more information, see [Instance Types](#) in the *Amazon Elastic Compute Cloud User Guide*.

Default: `m1.small`

- **Placement** (*dict*) – The placement for the instance.

– **AvailabilityZone** (*string*) –

The Availability Zone of the instance.

– **GroupName** (*string*) –

The name of the placement group the instance is in (for cluster compute instances).

– **Tenancy** (*string*) –

The tenancy of the instance (if the instance is running in a VPC). An instance with a tenancy of `dedicated` runs on single-tenant hardware.

- **KernelId** (*string*) – The ID of the kernel.

Warning: We recommend that you use PV-GRUB instead of kernels and RAM disks. For more information, see [PV-GRUB](#) in the *Amazon Elastic Compute Cloud User Guide*.

- **RamdiskId** (*string*) – The ID of the RAM disk.

Warning: We recommend that you use PV-GRUB instead of kernels and RAM disks. For more information, see [PV-GRUB](#) in the *Amazon Elastic Compute Cloud User Guide*.

- **BlockDeviceMappings** (*list*) – The block device mapping.

– (*dict*) –

Describes a block device mapping.

* **VirtualName** (*string*) –

The virtual device name (`ephemeral N`). Instance store volumes are numbered starting from 0. An instance type with 2 available instance store volumes can specify mappings for `ephemeral0` and `ephemeral1`. The number of available instance store volumes depends on the instance type. After you connect to the instance, you must mount the volume.

Constraints: For M3 instances, you must specify instance store volumes in the block device mapping for the instance. When you launch an M3 instance, we ignore any instance store volumes specified in the block device mapping for the AMI.

* **DeviceName** (*string*) –

The device name exposed to the instance (for example, `/dev/sdh` or `xvdh`).

* **Ebs** (*dict*) –

Parameters used to automatically set up EBS volumes when the instance is launched.

· **SnapshotId** (*string*) –

The ID of the snapshot.

· **VolumeSize** (*integer*) –

The size of the volume, in GiB.

Constraints: 1-1024 for standard volumes, 1-16384 for gp2 volumes, and 4-16384 for io1 volumes. If you specify a snapshot, the volume size must be equal to or larger than the snapshot size.

Default: If you're creating the volume from a snapshot and don't specify a volume size, the default is the snapshot size.

· **DeleteOnTermination** (*boolean*) –

Indicates whether the EBS volume is deleted on instance termination.

· **VolumeType** (*string*) –

The volume type. `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, and `standard` for Magnetic volumes.

Default: `standard`

· **Iops** (*integer*) –

The number of I/O operations per second (IOPS) that the volume supports. For Provisioned IOPS (SSD) volumes, this represents the number of IOPS that are provisioned for the volume. For General Purpose (SSD) volumes, this represents the baseline performance of the volume and the rate at which the volume accumulates I/O credits for bursting. For more information on General Purpose (SSD) baseline performance, I/O credits, and bursting, see [Amazon EBS Volume Types](#) in the *Amazon Elastic Compute Cloud User Guide*.

Constraint: Range is 100 to 20000 for Provisioned IOPS (SSD) volumes and 3 to 10000 for General Purpose (SSD) volumes.

Condition: This parameter is required for requests to create `io1` volumes; it is not used in requests to create `standard` or `gp2` volumes.

· **Encrypted** (*boolean*) –

Indicates whether the EBS volume is encrypted. Encrypted Amazon EBS volumes may only be attached to instances that support Amazon EBS encryption.

* **NoDevice** (*string*) –

Suppresses the specified device included in the block device mapping of the AMI.

- **Monitoring** (*dict*) – The monitoring for the instance.
 - **Enabled** (*boolean*) – [REQUIRED]

Indicates whether monitoring is enabled for the instance.

- **SubnetId** (*string*) – [EC2-VPC] The ID of the subnet to launch the instance into.
- **DisableApiTermination** (*boolean*) – If you set this parameter to `true`, you can't terminate the instance using the Amazon EC2 console, CLI, or API; otherwise, you can. If you set this parameter to `true` and then later want to be able to terminate the instance, you must first change the value of the `disableApiTermination` attribute to `false` using `ModifyInstanceAttribute`. Alternatively, if you set `InstanceInitiatedShutdownBehavior` to `terminate`, you can terminate the instance by running the shutdown command from the instance.

Default: `false`

- **InstanceInitiatedShutdownBehavior** (*string*) – Indicates whether an instance stops or terminates when you initiate shutdown from the instance (using the operating system command for system shutdown).

Default: `stop`

- **PrivateIpAddress** (*string*) – [EC2-VPC] The primary IP address. You must specify a value from the IP address range of the subnet.

Only one private IP address can be designated as primary. Therefore, you can't specify this parameter if `PrivateIpAddresses.n.Primary` is set to `true` and `PrivateIpAddresses.n.PrivateIpAddress` is set to an IP address.

Default: We select an IP address from the IP address range of the subnet.

- **ClientToken** (*string*) – Unique, case-sensitive identifier you provide to ensure the idempotency of the request. For more information, see [Ensuring Idempotency](#).

Constraints: Maximum 64 ASCII characters

- **AdditionalInfo** (*string*) – Reserved.
- **NetworkInterfaces** (*list*) – One or more network interfaces.
 - (*dict*) –

Describes a network interface.

- * **NetworkInterfaceId** (*string*) –

The ID of the network interface.

- * **DeviceIndex** (*integer*) –

The index of the device on the instance for the network interface attachment. If you are specifying a network interface in a `RunInstances` request, you must provide the device index.

- * **SubnetId** (*string*) –

The ID of the subnet associated with the network string. Applies only if creating a network interface when launching an instance.

- * **Description** (*string*) –

The description of the network interface. Applies only if creating a network interface when launching an instance.

- * **PrivateIpAddress** (*string*) –

The private IP address of the network interface. Applies only if creating a network interface when launching an instance.

- * **Groups** (*list*) –

The IDs of the security groups for the network interface. Applies only if creating a network interface when launching an instance.

- (*string*) –

- * **DeleteOnTermination** (*boolean*) –

If set to `true`, the interface is deleted when the instance is terminated. You can specify `true` only if creating a new network interface when launching an instance.

- * **PrivateIpAddresses** (*list*) –

One or more private IP addresses to assign to the network interface. Only one private IP address can be designated as primary.

- (*dict*) –

Describes a secondary private IP address for a network interface.

- **PrivateIpAddress** (*string*) – [REQUIRED]

The private IP addresses.

- **Primary** (*boolean*) –

Indicates whether the private IP address is the primary private IP address. Only one IP address can be designated as primary.

- * **SecondaryPrivateIpAddressCount** (*integer*) –

The number of secondary private IP addresses. You can't specify this option and specify more than one private IP address using the private IP addresses option.

- * **AssociatePublicIpAddress** (*boolean*) –

Indicates whether to assign a public IP address to an instance you launch in a VPC. The public IP address can only be assigned to a network interface for `eth0`, and can only be assigned to a new network interface, not an existing one. You cannot specify more than one network interface in the request. If launching into a default subnet, the default value is `true`.

- **IamInstanceProfile** (*dict*) – The IAM instance profile.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) of the instance profile.

- **Name** (*string*) –

The name of the instance profile.

- **EbsOptimized** (*boolean*) – Indicates whether the instance is optimized for EBS I/O. This optimization provides dedicated throughput to Amazon EBS and an optimized configuration stack to provide optimal EBS I/O performance. This optimization isn't available with all instance types. Additional usage charges apply when using an EBS-optimized instance.

Default: `false`

Return type `dict`

Returns

Response Syntax

```
{
    'ReservationId': 'string',
    'OwnerId': 'string',
```

```

'RequesterId': 'string',
'Groups': [
    {
        'GroupName': 'string',
        'GroupId': 'string'
    },
],
'Instances': [
    {
        'InstanceId': 'string',
        'ImageId': 'string',
        'State': {
            'Code': 123,
            'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stopping'
        },
        'PrivateDnsName': 'string',
        'PublicDnsName': 'string',
        'StateTransitionReason': 'string',
        'KeyName': 'string',
        'AmiLaunchIndex': 123,
        'ProductCodes': [
            {
                'ProductCodeId': 'string',
                'ProductCodeType': 'devpay'|'marketplace'
            },
        ],
        'InstanceType': 't1.micro'|'m1.small'|'m1.medium'|'m1.large'|'m1.xlarge',
        'LaunchTime': datetime(2015, 1, 1),
        'Placement': {
            'AvailabilityZone': 'string',
            'GroupName': 'string',
            'Tenancy': 'default'|'dedicated'
        },
        'KernelId': 'string',
        'RamdiskId': 'string',
        'Platform': 'Windows',
        'Monitoring': {
            'State': 'disabled'|'disabling'|'enabled'|'pending'
        },
        'SubnetId': 'string',
        'VpcId': 'string',
        'PrivateIpAddress': 'string',
        'PublicIpAddress': 'string',
        'StateReason': {
            'Code': 'string',
            'Message': 'string'
        },
        'Architecture': 'i386'|'x86_64',
        'RootDeviceType': 'ebs'|'instance-store',
        'RootDeviceName': 'string',
        'BlockDeviceMappings': [
            {
                'DeviceName': 'string',
                'Ebs': {
                    'VolumeId': 'string',
                    'Status': 'attaching'|'attached'|'detaching'|'detached',
                    'AttachTime': datetime(2015, 1, 1),
                    'DeleteOnTermination': True|False
                }
            },
        ],
    },
],

```

```
        },
    },
],
'VirtualizationType': 'hvm'|'paravirtual',
'InstanceLifecycle': 'spot',
'SpotInstanceRequestId': 'string',
'ClientToken': 'string',
'Tags': [
    {
        'Key': 'string',
        'Value': 'string'
    },
],
'SecurityGroups': [
    {
        'GroupName': 'string',
        'GroupId': 'string'
    },
],
'SourceDestCheck': True|False,
'Hypervisor': 'ovm'|'xen',
'NetworkInterfaces': [
    {
        'NetworkInterfaceId': 'string',
        'SubnetId': 'string',
        'VpcId': 'string',
        'Description': 'string',
        'OwnerId': 'string',
        'Status': 'available'|'attaching'|'in-use'|'detaching',
        'MacAddress': 'string',
        'PrivateIpAddress': 'string',
        'PrivateDnsName': 'string',
        'SourceDestCheck': True|False,
        'Groups': [
            {
                'GroupName': 'string',
                'GroupId': 'string'
            },
        ],
    },
],
'Attachment': {
    'AttachmentId': 'string',
    'DeviceIndex': 123,
    'Status': 'attaching'|'attached'|'detaching'|'detached',
    'AttachTime': datetime(2015, 1, 1),
    'DeleteOnTermination': True|False
},
'Association': {
    'PublicIp': 'string',
    'PublicDnsName': 'string',
    'IpOwnerId': 'string'
},
'PrivateIpAddresses': [
    {
        'PrivateIpAddress': 'string',
        'PrivateDnsName': 'string',
        'Primary': True|False,
        'Association': {
            'PublicIp': 'string',
```

```

        'PublicDnsName': 'string',
        'IpOwnerId': 'string'
    },
    ],
    },
    'IamInstanceProfile': {
        'Arn': 'string',
        'Id': 'string'
    },
    'EbsOptimized': True|False,
    'SriovNetSupport': 'string'
},
]
}

```

Response Structure

- *(dict)* –

One or more reservations.

- **ReservationId** (*string*) –

The ID of the reservation.

- **OwnerId** (*string*) –

The ID of the AWS account that owns the reservation.

- **RequesterId** (*string*) –

The ID of the requester that launched the instances on your behalf (for example, AWS Management Console or Auto Scaling).

- **Groups** (*list*) –

One or more security groups.

- * *(dict)* –

Describes a security group.

- **GroupName** (*string*) –

The name of the security group.

- **GroupId** (*string*) –

The ID of the security group.

- **Instances** (*list*) –

One or more instances.

- * *(dict)* –

Describes an instance.

- **InstanceId** (*string*) –

The ID of the instance.

- **ImageId** (*string*) –

The ID of the AMI used to launch the instance.

- **State** (*dict*) –

The current state of the instance.

- **Code** (*integer*) –

The low byte represents the state. The high byte is an opaque internal value and should be ignored.

- 0: pending
- 16: running
- 32: shutting-down
- 48: terminated
- 64: stopping
- 80: stopped
- **Name** (*string*) –
The current state of the instance.
- **PrivateDnsName** (*string*) –
The private DNS name assigned to the instance. This DNS name can only be used inside the Amazon EC2 network. This name is not available until the instance enters the `running` state.
- **PublicDnsName** (*string*) –
The public DNS name assigned to the instance. This name is not available until the instance enters the `running` state.
- **StateTransitionReason** (*string*) –
The reason for the most recent state transition. This might be an empty string.
- **KeyName** (*string*) –
The name of the key pair, if this instance was launched with an associated key pair.
- **AmiLaunchIndex** (*integer*) –
The AMI launch index, which can be used to find this instance in the launch group.
- **ProductCodes** (*list*) –
The product codes attached to this instance.
- (*dict*) –
Describes a product code.
- **ProductCodeId** (*string*) –
The product code.
- **ProductCodeType** (*string*) –
The type of product code.
- **InstanceType** (*string*) –
The instance type.
- **LaunchTime** (*datetime*) –
The time the instance was launched.
- **Placement** (*dict*) –
The location where the instance launched.
- **AvailabilityZone** (*string*) –
The Availability Zone of the instance.
- **GroupName** (*string*) –
The name of the placement group the instance is in (for cluster compute instances).
- **Tenancy** (*string*) –

The tenancy of the instance (if the instance is running in a VPC). An instance with a tenancy of `dedicated` runs on single-tenant hardware.

- **KernelId** (*string*) –

The kernel associated with this instance.

- **RamdiskId** (*string*) –

The RAM disk associated with this instance.

- **Platform** (*string*) –

The value is `Windows` for Windows instances; otherwise blank.

- **Monitoring** (*dict*) –

The monitoring information for the instance.

- **State** (*string*) –

Indicates whether monitoring is enabled for the instance.

- **SubnetId** (*string*) –

The ID of the subnet in which the instance is running.

- **VpcId** (*string*) –

The ID of the VPC in which the instance is running.

- **PrivateIpAddress** (*string*) –

The private IP address assigned to the instance.

- **PublicIpAddress** (*string*) –

The public IP address assigned to the instance.

- **StateReason** (*dict*) –

The reason for the most recent state transition.

- **Code** (*string*) –

The reason code for the state change.

- **Message** (*string*) –

The message for the state change.

- `Server.SpotInstanceTermination` : A Spot Instance was terminated due to an increase in the market price.

- `Server.InternalError` : An internal error occurred during instance launch, resulting in termination.

- `Server.InsufficientInstanceCapacity` : There was insufficient instance capacity to satisfy the launch request.

- `Client.InternalError` : A client error caused the instance to terminate on launch.

- `Client.InstanceInitiatedShutdown` : The instance was shut down using the `shutdown -h` command from the instance.

- `Client.UserInitiatedShutdown` : The instance was shut down using the Amazon EC2 API.

- `Client.VolumeLimitExceeded` : The volume limit was exceeded.

- `Client.InvalidSnapshot.NotFound` : The specified snapshot was not found.

- **Architecture** (*string*) –

The architecture of the image.

- **RootDeviceType** (*string*) –

The root device type used by the AMI. The AMI can use an EBS volume or an instance store volume.

- **RootDeviceName** (*string*) –

The root device name (for example, `/dev/sda1` or `/dev/xvda`).

- **BlockDeviceMappings** (*list*) –

Any block device mapping entries for the instance.

- (*dict*) –

Describes a block device mapping.

- **DeviceName** (*string*) –

The device name exposed to the instance (for example, `/dev/sdh` or `xvdh`).

- **Ebs** (*dict*) –

Parameters used to automatically set up EBS volumes when the instance is launched.

- **VolumeId** (*string*) –

The ID of the EBS volume.

- **Status** (*string*) –

The attachment state.

- **AttachTime** (*datetime*) –

The time stamp when the attachment initiated.

- **DeleteOnTermination** (*boolean*) –

Indicates whether the volume is deleted on instance termination.

- **VirtualizationType** (*string*) –

The virtualization type of the instance.

- **InstanceLifecycle** (*string*) –

Indicates whether this is a Spot Instance.

- **SpotInstanceRequestId** (*string*) –

The ID of the Spot Instance request.

- **ClientToken** (*string*) –

The idempotency token you provided when you launched the instance.

- **Tags** (*list*) –

Any tags assigned to the instance.

- (*dict*) –

Describes a tag.

- **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

- **Value** (*string*) –
The value of the tag.
Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.
- **SecurityGroups** (*list*) –
One or more security groups for the instance.
- (*dict*) –
Describes a security group.
- **GroupName** (*string*) –
The name of the security group.
- **GroupId** (*string*) –
The ID of the security group.
- **SourceDestCheck** (*boolean*) –
Specifies whether to enable an instance launched in a VPC to perform NAT. This controls whether source/destination checking is enabled on the instance. A value of `true` means checking is enabled, and `false` means checking is disabled. The value must be `false` for the instance to perform NAT. For more information, see [NAT Instances](#) in the *Amazon Virtual Private Cloud User Guide*.
- **Hypervisor** (*string*) –
The hypervisor type of the instance.
- **NetworkInterfaces** (*list*) –
[EC2-VPC] One or more network interfaces for the instance.
- (*dict*) –
Describes a network interface.
- **NetworkInterfaceId** (*string*) –
The ID of the network interface.
- **SubnetId** (*string*) –
The ID of the subnet.
- **VpcId** (*string*) –
The ID of the VPC.
- **Description** (*string*) –
The description.
- **OwnerId** (*string*) –
The ID of the AWS account that created the network interface.
- **Status** (*string*) –
The status of the network interface.
- **MacAddress** (*string*) –
The MAC address.
- **PrivateIpAddress** (*string*) –
The IP address of the network interface within the subnet.
- **PrivateDnsName** (*string*) –
The private DNS name.

- **SourceDestCheck** (*boolean*) –
Indicates whether to validate network traffic to or from this network interface.
- **Groups** (*list*) –
One or more security groups.
- (*dict*) –
Describes a security group.
- **GroupName** (*string*) –
The name of the security group.
- **GroupId** (*string*) –
The ID of the security group.
- **Attachment** (*dict*) –
The network interface attachment.
- **AttachmentId** (*string*) –
The ID of the network interface attachment.
- **DeviceIndex** (*integer*) –
The index of the device on the instance for the network interface attachment.
- **Status** (*string*) –
The attachment state.
- **AttachTime** (*datetime*) –
The time stamp when the attachment initiated.
- **DeleteOnTermination** (*boolean*) –
Indicates whether the network interface is deleted when the instance is terminated.
- **Association** (*dict*) –
The association information for an Elastic IP associated with the network interface.
- **PublicIp** (*string*) –
The public IP address or Elastic IP address bound to the network interface.
- **PublicDnsName** (*string*) –
The public DNS name.
- **IpOwnerId** (*string*) –
The ID of the owner of the Elastic IP address.
- **PrivateIpAddresses** (*list*) –
The private IP addresses associated with the network interface.
- (*dict*) –
Describes a private IP address.
- **PrivateIpAddress** (*string*) –
The private IP address of the network interface.
- **PrivateDnsName** (*string*) –
The private DNS name.

- **Primary** (*boolean*) –
Indicates whether this IP address is the primary private IP address of the network interface.
- **Association** (*dict*) –
The association information for an Elastic IP address for the network interface.
- **PublicIp** (*string*) –
The public IP address or Elastic IP address bound to the network interface.
- **PublicDnsName** (*string*) –
The public DNS name.
- **IpOwnerId** (*string*) –
The ID of the owner of the Elastic IP address.
- **IamInstanceProfile** (*dict*) –
The IAM instance profile associated with the instance.
- **Arn** (*string*) –
The Amazon Resource Name (ARN) of the instance profile.
- **Id** (*string*) –
The ID of the instance profile.
- **EbsOptimized** (*boolean*) –
Indicates whether the instance is optimized for EBS I/O. This optimization provides dedicated throughput to Amazon EBS and an optimized configuration stack to provide optimal I/O performance. This optimization isn't available with all instance types. Additional usage charges apply when using an EBS Optimized instance.
- **SriovNetSupport** (*string*) –
Specifies whether enhanced networking is enabled.

start_instances (***kwargs*)

Starts an Amazon EBS-backed AMI that you've previously stopped.

Instances that use Amazon EBS volumes as their root devices can be quickly stopped and started. When an instance is stopped, the compute resources are released and you are not billed for hourly instance usage. However, your root partition Amazon EBS volume remains, continues to persist your data, and you are charged for Amazon EBS volume usage. You can restart your instance at any time. Each time you transition an instance from stopped to started, Amazon EC2 charges a full instance hour, even if transitions happen multiple times within a single hour.

Before stopping an instance, make sure it is in a state from which it can be restarted. Stopping an instance does not preserve data stored in RAM.

Performing this operation on an instance that uses an instance store as its root device returns an error.

For more information, see [Stopping Instances](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.start_instances(
    InstanceIds=[
        'string',
    ],
```

```
AdditionalInfo='string',  
DryRun=True|False  
)
```

Parameters

- **InstanceIds** (*list*) – [REQUIRED]

One or more instance IDs.

– (*string*) –

- **AdditionalInfo** (*string*) – Reserved.
- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns**Response Syntax**

```
{  
  'StartingInstances': [  
    {  
      'InstanceId': 'string',  
      'CurrentState': {  
        'Code': 123,  
        'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stopping'  
      },  
      'PreviousState': {  
        'Code': 123,  
        'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stopping'  
      }  
    },  
  ],  
}
```

Response Structure

- (*dict*) –
 - **StartingInstances** (*list*) –

Information about one or more started instances.

* (*dict*) –

Describes an instance state change.

- **InstanceId** (*string*) –

The ID of the instance.

- **CurrentState** (*dict*) –

The current state of the instance.

- **Code** (*integer*) –

The low byte represents the state. The high byte is an opaque internal value and should be ignored.

- 0: pending
- 16: running
- 32: shutting-down
- 48: terminated
- 64: stopping
- 80: stopped

- **Name** (*string*) –
The current state of the instance.
- **PreviousState** (*dict*) –
The previous state of the instance.
- **Code** (*integer*) –
The low byte represents the state. The high byte is an opaque internal value and should be ignored.
- 0 : pending
- 16 : running
- 32 : shutting-down
- 48 : terminated
- 64 : stopping
- 80 : stopped
- **Name** (*string*) –
The current state of the instance.

stop_instances (***kwargs*)

Stops an Amazon EBS-backed instance. Each time you transition an instance from stopped to started, Amazon EC2 charges a full instance hour, even if transitions happen multiple times within a single hour.

You can't start or stop Spot Instances.

Instances that use Amazon EBS volumes as their root devices can be quickly stopped and started. When an instance is stopped, the compute resources are released and you are not billed for hourly instance usage. However, your root partition Amazon EBS volume remains, continues to persist your data, and you are charged for Amazon EBS volume usage. You can restart your instance at any time.

Before stopping an instance, make sure it is in a state from which it can be restarted. Stopping an instance does not preserve data stored in RAM.

Performing this operation on an instance that uses an instance store as its root device returns an error.

You can stop, start, and terminate EBS-backed instances. You can only terminate instance store-backed instances. What happens to an instance differs if you stop it or terminate it. For example, when you stop an instance, the root device and any other devices attached to the instance persist. When you terminate an instance, the root device and any other devices attached during the instance launch are automatically deleted. For more information about the differences between stopping and terminating instances, see [Instance Lifecycle](#) in the *Amazon Elastic Compute Cloud User Guide*.

For more information about troubleshooting, see [Troubleshooting Stopping Your Instance](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.stop_instances(
    DryRun=True|False,
    InstanceIds=[
        'string',
    ],
    Force=True|False
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **InstanceIds** (*list*) – [REQUIRED]

One or more instance IDs.

– (*string*) –

- **Force** (*boolean*) – Forces the instances to stop. The instances do not have an opportunity to flush file system caches or file system metadata. If you use this option, you must perform file system check and repair procedures. This option is not recommended for Windows instances.

Default: false

Return type dict

Returns

Response Syntax

```
{
  'StoppingInstances': [
    {
      'InstanceId': 'string',
      'CurrentState': {
        'Code': 123,
        'Name': 'pending' | 'running' | 'shutting-down' | 'terminated' | 'stopping',
      },
      'PreviousState': {
        'Code': 123,
        'Name': 'pending' | 'running' | 'shutting-down' | 'terminated' | 'stopping',
      },
    },
  ],
}
```

Response Structure

- (*dict*) –

– **StoppingInstances** (*list*) –

Information about one or more stopped instances.

* (*dict*) –

Describes an instance state change.

· **InstanceId** (*string*) –

The ID of the instance.

· **CurrentState** (*dict*) –

The current state of the instance.

· **Code** (*integer*) –

The low byte represents the state. The high byte is an opaque internal value and should be ignored.

· 0 : pending

· 16 : running

· 32 : shutting-down

· 48 : terminated

· 64 : stopping

· 80 : stopped

· **Name** (*string*) –

The current state of the instance.

· **PreviousState** (*dict*) –

The previous state of the instance.

- **Code** (*integer*) –

The low byte represents the state. The high byte is an opaque internal value and should be ignored.

- 0 : pending
- 16 : running
- 32 : shutting-down
- 48 : terminated
- 64 : stopping
- 80 : stopped
- **Name** (*string*) –

The current state of the instance.

terminate_instances (***kwargs*)

Shuts down one or more instances. This operation is idempotent; if you terminate an instance more than once, each call succeeds.

Terminated instances remain visible after termination (for approximately one hour).

By default, Amazon EC2 deletes all EBS volumes that were attached when the instance launched. Volumes attached after instance launch continue running.

You can stop, start, and terminate EBS-backed instances. You can only terminate instance store-backed instances. What happens to an instance differs if you stop it or terminate it. For example, when you stop an instance, the root device and any other devices attached to the instance persist. When you terminate an instance, any attached EBS volumes with the `DeleteOnTermination` block device mapping parameter set to `true` are automatically deleted. For more information about the differences between stopping and terminating instances, see [Instance Lifecycle](#) in the *Amazon Elastic Compute Cloud User Guide*.

For more information about troubleshooting, see [Troubleshooting Terminating Your Instance](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.terminate_instances(
    DryRun=True|False,
    InstanceIds=[
        'string',
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceIds** (*list*) – **[REQUIRED]**

One or more instance IDs.

– (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'TerminatingInstances': [
```

```
{
    'InstanceId': 'string',
    'CurrentState': {
        'Code': 123,
        'Name': 'pending' | 'running' | 'shutting-down' | 'terminated' | 'stopping',
    },
    'PreviousState': {
        'Code': 123,
        'Name': 'pending' | 'running' | 'shutting-down' | 'terminated' | 'stopping',
    },
},
]
```

Response Structure

- (*dict*) –
 - **TerminatingInstances** (*list*) –

Information about one or more terminated instances.

- * (*dict*) –

Describes an instance state change.

- **InstanceId** (*string*) –
The ID of the instance.
- **CurrentState** (*dict*) –
The current state of the instance.
- **Code** (*integer*) –
The low byte represents the state. The high byte is an opaque internal value and should be ignored.
 - 0: pending
 - 16: running
 - 32: shutting-down
 - 48: terminated
 - 64: stopping
 - 80: stopped
- **Name** (*string*) –
The current state of the instance.
- **PreviousState** (*dict*) –
The previous state of the instance.
- **Code** (*integer*) –
The low byte represents the state. The high byte is an opaque internal value and should be ignored.
 - 0: pending
 - 16: running
 - 32: shutting-down
 - 48: terminated
 - 64: stopping
 - 80: stopped
- **Name** (*string*) –
The current state of the instance.

unassign_private_ip_addresses (***kwargs*)

Unassigns one or more secondary private IP addresses from a network interface.

Request Syntax

```
response = client.unassign_private_ip_addresses(
    NetworkInterfaceId='string',
    PrivateIpAddresses=[
        'string',
    ]
)
```

Parameters

- **NetworkInterfaceId** (*string*) – [REQUIRED]

The ID of the network interface.

- **PrivateIpAddresses** (*list*) – [REQUIRED]

The secondary private IP addresses to unassign from the network interface. You can specify this option multiple times to unassign more than one IP address.

– (*string*) –

Returns None

unmonitor_instances (***kwargs*)

Disables monitoring for a running instance. For more information about monitoring instances, see [Monitoring Your Instances and Volumes](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = client.unmonitor_instances(
    DryRun=True|False,
    InstanceIds=[
        'string',
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceIds** (*list*) – [REQUIRED]

One or more instance IDs.

– (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'InstanceMonitorings': [
        {
            'InstanceId': 'string',
            'Monitoring': {
                'State': 'disabled'|'disabling'|'enabled'|'pending'
            }
        },
    ]
}
```

Response Structure

- *(dict)* –
 - **InstanceMonitorings** (*list*) –
Monitoring information for one or more instances.
 - * *(dict)* –
Describes the monitoring information of the instance.
 - **InstanceId** (*string*) –
The ID of the instance.
 - **Monitoring** (*dict*) –
The monitoring information.
 - **State** (*string*) –
Indicates whether monitoring is enabled for the instance.

Paginators

The available paginators are:

- `EC2.Paginator.DescribeInstanceStatus`
- `EC2.Paginator.DescribeInstances`
- `EC2.Paginator.DescribeReservedInstancesModifications`
- `EC2.Paginator.DescribeReservedInstancesOfferings`
- `EC2.Paginator.DescribeSnapshots`
- `EC2.Paginator.DescribeSpotPriceHistory`
- `EC2.Paginator.DescribeTags`
- `EC2.Paginator.DescribeVolumeStatus`

class `EC2.Paginator.DescribeInstanceStatus`

```
paginator = client.get_paginator('describe_instance_status')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `EC2.Client.describe_instance_status()`.

Request Syntax

```
response_iterator = paginator.paginate(
    DryRun=True|False,
    InstanceIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
```

```

IncludeAllInstances=True|False,
PaginationConfig={
    'MaxItems': 123,
    'PageSize': 123,
    'StartingToken': 'string'
}
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceIds** (*list*) – One or more instance IDs.

Default: Describes all your instances.

Constraints: Maximum 100 explicitly specified instance IDs.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `availability-zone` - The Availability Zone of the instance.
 - `event.code` - The code for the scheduled event (`instance-reboot` | `system-reboot` | `system-maintenance` | `instance-retirement` | `instance-stop`).
 - `event.description` - A description of the event.
 - `event.not-after` - The latest end time for the scheduled event (for example, `2014-09-15T17:15:20.000Z`).
 - `event.not-before` - The earliest start time for the scheduled event (for example, `2014-09-15T17:15:20.000Z`).
 - `instance-state-code` - The code for the instance state, as a 16-bit unsigned integer. The high byte is an opaque internal value and should be ignored. The low byte is set based on the state represented. The valid values are 0 (pending), 16 (running), 32 (shutting-down), 48 (terminated), 64 (stopping), and 80 (stopped).
 - `instance-state-name` - The state of the instance (`pending` | `running` | `shutting-down` | `terminated` | `stopping` | `stopped`).
 - `instance-status.reachability` - Filters on instance status where the name is `reachability` (`passed` | `failed` | `initializing` | `insufficient-data`).
 - `instance-status.status` - The status of the instance (`ok` | `impaired` | `initializing` | `insufficient-data` | `not-applicable`).
 - `system-status.reachability` - Filters on system status where the name is `reachability` (`passed` | `failed` | `initializing` | `insufficient-data`).
 - `system-status.status` - The system status of the instance (`ok` | `impaired` | `initializing` | `insufficient-data` | `not-applicable`).
 - (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

- **IncludeAllInstances** (*boolean*) – When `true`, includes the health status for all instances. When `false`, includes the health status for running instances only.

Default: `false`

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

– **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

– **PageSize** (*integer*) –

The size of each page.

– **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
  'InstanceStatuses': [
    {
      'InstanceId': 'string',
      'AvailabilityZone': 'string',
      'Events': [
        {
          'Code': 'instance-reboot'|'system-reboot'|'system-maintenance',
          'Description': 'string',
          'NotBefore': datetime(2015, 1, 1),
          'NotAfter': datetime(2015, 1, 1)
        },
      ],
      'InstanceState': {
        'Code': 123,
        'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stopping',
      },
      'SystemStatus': {
        'Status': 'ok'|'impaired'|'insufficient-data'|'not-applicable'|'in-progress',
        'Details': [
          {
            'Name': 'reachability',
            'Status': 'passed'|'failed'|'insufficient-data'|'initializing',
            'ImpairedSince': datetime(2015, 1, 1)
          },
        ],
      },
    },
  ],
  'InstanceStatus': {
    'Status': 'ok'|'impaired'|'insufficient-data'|'not-applicable'|'in-progress',
    'Details': [
      {

```

```

        'Name': 'reachability',
        'Status': 'passed'|'failed'|'insufficient-data'|'initializ
        'ImpairedSince': datetime(2015, 1, 1)
    },
]
}
},
],
}

```

Response Structure

- (*dict*) –
 - **InstanceStatuses** (*list*) –

One or more instance status descriptions.

- * (*dict*) –

Describes the status of an instance.

- **InstanceId** (*string*) –

The ID of the instance.

- **AvailabilityZone** (*string*) –

The Availability Zone of the instance.

- **Events** (*list*) –

Any scheduled events associated with the instance.

- (*dict*) –

Describes a scheduled event for an instance.

- **Code** (*string*) –

The event code.

- **Description** (*string*) –

A description of the event.

After a scheduled event is completed, it can still be described for up to a week. If the event has been completed, this description starts with the following text: [Completed].

- **NotBefore** (*datetime*) –

The earliest scheduled start time for the event.

- **NotAfter** (*datetime*) –

The latest scheduled end time for the event.

- **InstanceState** (*dict*) –

The intended state of the instance. DescribeInstanceStatus requires that an instance be in the `running` state.

- **Code** (*integer*) –

The low byte represents the state. The high byte is an opaque internal value and should be ignored.

- 0 : pending
- 16 : running
- 32 : shutting-down
- 48 : terminated
- 64 : stopping
- 80 : stopped

- **Name** (*string*) –
The current state of the instance.
- **SystemStatus** (*dict*) –
Reports impaired functionality that stems from issues related to the systems that support an instance, such as hardware failures and network connectivity problems.
- **Status** (*string*) –
The status.
- **Details** (*list*) –
The system instance health or application instance health.
- (*dict*) –
Describes the instance status.
- **Name** (*string*) –
The type of instance status.
- **Status** (*string*) –
The status.
- **ImpairedSince** (*datetime*) –
The time when a status check failed. For an instance that was launched and impaired, this is the time when the instance was launched.
- **InstanceStatus** (*dict*) –
Reports impaired functionality that stems from issues internal to the instance, such as impaired reachability.
- **Status** (*string*) –
The status.
- **Details** (*list*) –
The system instance health or application instance health.
- (*dict*) –
Describes the instance status.
- **Name** (*string*) –
The type of instance status.
- **Status** (*string*) –
The status.
- **ImpairedSince** (*datetime*) –
The time when a status check failed. For an instance that was launched and impaired, this is the time when the instance was launched.

class `EC2.Paginator.DescribeInstances`

```
paginator = client.get_paginator('describe_instances')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `EC2.Client.describe_instances()`.

Request Syntax

```
response_iterator = paginator.paginate(
    DryRun=True|False,
    InstanceIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceIds** (*list*) – One or more instance IDs.

Default: Describes all your instances.

– (*string*) –

- **Filters** (*list*) – One or more filters.
 - `architecture` - The instance architecture (i386|x86_64).
 - `availability-zone` - The Availability Zone of the instance.
 - `block-device-mapping.attach-time` - The attach time for an EBS volume mapped to the instance, for example, 2010-09-15T17:15:20.000Z.
 - `block-device-mapping.delete-on-termination` - A Boolean that indicates whether the EBS volume is deleted on instance termination.
 - `block-device-mapping.device-name` - The device name for the EBS volume (for example, /dev/sdh or xvdh).
 - `block-device-mapping.status` - The status for the EBS volume (attaching|attached|detaching|detached).
 - `block-device-mapping.volume-id` - The volume ID of the EBS volume.
 - `client-token` - The idempotency token you provided when you launched the instance.
 - `dns-name` - The public DNS name of the instance.
 - `group-id` - The ID of the security group for the instance. EC2-Classic only.
 - `group-name` - The name of the security group for the instance. EC2-Classic only.
 - `hypervisor` - The hypervisor type of the instance (ovm|xen).
 - `iam-instance-profile.arn` - The instance profile associated with the instance. Specified as an ARN.

- `image-id` - The ID of the image used to launch the instance.
- `instance-id` - The ID of the instance.
- `instance-lifecycle` - Indicates whether this is a Spot Instance (`spot`).
- `instance-state-code` - The state of the instance, as a 16-bit unsigned integer. The high byte is an opaque internal value and should be ignored. The low byte is set based on the state represented. The valid values are: 0 (pending), 16 (running), 32 (shutting-down), 48 (terminated), 64 (stopping), and 80 (stopped).
- `instance-state-name` - The state of the instance (`pending` | `running` | `shutting-down` | `terminated` | `stopping` | `stopped`).
- `instance-type` - The type of instance (for example, `t2.micro`).
- `instance.group-id` - The ID of the security group for the instance.
- `instance.group-name` - The name of the security group for the instance.
- `ip-address` - The public IP address of the instance.
- `kernel-id` - The kernel ID.
- `key-name` - The name of the key pair used when the instance was launched.
- `launch-index` - When launching multiple instances, this is the index for the instance in the launch group (for example, 0, 1, 2, and so on).
- `launch-time` - The time when the instance was launched.
- `monitoring-state` - Indicates whether monitoring is enabled for the instance (`disabled` | `enabled`).
- `owner-id` - The AWS account ID of the instance owner.
- `placement-group-name` - The name of the placement group for the instance.
- `platform` - The platform. Use `windows` if you have Windows instances; otherwise, leave blank.
- `private-dns-name` - The private DNS name of the instance.
- `private-ip-address` - The private IP address of the instance.
- `product-code` - The product code associated with the AMI used to launch the instance.
- `product-code.type` - The type of product code (`devpay` | `marketplace`).
- `ramdisk-id` - The RAM disk ID.
- `reason` - The reason for the current state of the instance (for example, shows “User Initiated [date]” when you stop or terminate the instance). Similar to the `state-reason-code` filter.
- `requester-id` - The ID of the entity that launched the instance on your behalf (for example, AWS Management Console, Auto Scaling, and so on).
- `reservation-id` - The ID of the instance’s reservation. A reservation ID is created any time you launch an instance. A reservation ID has a one-to-one relationship with an instance launch request, but can be associated with more than one instance if you launch multiple instances using the same launch request. For example, if you launch one instance, you’ll get one reservation ID. If you launch ten instances using the same launch request, you’ll also get one reservation ID.
- `root-device-name` - The name of the root device for the instance (for example, `/dev/sda1` or `/dev/xvda`).
- `root-device-type` - The type of root device that the instance uses (`ebs` | `instance-store`).

- `source-dest-check` - Indicates whether the instance performs source/destination checking. A value of `true` means that checking is enabled, and `false` means checking is disabled. The value must be `false` for the instance to perform network address translation (NAT) in your VPC.
- `spot-instance-request-id` - The ID of the Spot Instance request.
- `state-reason-code` - The reason code for the state change.
- `state-reason-message` - A message that describes the state change.
- `subnet-id` - The ID of the subnet for the instance.
- `tag :key =*value*` - The key/value combination of a tag assigned to the resource, where `tag :key` is the tag's key.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter "`tag-key=Purpose`" and the filter "`tag-value=X`", you get any resources assigned both the tag key `Purpose` (regardless of what the tag's value is), and the tag value `X` (regardless of what the tag's key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `tenancy` - The tenancy of an instance (`dedicated`|`default`).
- `virtualization-type` - The virtualization type of the instance (`paravirtual`|`hvm`).
- `vpc-id` - The ID of the VPC that the instance is running in.
- `network-interface.description` - The description of the network interface.
- `network-interface.subnet-id` - The ID of the subnet for the network interface.
- `network-interface.vpc-id` - The ID of the VPC for the network interface.
- `network-interface.network-interface.id` - The ID of the network interface.
- `network-interface.owner-id` - The ID of the owner of the network interface.
- `network-interface.availability-zone` - The Availability Zone for the network interface.
- `network-interface.requester-id` - The requester ID for the network interface.
- `network-interface.requester-managed` - Indicates whether the network interface is being managed by AWS.
- `network-interface.status` - The status of the network interface (`available`)|`in-use`).
- `network-interface.mac-address` - The MAC address of the network interface.
- `network-interface.private-dns-name` - The private DNS name of the network interface.
- `network-interface.source-dest-check` - Whether the network interface performs source/destination checking. A value of `true` means checking is enabled, and `false` means checking is disabled. The value must be `false` for the network interface to perform network address translation (NAT) in your VPC.
- `network-interface.group-id` - The ID of a security group associated with the network interface.
- `network-interface.group-name` - The name of a security group associated with the network interface.

- `network-interface.attachment.attachment-id` - The ID of the interface attachment.
- `network-interface.attachment.instance-id` - The ID of the instance to which the network interface is attached.
- `network-interface.attachment.instance-owner-id` - The owner ID of the instance to which the network interface is attached.
- `network-interface.addresses.private-ip-address` - The private IP address associated with the network interface.
- `network-interface.attachment.device-index` - The device index to which the network interface is attached.
- `network-interface.attachment.status` - The status of the attachment (`attaching` | `attached` | `detaching` | `detached`).
- `network-interface.attachment.attach-time` - The time that the network interface was attached to an instance.
- `network-interface.attachment.delete-on-termination` - Specifies whether the attachment is deleted when an instance is terminated.
- `network-interface.addresses.primary` - Specifies whether the IP address of the network interface is the primary private IP address.
- `network-interface.addresses.association.public-ip` - The ID of the association of an Elastic IP address with a network interface.
- `network-interface.addresses.association.ip-owner-id` - The owner ID of the private IP address associated with the network interface.
- `association.public-ip` - The address of the Elastic IP address bound to the network interface.
- `association.ip-owner-id` - The owner of the Elastic IP address associated with the network interface.
- `association.allocation-id` - The allocation ID returned when you allocated the Elastic IP address for your network interface.
- `association.association-id` - The association ID returned when the network interface was associated with an IP address.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

 - * **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.
 - * **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.
 - (*string*) -
- **PaginationConfig** (*dict*) - A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) -

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) -

The size of each page.
 - **StartingToken** (*string*) -

A token to specify where to start paginating. This is the NextToken from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Reservations': [
    {
      'ReservationId': 'string',
      'OwnerId': 'string',
      'RequesterId': 'string',
      'Groups': [
        {
          'GroupName': 'string',
          'GroupId': 'string'
        },
      ],
      'Instances': [
        {
          'InstanceId': 'string',
          'ImageId': 'string',
          'State': {
            'Code': 123,
            'Name': 'pending'|'running'|'shutting-down'|'terminated'|'
          },
          'PrivateDnsName': 'string',
          'PublicDnsName': 'string',
          'StateTransitionReason': 'string',
          'KeyName': 'string',
          'AmiLaunchIndex': 123,
          'ProductCodes': [
            {
              'ProductCodeId': 'string',
              'ProductCodeType': 'devpay'|'marketplace'
            },
          ],
          'InstanceType': 't1.micro'|'m1.small'|'m1.medium'|'m1.large'|'
          'LaunchTime': datetime(2015, 1, 1),
          'Placement': {
            'AvailabilityZone': 'string',
            'GroupName': 'string',
            'Tenancy': 'default'|'dedicated'
          },
          'KernelId': 'string',
          'RamdiskId': 'string',
          'Platform': 'Windows',
          'Monitoring': {
            'State': 'disabled'|'disabling'|'enabled'|'pending'
          },
          'SubnetId': 'string',
          'VpcId': 'string',
          'PrivateIpAddress': 'string',
          'PublicIpAddress': 'string',
          'StateReason': {
            'Code': 'string',
            'Message': 'string'
          }
        },
      ],
    },
  ],
}
```

```
},
'Architecture': 'i386'|'x86_64',
'RootDeviceType': 'ebs'|'instance-store',
'RootDeviceName': 'string',
'BlockDeviceMappings': [
    {
        'DeviceName': 'string',
        'Ebs': {
            'VolumeId': 'string',
            'Status': 'attaching'|'attached'|'detaching'|'detached',
            'AttachTime': datetime(2015, 1, 1),
            'DeleteOnTermination': True|False
        }
    },
],
'VirtualizationType': 'hvm'|'paravirtual',
'InstanceLifecycle': 'spot',
'SpotInstanceRequestId': 'string',
'ClientToken': 'string',
'Tags': [
    {
        'Key': 'string',
        'Value': 'string'
    },
],
'SecurityGroups': [
    {
        'GroupName': 'string',
        'GroupId': 'string'
    },
],
'SourceDestCheck': True|False,
'Hypervisor': 'ovm'|'xen',
'NetworkInterfaces': [
    {
        'NetworkInterfaceId': 'string',
        'SubnetId': 'string',
        'VpcId': 'string',
        'Description': 'string',
        'OwnerId': 'string',
        'Status': 'available'|'attaching'|'in-use'|'detaching',
        'MacAddress': 'string',
        'PrivateIpAddress': 'string',
        'PrivateDnsName': 'string',
        'SourceDestCheck': True|False,
        'Groups': [
            {
                'GroupName': 'string',
                'GroupId': 'string'
            },
        ],
    },
],
'Attachment': {
    'AttachmentId': 'string',
    'DeviceIndex': 123,
    'Status': 'attaching'|'attached'|'detaching'|'detached',
    'AttachTime': datetime(2015, 1, 1),
    'DeleteOnTermination': True|False
},
```

```

        'Association': {
            'PublicIp': 'string',
            'PublicDnsName': 'string',
            'IpOwnerId': 'string'
        },
        'PrivateIpAddresses': [
            {
                'PrivateIpAddress': 'string',
                'PrivateDnsName': 'string',
                'Primary': True|False,
                'Association': {
                    'PublicIp': 'string',
                    'PublicDnsName': 'string',
                    'IpOwnerId': 'string'
                }
            },
            ...
        ],
        'IamInstanceProfile': {
            'Arn': 'string',
            'Id': 'string'
        },
        'EbsOptimized': True|False,
        'SriovNetSupport': 'string'
    },
    ...
],
}

```

Response Structure

- *(dict)* –
 - **Reservations** (*list*) –
 - One or more reservations.
 - * *(dict)* –
 - Describes a reservation.
 - **ReservationId** (*string*) –
 - The ID of the reservation.
 - **OwnerId** (*string*) –
 - The ID of the AWS account that owns the reservation.
 - **RequesterId** (*string*) –
 - The ID of the requester that launched the instances on your behalf (for example, AWS Management Console or Auto Scaling).
 - **Groups** (*list*) –
 - One or more security groups.
 - *(dict)* –
 - Describes a security group.
 - **GroupName** (*string*) –
 - The name of the security group.

- **GroupId** (*string*) –
The ID of the security group.
- **Instances** (*list*) –
One or more instances.
- (*dict*) –
Describes an instance.
- **InstanceId** (*string*) –
The ID of the instance.
- **ImageId** (*string*) –
The ID of the AMI used to launch the instance.
- **State** (*dict*) –
The current state of the instance.
- **Code** (*integer*) –
The low byte represents the state. The high byte is an opaque internal value and should be ignored.
 - 0 : pending
 - 16 : running
 - 32 : shutting-down
 - 48 : terminated
 - 64 : stopping
 - 80 : stopped
- **Name** (*string*) –
The current state of the instance.
- **PrivateDnsName** (*string*) –
The private DNS name assigned to the instance. This DNS name can only be used inside the Amazon EC2 network. This name is not available until the instance enters the `running` state.
- **PublicDnsName** (*string*) –
The public DNS name assigned to the instance. This name is not available until the instance enters the `running` state.
- **StateTransitionReason** (*string*) –
The reason for the most recent state transition. This might be an empty string.
- **KeyName** (*string*) –
The name of the key pair, if this instance was launched with an associated key pair.
- **AmiLaunchIndex** (*integer*) –
The AMI launch index, which can be used to find this instance in the launch group.
- **ProductCodes** (*list*) –
The product codes attached to this instance.
- (*dict*) –
Describes a product code.
- **ProductCodeId** (*string*) –
The product code.

- **ProductCodeType** (*string*) –
The type of product code.
- **InstanceType** (*string*) –
The instance type.
- **LaunchTime** (*datetime*) –
The time the instance was launched.
- **Placement** (*dict*) –
The location where the instance launched.
- **AvailabilityZone** (*string*) –
The Availability Zone of the instance.
- **GroupName** (*string*) –
The name of the placement group the instance is in (for cluster compute instances).
- **Tenancy** (*string*) –
The tenancy of the instance (if the instance is running in a VPC). An instance with a tenancy of `dedicated` runs on single-tenant hardware.
- **KernelId** (*string*) –
The kernel associated with this instance.
- **RamdiskId** (*string*) –
The RAM disk associated with this instance.
- **Platform** (*string*) –
The value is `Windows` for Windows instances; otherwise blank.
- **Monitoring** (*dict*) –
The monitoring information for the instance.
- **State** (*string*) –
Indicates whether monitoring is enabled for the instance.
- **SubnetId** (*string*) –
The ID of the subnet in which the instance is running.
- **VpcId** (*string*) –
The ID of the VPC in which the instance is running.
- **PrivateIpAddress** (*string*) –
The private IP address assigned to the instance.
- **PublicIpAddress** (*string*) –
The public IP address assigned to the instance.
- **StateReason** (*dict*) –
The reason for the most recent state transition.
- **Code** (*string*) –
The reason code for the state change.
- **Message** (*string*) –
The message for the state change.
- `Server.SpotInstanceTermination` : A Spot Instance was terminated due to an increase in the market price.

- `Server.InternalError` : An internal error occurred during instance launch, resulting in termination.
- `Server.InsufficientInstanceCapacity` : There was insufficient instance capacity to satisfy the launch request.
- `Client.InternalError` : A client error caused the instance to terminate on launch.
- `Client.InstanceInitiatedShutdown` : The instance was shut down using the shutdown `-h` command from the instance.
- `Client.UserInitiatedShutdown` : The instance was shut down using the Amazon EC2 API.
- `Client.VolumeLimitExceeded` : The volume limit was exceeded.
- `Client.InvalidSnapshot.NotFound` : The specified snapshot was not found.
- **Architecture** (*string*) –
The architecture of the image.
- **RootDeviceType** (*string*) –
The root device type used by the AMI. The AMI can use an EBS volume or an instance store volume.
- **RootDeviceName** (*string*) –
The root device name (for example, `/dev/sda1` or `/dev/xvda`).
- **BlockDeviceMappings** (*list*) –
Any block device mapping entries for the instance.
(*dict*) –
Describes a block device mapping.
- **DeviceName** (*string*) –
The device name exposed to the instance (for example, `/dev/sdh` or `xvdh`).
- **Ebs** (*dict*) –
Parameters used to automatically set up EBS volumes when the instance is launched.
- **VolumeId** (*string*) –
The ID of the EBS volume.
- **Status** (*string*) –
The attachment state.
- **AttachTime** (*datetime*) –
The time stamp when the attachment initiated.
- **DeleteOnTermination** (*boolean*) –
Indicates whether the volume is deleted on instance termination.
- **VirtualizationType** (*string*) –
The virtualization type of the instance.
- **InstanceLifecycle** (*string*) –
Indicates whether this is a Spot Instance.

- **SpotInstanceRequestId** (*string*) –
The ID of the Spot Instance request.
- **ClientToken** (*string*) –
The idempotency token you provided when you launched the instance.
- **Tags** (*list*) –
Any tags assigned to the instance.
- (*dict*) –
Describes a tag.
- **Key** (*string*) –
The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`
- **Value** (*string*) –
The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.
- **SecurityGroups** (*list*) –
One or more security groups for the instance.
- (*dict*) –
Describes a security group.
- **GroupName** (*string*) –
The name of the security group.
- **GroupId** (*string*) –
The ID of the security group.
- **SourceDestCheck** (*boolean*) –
Specifies whether to enable an instance launched in a VPC to perform NAT. This controls whether source/destination checking is enabled on the instance. A value of `true` means checking is enabled, and `false` means checking is disabled. The value must be `false` for the instance to perform NAT. For more information, see [NAT Instances](#) in the *Amazon Virtual Private Cloud User Guide*.
- **Hypervisor** (*string*) –
The hypervisor type of the instance.
- **NetworkInterfaces** (*list*) –
[EC2-VPC] One or more network interfaces for the instance.
- (*dict*) –
Describes a network interface.
- **NetworkInterfaceId** (*string*) –
The ID of the network interface.
- **SubnetId** (*string*) –
The ID of the subnet.

- **VpcId** (*string*) –
The ID of the VPC.
- **Description** (*string*) –
The description.
- **OwnerId** (*string*) –
The ID of the AWS account that created the network interface.
- **Status** (*string*) –
The status of the network interface.
- **MacAddress** (*string*) –
The MAC address.
- **PrivateIpAddress** (*string*) –
The IP address of the network interface within the subnet.
- **PrivateDnsName** (*string*) –
The private DNS name.
- **SourceDestCheck** (*boolean*) –
Indicates whether to validate network traffic to or from this network interface.
- **Groups** (*list*) –
One or more security groups.
- (*dict*) –
Describes a security group.
- **GroupName** (*string*) –
The name of the security group.
- **GroupId** (*string*) –
The ID of the security group.
- **Attachment** (*dict*) –
The network interface attachment.
- **AttachmentId** (*string*) –
The ID of the network interface attachment.
- **DeviceIndex** (*integer*) –
The index of the device on the instance for the network interface attachment.
- **Status** (*string*) –
The attachment state.
- **AttachTime** (*datetime*) –
The time stamp when the attachment initiated.
- **DeleteOnTermination** (*boolean*) –
Indicates whether the network interface is deleted when the instance is terminated.
- **Association** (*dict*) –
The association information for an Elastic IP associated with the network interface.
- **PublicIp** (*string*) –

The public IP address or Elastic IP address bound to the network interface.

- **PublicDnsName** (*string*) –

The public DNS name.

- **IpOwnerId** (*string*) –

The ID of the owner of the Elastic IP address.

- **PrivateIpAddresses** (*list*) –

The private IP addresses associated with the network interface.

- (*dict*) –

Describes a private IP address.

- **PrivateIpAddress** (*string*) –

The private IP address of the network interface.

- **PrivateDnsName** (*string*) –

The private DNS name.

- **Primary** (*boolean*) –

Indicates whether this IP address is the primary private IP address of the network interface.

- **Association** (*dict*) –

The association information for an Elastic IP address for the network interface.

- **PublicIp** (*string*) –

The public IP address or Elastic IP address bound to the network interface.

- **PublicDnsName** (*string*) –

The public DNS name.

- **IpOwnerId** (*string*) –

The ID of the owner of the Elastic IP address.

- **IamInstanceProfile** (*dict*) –

The IAM instance profile associated with the instance.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) of the instance profile.

- **Id** (*string*) –

The ID of the instance profile.

- **EbsOptimized** (*boolean*) –

Indicates whether the instance is optimized for EBS I/O. This optimization provides dedicated throughput to Amazon EBS and an optimized configuration stack to provide optimal I/O performance. This optimization isn't available with all instance types. Additional usage charges apply when using an EBS Optimized instance.

- **SriovNetSupport** (*string*) –

Specifies whether enhanced networking is enabled.

`class EC2.Paginator.DescribeReservedInstancesModifications`

```
paginator = client.get_paginator('describe_reserved_instances_modifications')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `EC2.Client.describe_reserved_instances_modifications()`.

Request Syntax

```
response_iterator = paginator.paginate(
    ReservedInstancesModificationIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **ReservedInstancesModificationIds** (*list*) – IDs for the submitted modification request.
 - (*string*) –
- **Filters** (*list*) – One or more filters.
 - `client-token` - The idempotency token for the modification request.
 - `create-date` - The time when the modification request was created.
 - `effective-date` - The time when the modification becomes effective.
 - `modification-result.reserved-instances-id` - The ID for the Reserved Instances created as part of the modification request. This ID is only available when the status of the modification is fulfilled.
 - `modification-result.target-configuration.availability-zone` - The Availability Zone for the new Reserved Instances.
 - `modification-result.target-configuration.instance-count` - The number of new Reserved Instances.
 - `modification-result.target-configuration.instance-type` - The instance type of the new Reserved Instances.
 - `modification-result.target-configuration.platform` - The network platform of the new Reserved Instances (EC2-Classic | EC2-VPC).
 - `reserved-instances-id` - The ID of the Reserved Instances modified.
 - `reserved-instances-modification-id` - The ID of the modification request.
 - `status` - The status of the Reserved Instances modification request (processing | fulfilled | failed).
 - `status-message` - The reason for the status.

- `update-date` - The time when the modification request was last updated.
- *(dict)* –
A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.
 - * **Name** (*string*) –
The name of the filter. Filter names are case-sensitive.
 - * **Values** (*list*) –
One or more filter values. Filter values are case-sensitive.
 - (*string*) –
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'ReservedInstancesModifications': [
    {
      'ReservedInstancesModificationId': 'string',
      'ReservedInstancesIds': [
        {
          'ReservedInstancesId': 'string'
        },
      ],
      'ModificationResults': [
        {
          'ReservedInstancesId': 'string',
          'TargetConfiguration': {
            'AvailabilityZone': 'string',
            'Platform': 'string',
            'InstanceCount': 123,
            'InstanceType': 't1.micro'|'m1.small'|'m1.medium'|'m1.large'
          }
        },
      ],
      'CreateDate': datetime(2015, 1, 1),
      'UpdateDate': datetime(2015, 1, 1),
      'EffectiveDate': datetime(2015, 1, 1),
      'Status': 'string',
      'StatusMessage': 'string',
      'ClientToken': 'string'
    }
  ]
}
```

```
    },  
    ],  
}
```

Response Structure

- (*dict*) –
 - **ReservedInstancesModifications** (*list*) –

The Reserved Instance modification information.

- * (*dict*) –

Describes a Reserved Instance modification.

- **ReservedInstancesModificationId** (*string*) –

A unique ID for the Reserved Instance modification.

- **ReservedInstancesIds** (*list*) –

The IDs of one or more Reserved Instances.

- (*dict*) –

Describes the ID of a Reserved Instance.

- **ReservedInstancesId** (*string*) –

The ID of the Reserved Instance.

- **ModificationResults** (*list*) –

Contains target configurations along with their corresponding new Reserved Instance IDs.

- (*dict*) –

- **ReservedInstancesId** (*string*) –

The ID for the Reserved Instances that were created as part of the modification request. This field is only available when the modification is fulfilled.

- **TargetConfiguration** (*dict*) –

The target Reserved Instances configurations supplied as part of the modification request.

- **AvailabilityZone** (*string*) –

The Availability Zone for the modified Reserved Instances.

- **Platform** (*string*) –

The network platform of the modified Reserved Instances, which is either EC2-Classic or EC2-VPC.

- **InstanceCount** (*integer*) –

The number of modified Reserved Instances.

- **InstanceType** (*string*) –

The instance type for the modified Reserved Instances.

- **CreateDate** (*datetime*) –

The time when the modification request was created.

- **UpdateDate** (*datetime*) –

The time when the modification request was last updated.

- **EffectiveDate** (*datetime*) –

The time for the modification to become effective.

- **Status** (*string*) –
The status of the Reserved Instances modification request.
- **StatusMessage** (*string*) –
The reason for the status.
- **ClientToken** (*string*) –
A unique, case-sensitive key supplied by the client to ensure that the request is idempotent. For more information, see [Ensuring Idempotency](#).

class `EC2.Paginator.DescribeReservedInstancesOfferings`

```
paginator = client.get_paginator('describe_reserved_instances_offerings')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `EC2.Client.describe_reserved_instances_offerings()`.

Request Syntax

```
response_iterator = paginator.paginate(
    DryRun=True|False,
    ReservedInstancesOfferingIds=[
        'string',
    ],
    InstanceType='t1.micro'|'m1.small'|'m1.medium'|'m1.large'|'m1.xlarge'|'m3.medium'|'m3.l
    AvailabilityZone='string',
    ProductDescription='Linux/UNIX'|'Linux/UNIX (Amazon VPC)'|'Windows'|'Windows (Amazon VP
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    InstanceTenancy='default'|'dedicated',
    OfferingType='Heavy Utilization'|'Medium Utilization'|'Light Utilization'|'No Upfront'
    IncludeMarketplace=True|False,
    MinDuration=123,
    MaxDuration=123,
    MaxInstanceCount=123,
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **ReservedInstancesOfferingIds** (*list*) – One or more Reserved Instances offering IDs.

- *(string)* –
- **InstanceType** (*string*) – The instance type on which the Reserved Instance can be used. For more information, see [Instance Types](#) in the *Amazon Elastic Compute Cloud User Guide*.
- **AvailabilityZone** (*string*) – The Availability Zone in which the Reserved Instance can be used.
- **ProductDescription** (*string*) – The Reserved Instance product platform description. Instances that include (Amazon VPC) in the description are for use with Amazon VPC.
- **Filters** (*list*) – One or more filters.
 - `availability-zone` - The Availability Zone where the Reserved Instance can be used.
 - `duration` - The duration of the Reserved Instance (for example, one year or three years), in seconds (31536000|94608000).
 - `fixed-price` - The purchase price of the Reserved Instance (for example, 9800.0).
 - `instance-type` - The instance type on which the Reserved Instance can be used.
 - `marketplace` - Set to `true` to show only Reserved Instance Marketplace offerings. When this filter is not used, which is the default behavior, all offerings from AWS and Reserved Instance Marketplace are listed.
 - `product-description` - The Reserved Instance product platform description. Instances that include (Amazon VPC) in the product platform description will only be displayed to EC2-Classic account holders and are for use with Amazon VPC. (Linux/UNIX | Linux/UNIX (Amazon VPC) | SUSE Linux | SUSE Linux (Amazon VPC) | Red Hat Enterprise Linux | Red Hat Enterprise Linux (Amazon VPC) | Windows | Windows (Amazon VPC) | Windows with SQL Server Standard | Windows with SQL Server Standard (Amazon VPC) | Windows with SQL Server Web | Windows with SQL Server Web (Amazon VPC) | Windows with SQL Server Enterprise | Windows with SQL Server Enterprise (Amazon VPC))
 - `reserved-instances-offering-id` - The Reserved Instances offering ID.
 - `usage-price` - The usage price of the Reserved Instance, per hour (for example, 0.84).
- *(dict)* –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

- * **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

- * **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

- *(string)* –

- **InstanceTenancy** (*string*) – The tenancy of the Reserved Instance offering. A Reserved Instance with `dedicated` tenancy runs on single-tenant hardware and can only be launched within a VPC.

Default: `default`

- **OfferingType** (*string*) – The Reserved Instance offering type. If you are us-

ing tools that predate the 2011-11-01 API version, you only have access to the Medium Utilization Reserved Instance offering type.

- **IncludeMarketplace** (*boolean*) – Include Marketplace offerings in the response.
- **MinDuration** (*integer*) – The minimum duration (in seconds) to filter when searching for offerings.

Default: 2592000 (1 month)

- **MaxDuration** (*integer*) – The maximum duration (in seconds) to filter when searching for offerings.

Default: 94608000 (3 years)

- **MaxInstanceCount** (*integer*) – The maximum number of instances to filter when searching for offerings.

Default: 20

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict
Returns

Response Syntax

```
{
  'ReservedInstancesOfferings': [
    {
      'ReservedInstancesOfferingId': 'string',
      'InstanceType': 't1.micro'|'m1.small'|'m1.medium'|'m1.large'|'m1.xlarge',
      'AvailabilityZone': 'string',
      'Duration': 123,
      'UsagePrice': ...,
      'FixedPrice': ...,
      'ProductDescription': 'Linux/UNIX'|'Linux/UNIX (Amazon VPC)'|'Windows',
      'InstanceTenancy': 'default'|'dedicated',
      'CurrencyCode': 'USD',
      'OfferingType': 'Heavy Utilization'|'Medium Utilization'|'Light Utilization',
      'RecurringCharges': [
        {
          'Frequency': 'Hourly',
          'Amount': 123.0
        },
      ],
      'Marketplace': True|False,
      'PricingDetails': [
        {
          'Price': 123.0,
          'Count': 123
        },
      ],
    },
  ],
}
```

```
    ],  
    },  
  ],  
}
```

Response Structure

- *(dict)* –

- **ReservedInstancesOfferings** (*list*) –

A list of Reserved Instances offerings.

- * *(dict)* –

Describes a Reserved Instance offering.

- **ReservedInstancesOfferingId** (*string*) –

The ID of the Reserved Instance offering.

- **InstanceType** (*string*) –

The instance type on which the Reserved Instance can be used.

- **AvailabilityZone** (*string*) –

The Availability Zone in which the Reserved Instance can be used.

- **Duration** (*integer*) –

The duration of the Reserved Instance, in seconds.

- **UsagePrice** (*float*) –

The usage price of the Reserved Instance, per hour.

- **FixedPrice** (*float*) –

The purchase price of the Reserved Instance.

- **ProductDescription** (*string*) –

The Reserved Instance product platform description.

- **InstanceTenancy** (*string*) –

The tenancy of the reserved instance.

- **CurrencyCode** (*string*) –

The currency of the Reserved Instance offering you are purchasing. It's specified using ISO 4217 standard currency codes. At this time, the only supported currency is USD .

- **OfferingType** (*string*) –

The Reserved Instance offering type.

- **RecurringCharges** (*list*) –

The recurring charge tag assigned to the resource.

- *(dict)* –

Describes a recurring charge.

- **Frequency** (*string*) –

The frequency of the recurring charge.

- **Amount** (*float*) –

The amount of the recurring charge.

- **Marketplace** (*boolean*) –
Indicates whether the offering is available through the Reserved Instance Marketplace (resale) or AWS. If it's a Reserved Instance Marketplace offering, this is `true`.
- **PricingDetails** (*list*) –
The pricing details of the Reserved Instance offering.
- (*dict*) –
Describes a Reserved Instance offering.
- **Price** (*float*) –
The price per instance.
- **Count** (*integer*) –
The number of instances available for the price.

class `EC2.Paginator.DescribeSnapshots`

```
paginator = client.get_paginator('describe_snapshots')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `EC2.Client.describe_snapshots()`.

Request Syntax

```
response_iterator = paginator.paginate(
    DryRun=True|False,
    SnapshotIds=[
        'string',
    ],
    OwnerIds=[
        'string',
    ],
    RestorableByUserIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **SnapshotIds** (*list*) – One or more snapshot IDs.

Default: Describes snapshots for which you have launch permissions.

– (*string*) –

- **OwnerIds** (*list*) – Returns the snapshots owned by the specified owner. Multiple owners can be specified.

– (*string*) –

- **RestorableByUserIds** (*list*) – One or more AWS accounts IDs that can create volumes from the snapshot.

– (*string*) –

- **Filters** (*list*) – One or more filters.

– **description** - A description of the snapshot.

– **owner-alias** - The AWS account alias (for example, `amazon`) that owns the snapshot.

– **owner-id** - The ID of the AWS account that owns the snapshot.

– **progress** - The progress of the snapshot, as a percentage (for example, 80%).

– **snapshot-id** - The snapshot ID.

– **start-time** - The time stamp when the snapshot was initiated.

– **status** - The status of the snapshot (`pending` | `completed` | `error`).

– **tag :key =*value*** - The key/value combination of a tag assigned to the resource.

– **tag-key** - The key of a tag assigned to the resource. This filter is independent of the **tag-value** filter. For example, if you use both the filter “**tag-key=Purpose**” and the filter “**tag-value=X**”, you get any resources assigned both the tag key Purpose (regardless of what the tag’s value is), and the tag value X (regardless of what the tag’s key is). If you want to list only resources where Purpose is X, see the **tag :key =*value*** filter.

– **tag-value** - The value of a tag assigned to the resource. This filter is independent of the **tag-key** filter.

– **volume-id** - The ID of the volume the snapshot is for.

– **volume-size** - The size of the volume, in GiB.

– (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

– **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

– **PageSize** (*integer*) –

The size of each page.

– **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Snapshots': [
    {
      'SnapshotId': 'string',
      'VolumeId': 'string',
      'State': 'pending'|'completed'|'error',
      'StateMessage': 'string',
      'StartTime': datetime(2015, 1, 1),
      'Progress': 'string',
      'OwnerId': 'string',
      'Description': 'string',
      'VolumeSize': 123,
      'OwnerAlias': 'string',
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        },
      ],
      'Encrypted': True|False,
      'KmsKeyId': 'string',
      'DataEncryptionKeyId': 'string'
    },
  ],
}
```

Response Structure

- (dict) –

- **Snapshots** (list) –

Information about the snapshots.

- * (dict) –

Describes a snapshot.

- **SnapshotId** (string) –

The ID of the snapshot. Each snapshot receives a unique identifier when it is created.

- **VolumeId** (string) –

The ID of the volume that was used to create the snapshot.

- **State** (string) –

The snapshot state.

- **StateMessage** (string) –

Encrypted Amazon EBS snapshots are copied asynchronously. If a snapshot copy operation fails (for example, if the proper AWS Key Management Service (AWS KMS) permissions are not obtained) this field displays error state details to help you diagnose why the error occurred. This parameter is only returned by the `DescribeSnapshots` API operation.

- **StartTime** (*datetime*) –
The time stamp when the snapshot was initiated.
- **Progress** (*string*) –
The progress of the snapshot, as a percentage.
- **OwnerId** (*string*) –
The AWS account ID of the EBS snapshot owner.
- **Description** (*string*) –
The description for the snapshot.
- **VolumeSize** (*integer*) –
The size of the volume, in GiB.
- **OwnerAlias** (*string*) –
The AWS account alias (for example, `amazon`, `self`) or AWS account ID that owns the snapshot.
- **Tags** (*list*) –
Any tags assigned to the snapshot.
- (*dict*) –
Describes a tag.
- **Key** (*string*) –
The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws` :
- **Value** (*string*) –
The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.
- **Encrypted** (*boolean*) –
Indicates whether the snapshot is encrypted.
- **KmsKeyId** (*string*) –
The full ARN of the AWS Key Management Service (AWS KMS) customer master key (CMK) that was used to protect the volume encryption key for the parent volume.
- **DataEncryptionKeyId** (*string*) –
The data encryption key identifier for the snapshot. This value is a unique identifier that corresponds to the data encryption key that was used to encrypt the original volume or snapshot copy. Because data encryption keys are inherited by volumes created from snapshots, and vice versa, if snapshots share the same data encryption key identifier, then they belong to the same volume/snapshot lineage. This parameter is only returned by the DescribeSnapshots API operation.

```
class EC2.Paginator.DescribeSpotPriceHistory
```

```
paginator = client.get_paginator('describe_spot_price_history')
```


paginate (**kwargs)

Creates an iterator that will paginate through responses from `EC2.Client.describe_spot_price_history()`.

Request Syntax

```
response_iterator = paginator.paginate(
    DryRun=True|False,
    StartTime=datetime(2015, 1, 1),
    EndTime=datetime(2015, 1, 1),
    InstanceTypes=[
        't1.micro'|'m1.small'|'m1.medium'|'m1.large'|'m1.xlarge'|'m3.medium'|'m3.large'|'m3
    ],
    ProductDescriptions=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    AvailabilityZone='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **StartTime** (*datetime*) – The date and time, up to the past 90 days, from which to start retrieving the price history data, in UTC format (for example, `YYYY-MM-DD T*HH* :MM :SS Z`).
- **EndTime** (*datetime*) – The date and time, up to the current date, from which to stop retrieving the price history data, in UTC format (for example, `YYYY-MM-DD T*HH* :MM :SS Z`).
- **InstanceTypes** (*list*) – Filters the results by the specified instance types.
 - (*string*) –
- **ProductDescriptions** (*list*) – Filters the results by the specified basic product descriptions.
 - (*string*) –
- **Filters** (*list*) – One or more filters.
 - `availability-zone` - The Availability Zone for which prices should be returned.
 - `instance-type` - The type of instance (for example, `m1.small`).
 - `product-description` - The product description for the Spot price (Linux/UNIX|SUSE Linux|Windows|Linux/UNIX (Amazon VPC)|SUSE Linux (Amazon VPC)|Windows (Amazon VPC)).

- `spot-price` - The Spot price. The value must match exactly (or use wildcards; greater than or less than comparison is not supported).
- `timestamp` - The timestamp of the Spot price history, in UTC format (for example, `YYYY-MM-DD T*HH* :MM :SS Z`). You can use wildcards (`*` and `?`). Greater than or less than comparison is not supported.
- *(dict)* -
A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.
 - * **Name** (*string*) -
The name of the filter. Filter names are case-sensitive.
 - * **Values** (*list*) -
One or more filter values. Filter values are case-sensitive.
 - (*string*) -
- **AvailabilityZone** (*string*) - Filters the results by the specified Availability Zone.
- **PaginationConfig** (*dict*) - A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) -
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) -
The size of each page.
 - **StartingToken** (*string*) -
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'SpotPriceHistory': [
    {
      'InstanceType': 't1.micro'|'m1.small'|'m1.medium'|'m1.large'|'m1.xlarge',
      'ProductDescription': 'Linux/UNIX'|'Linux/UNIX (Amazon VPC)'|'Windows',
      'SpotPrice': 'string',
      'Timestamp': datetime(2015, 1, 1),
      'AvailabilityZone': 'string'
    },
    ...
  ],
}
```

Response Structure

- (*dict*) -
Contains the output of `DescribeSpotPriceHistory`.
 - **SpotPriceHistory** (*list*) -
The historical Spot prices.
 - * (*dict*) -

Describes the maximum hourly price (bid) for any Spot instance launched to fulfill the request.

- **InstanceType** (*string*) –
The instance type.
- **ProductDescription** (*string*) –
A general description of the AMI.
- **SpotPrice** (*string*) –
The maximum price (bid) that you are willing to pay for a Spot instance.
- **Timestamp** (*datetime*) –
The date and time the request was created, in UTC format (for example, *YYYY-MM-DD T*HH* :MM :SS Z*).
- **AvailabilityZone** (*string*) –
The Availability Zone.

class `EC2.Paginator.DescribeTags`

```
paginator = client.get_paginator('describe_tags')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `EC2.Client.describe_tags()`.

Request Syntax

```
response_iterator = paginator.paginate(
    DryRun=True|False,
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Filters** (*list*) – One or more filters.
 - `key` - The tag key.
 - `resource-id` - The resource ID.
 - `resource-type` - The resource type (`customer-gateway` | `dhcp-options` | `image` | `instance` | `internet-gateway` | `network-acl` | `network-interface` | `reserved-instances` | `route-table` | `security-group` | `snapshot` |

```
spot-instances-request | subnet | volume | vpc |
vpn-connection | vpn-gateway ).
```

- **value** - The tag value.

- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

- * **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

- * **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

- (*string*) -

- **PaginationConfig** (*dict*) - A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) -

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) -

The size of each page.

- **StartingToken** (*string*) -

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Tags': [
    {
      'ResourceId': 'string',
      'ResourceType': 'customer-gateway'|'dhcp-options'|'image'|'instance'|'
      'Key': 'string',
      'Value': 'string'
    },
  ],
}
```

Response Structure

- (*dict*) -

- **Tags** (*list*) -

A list of tags.

- * (*dict*) -

Describes a tag.

- **ResourceId** (*string*) -

The ID of the resource. For example, `ami-1a2b3c4d`.

- **ResourceType** (*string*) -

The resource type.

- **Key** (*string*) –
The tag key.
- **Value** (*string*) –
The tag value.

class `EC2.Paginator.DescribeVolumeStatus`

```
paginator = client.get_paginator('describe_volume_status')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `EC2.Client.describe_volume_status()`.

Request Syntax

```
response_iterator = paginator.paginate(
    DryRun=True|False,
    VolumeIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VolumeIds** (*list*) – One or more volume IDs.
Default: Describes all your volumes.
– (*string*) –
- **Filters** (*list*) – One or more filters.
 - `action.code` - The action code for the event (for example, `enable-volume-io`).
 - `action.description` - A description of the action.
 - `action.event-id` - The event ID associated with the action.
 - `availability-zone` - The Availability Zone of the instance.
 - `event.description` - A description of the event.
 - `event.event-id` - The event ID.
 - `event.event-type` - The event type (for `io-enabled`: `passed` | `failed`; for `io-performance`: `io-performance:degraded`).

- `io-performance:severely-degraded` | `io-performance:stalled`).
- `event.not-after` - **The latest end time for the event.**
- `event.not-before` - **The earliest start time for the event.**
- `volume-status.details-name` - **The cause for**
`volume-status.status` (`io-enabled` | `io-performance`
`)`.
- `volume-status.details-status` - **The status of**
`volume-status.details-name` (for `io-enabled` : `passed`
`| failed` ; for `io-performance` : `normal` | `degraded` |
`severely-degraded` | `stalled`).
- `volume-status.status` - **The status of the volume** (`ok` | `impaired`
`| warning` | `insufficient-data`).

– (dict) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

- * **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

- *(string)* –

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

– **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

– **PageSize** (*integer*) –

The size of each page.

– **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'VolumeStatuses': [
    {
      'VolumeId': 'string',
      'AvailabilityZone': 'string',
      'VolumeStatus': {
        'Status': 'ok'|'impaired'|'insufficient-data',
        'Details': [
          {
            'Name': 'io-enabled'|'io-performance',
            'Status': 'string'
          }
        ]
      }
    }
  ],
}
```

```

    'Events': [
        {
            'EventType': 'string',
            'Description': 'string',
            'NotBefore': datetime(2015, 1, 1),
            'NotAfter': datetime(2015, 1, 1),
            'EventId': 'string'
        },
    ],
    'Actions': [
        {
            'Code': 'string',
            'Description': 'string',
            'EventType': 'string',
            'EventId': 'string'
        },
    ],
}

```

Response Structure

- *(dict)* –
 - **VolumeStatuses** (*list*) –
 - A list of volumes.
 - * *(dict)* –
 - Describes the volume status.
 - **VolumeId** (*string*) –
 - The volume ID.
 - **AvailabilityZone** (*string*) –
 - The Availability Zone of the volume.
 - **VolumeStatus** (*dict*) –
 - The volume status.
 - **Status** (*string*) –
 - The status of the volume.
 - **Details** (*list*) –
 - The details of the volume status.
 - *(dict)* –
 - Describes a volume status.
 - **Name** (*string*) –
 - The name of the volume status.
 - **Status** (*string*) –
 - The intended status of the volume status.
 - **Events** (*list*) –
 - A list of events associated with the volume.
 - *(dict)* –
 - Describes a volume status event.
 - **EventType** (*string*) –

- The type of this event.
- **Description** (*string*) –
A description of the event.
- **NotBefore** (*datetime*) –
The earliest start time of the event.
- **NotAfter** (*datetime*) –
The latest end time of the event.
- **EventId** (*string*) –
The ID of this event.
- **Actions** (*list*) –
The details of the operation.
- (*dict*) –
Describes a volume status operation code.
- **Code** (*string*) –
The code identifying the operation, for example, `enable-volume-io`.
- **Description** (*string*) –
A description of the operation.
- **EventType** (*string*) –
The event type associated with this operation.
- **EventId** (*string*) –
The ID of the event associated with this operation.

Waiters

The available waiters are:

- `EC2.Waiter.BundleTaskComplete`
- `EC2.Waiter.ConversionTaskCancelled`
- `EC2.Waiter.ConversionTaskCompleted`
- `EC2.Waiter.ConversionTaskDeleted`
- `EC2.Waiter.CustomerGatewayAvailable`
- `EC2.Waiter.ExportTaskCancelled`
- `EC2.Waiter.ExportTaskCompleted`
- `EC2.Waiter.ImageAvailable`
- `EC2.Waiter.InstanceExists`
- `EC2.Waiter.InstanceRunning`
- `EC2.Waiter.InstanceStatusOk`
- `EC2.Waiter.InstanceStopped`
- `EC2.Waiter.InstanceTerminated`
- `EC2.Waiter.PasswordDataAvailable`

- `EC2.Waiter.SnapshotCompleted`
- `EC2.Waiter.SpotInstanceRequestFulfilled`
- `EC2.Waiter.SubnetAvailable`
- `EC2.Waiter.SystemStatusOk`
- `EC2.Waiter.VolumeAvailable`
- `EC2.Waiter.VolumeDeleted`
- `EC2.Waiter.VolumeInUse`
- `EC2.Waiter.VpcAvailable`
- `EC2.Waiter.VpnConnectionAvailable`
- `EC2.Waiter.VpnConnectionDeleted`

class `EC2.Waiter.BundleTaskComplete`

```
waiter = client.get_waiter('bundle_task_complete')
```

wait (***kwargs*)

Polls `EC2.Client.describe_bundle_tasks()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(
    DryRun=True|False,
    BundleIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **BundleIds** (*list*) – One or more bundle task IDs.
Default: Describes all your bundle tasks.
 - (*string*) –
- **Filters** (*list*) – One or more filters.
 - `bundle-id` - The ID of the bundle task.
 - `error-code` - If the task failed, the error code returned.
 - `error-message` - If the task failed, the error message returned.
 - `instance-id` - The ID of the instance.

- `progress` - The level of task completion, as a percentage (for example, 20%).
- `s3-bucket` - The Amazon S3 bucket to store the AMI.
- `s3-prefix` - The beginning of the AMI name.
- `start-time` - The time the task started (for example, 2013-09-15T17:15:20.000Z).
- `state` - The state of the task (pending | waiting-for-shutdown | bundling | storing | cancelling | complete | failed).
- `update-time` - The time of the most recent update for the task.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

- * **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

- * **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

- (*string*) -

Returns None

class EC2.Waiter.**ConversionTaskCancelled**

```
waiter = client.get_waiter('conversion_task_cancelled')
```

wait (***kwargs*)

Polls `EC2.Client.describe_conversion_tasks()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(  
    DryRun=True|False,  
    Filters=[  
        {  
            'Name': 'string',  
            'Values': [  
                'string',  
            ]  
        },  
    ],  
    ConversionTaskIds=[  
        'string',  
    ]  
)
```

Parameters

- **DryRun** (*boolean*) - Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Filters** (*list*) - One or more filters.
 - *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

• **ConversionTaskIds** (*list*) – One or more conversion task IDs.

– (*string*) –

Returns None

class EC2.Waiter.**ConversionTaskCompleted**

```
waiter = client.get_waiter('conversion_task_completed')
```

wait (***kwargs*)

Polls `EC2.Client.describe_conversion_tasks()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(
    DryRun=True|False,
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    ConversionTaskIds=[
        'string',
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Filters** (*list*) – One or more filters.
 - (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

• **ConversionTaskIds** (*list*) – One or more conversion task IDs.

– (string) –

Returns None

class EC2.Waiter.**ConversionTaskDeleted**

```
waiter = client.get_waiter('conversion_task_deleted')
```

wait (**kwargs)

Polls `EC2.Client.describe_conversion_tasks()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(
    DryRun=True|False,
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    ConversionTaskIds=[
        'string',
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Filters** (*list*) – One or more filters.

– (dict) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

- **ConversionTaskIds** (*list*) – One or more conversion task IDs.

– (*string*) –

Returns None

class EC2.Waiter.**CustomerGatewayAvailable**

```
waiter = client.get_waiter('customer_gateway_available')
```

wait (***kwargs*)

Polls `EC2.Client.describe_customer_gateways()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(
    DryRun=True|False,
    CustomerGatewayIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **CustomerGatewayIds** (*list*) – One or more customer gateway IDs.

Default: Describes all your customer gateways.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `bgp-asn` - The customer gateway's Border Gateway Protocol (BGP) Autonomous System Number (ASN).
 - `customer-gateway-id` - The ID of the customer gateway.
 - `ip-address` - The IP address of the customer gateway's Internet-routable external interface.
 - `state` - The state of the customer gateway (`pending` | `available` | `deleting` | `deleted`).
 - `type` - The type of customer gateway. Currently, the only supported type is `ipsec.1`.
 - `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
 - `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag's value is), and the tag value `X` (regardless of what the tag's key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
 - `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

- * **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

• (*string*) –

Returns None

class EC2.Waiter.**ExportTaskCancelled**

```
waiter = client.get_waiter('export_task_cancelled')
```

wait (***kwargs*)

Polls `EC2.Client.describe_export_tasks()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(
    ExportTaskIds=[
        'string',
    ]
)
```

Parameters **ExportTaskIds** (*list*) – One or more export task IDs.

• (*string*) –

Returns None

class EC2.Waiter.**ExportTaskCompleted**

```
waiter = client.get_waiter('export_task_completed')
```

wait (***kwargs*)

Polls `EC2.Client.describe_export_tasks()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(
    ExportTaskIds=[
        'string',
    ]
)
```

Parameters **ExportTaskIds** (*list*) – One or more export task IDs.

• (*string*) –

Returns None

class EC2.Waiter.**ImageAvailable**

```
waiter = client.get_waiter('image_available')
```

wait (***kwargs*)

Polls `EC2.Client.describe_images()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```

waiter.wait(
    DryRun=True|False,
    ImageIds=[
        'string',
    ],
    Owners=[
        'string',
    ],
    ExecutableUsers=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **ImageIds** (*list*) – One or more image IDs.

Default: Describes all images available to you.

- (*string*) –
- **Owners** (*list*) – Filters the images by the owner. Specify an AWS account ID, `amazon` (owner is Amazon), `aws-marketplace` (owner is AWS Marketplace), `self` (owner is the sender of the request). Omitting this option returns all images for which you have launch permissions, regardless of ownership.
 - (*string*) –
- **ExecutableUsers** (*list*) – Scopes the images by users with explicit launch permissions. Specify an AWS account ID, `self` (the sender of the request), or `all` (public AMIs).
 - (*string*) –
- **Filters** (*list*) – One or more filters.
 - `architecture` - The image architecture (`i386` | `x86_64`).
 - `block-device-mapping.delete-on-termination` - A Boolean value that indicates whether the Amazon EBS volume is deleted on instance termination.
 - `block-device-mapping.device-name` - The device name for the EBS volume (for example, `/dev/sdh`).
 - `block-device-mapping.snapshot-id` - The ID of the snapshot used for the EBS volume.
 - `block-device-mapping.volume-size` - The volume size of the EBS volume, in GiB.
 - `block-device-mapping.volume-type` - The volume type of the EBS volume (`gp2` | `standard` | `io1`).
 - `description` - The description of the image (provided during image creation).
 - `hypervisor` - The hypervisor type (`ovm` | `xen`).

- `image-id` - The ID of the image.
- `image-type` - The image type (`machine` | `kernel` | `ramdisk`).
- `is-public` - A Boolean that indicates whether the image is public.
- `kernel-id` - The kernel ID.
- `manifest-location` - The location of the image manifest.
- `name` - The name of the AMI (provided during image creation).
- `owner-alias` - The AWS account alias (for example, `amazon`).
- `owner-id` - The AWS account ID of the image owner.
- `platform` - The platform. To only list Windows-based AMIs, use `windows`.
- `product-code` - The product code.
- `product-code.type` - The type of the product code (`devpay` | `marketplace`).
- `ramdisk-id` - The RAM disk ID.
- `root-device-name` - The name of the root device volume (for example, `/dev/sda1`).
- `root-device-type` - The type of the root device volume (`ebs` | `instance-store`).
- `state` - The state of the image (`available` | `pending` | `failed`).
- `state-reason-code` - The reason code for the state change.
- `state-reason-message` - The message for the state change.
- `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag’s value is), and the tag value `X` (regardless of what the tag’s key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `virtualization-type` - The virtualization type (`paravirtual` | `hvm`).
- *(dict)* –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

Returns None

class `EC2.Waiter.InstanceExists`

```
waiter = client.get_waiter('instance_exists')
```

wait (***kwargs*)

Polls `EC2.Client.describe_instances()` every 5 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(
    DryRun=True|False,
    InstanceIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceIds** (*list*) – One or more instance IDs.

Default: Describes all your instances.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `architecture` - The instance architecture (`i386` | `x86_64`).
 - `availability-zone` - The Availability Zone of the instance.
 - `block-device-mapping.attach-time` - The attach time for an EBS volume mapped to the instance, for example, `2010-09-15T17:15:20.000Z`.
 - `block-device-mapping.delete-on-termination` - A Boolean that indicates whether the EBS volume is deleted on instance termination.
 - `block-device-mapping.device-name` - The device name for the EBS volume (for example, `/dev/sdh` or `xvdh`).
 - `block-device-mapping.status` - The status for the EBS volume (`attaching` | `attached` | `detaching` | `detached`).
 - `block-device-mapping.volume-id` - The volume ID of the EBS volume.
 - `client-token` - The idempotency token you provided when you launched the instance.
 - `dns-name` - The public DNS name of the instance.
 - `group-id` - The ID of the security group for the instance. EC2-Classic only.
 - `group-name` - The name of the security group for the instance. EC2-Classic only.
 - `hypervisor` - The hypervisor type of the instance (`ovm` | `xen`).
 - `iam-instance-profile.arn` - The instance profile associated with the instance. Specified as an ARN.
 - `image-id` - The ID of the image used to launch the instance.
 - `instance-id` - The ID of the instance.

- `instance-lifecycle` - Indicates whether this is a Spot Instance (`spot`).
- `instance-state-code` - The state of the instance, as a 16-bit unsigned integer. The high byte is an opaque internal value and should be ignored. The low byte is set based on the state represented. The valid values are: 0 (pending), 16 (running), 32 (shutting-down), 48 (terminated), 64 (stopping), and 80 (stopped).
- `instance-state-name` - The state of the instance (`pending` | `running` | `shutting-down` | `terminated` | `stopping` | `stopped`).
- `instance-type` - The type of instance (for example, `t2.micro`).
- `instance.group-id` - The ID of the security group for the instance.
- `instance.group-name` - The name of the security group for the instance.
- `ip-address` - The public IP address of the instance.
- `kernel-id` - The kernel ID.
- `key-name` - The name of the key pair used when the instance was launched.
- `launch-index` - When launching multiple instances, this is the index for the instance in the launch group (for example, 0, 1, 2, and so on).
- `launch-time` - The time when the instance was launched.
- `monitoring-state` - Indicates whether monitoring is enabled for the instance (`disabled` | `enabled`).
- `owner-id` - The AWS account ID of the instance owner.
- `placement-group-name` - The name of the placement group for the instance.
- `platform` - The platform. Use `windows` if you have Windows instances; otherwise, leave blank.
- `private-dns-name` - The private DNS name of the instance.
- `private-ip-address` - The private IP address of the instance.
- `product-code` - The product code associated with the AMI used to launch the instance.
- `product-code.type` - The type of product code (`devpay` | `marketplace`).
- `ramdisk-id` - The RAM disk ID.
- `reason` - The reason for the current state of the instance (for example, shows “User Initiated [date]” when you stop or terminate the instance). Similar to the `state-reason-code` filter.
- `requester-id` - The ID of the entity that launched the instance on your behalf (for example, AWS Management Console, Auto Scaling, and so on).
- `reservation-id` - The ID of the instance’s reservation. A reservation ID is created any time you launch an instance. A reservation ID has a one-to-one relationship with an instance launch request, but can be associated with more than one instance if you launch multiple instances using the same launch request. For example, if you launch one instance, you’ll get one reservation ID. If you launch ten instances using the same launch request, you’ll also get one reservation ID.
- `root-device-name` - The name of the root device for the instance (for example, `/dev/sda1` or `/dev/xvda`).
- `root-device-type` - The type of root device that the instance uses (`ebs` | `instance-store`).
- `source-dest-check` - Indicates whether the instance performs source/destination checking. A value of `true` means that checking is

enabled, and `false` means checking is disabled. The value must be `false` for the instance to perform network address translation (NAT) in your VPC.

- `spot-instance-request-id` - The ID of the Spot Instance request.
- `state-reason-code` - The reason code for the state change.
- `state-reason-message` - A message that describes the state change.
- `subnet-id` - The ID of the subnet for the instance.
- `tag :key =*value*` - The key/value combination of a tag assigned to the resource, where `tag :key` is the tag's key.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter "`tag-key=Purpose`" and the filter "`tag-value=X`", you get any resources assigned both the tag key `Purpose` (regardless of what the tag's value is), and the tag value `X` (regardless of what the tag's key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `tenancy` - The tenancy of an instance (`dedicated|default`).
- `virtualization-type` - The virtualization type of the instance (`paravirtual|hvm`).
- `vpc-id` - The ID of the VPC that the instance is running in.
- `network-interface.description` - The description of the network interface.
- `network-interface.subnet-id` - The ID of the subnet for the network interface.
- `network-interface.vpc-id` - The ID of the VPC for the network interface.
- `network-interface.network-interface.id` - The ID of the network interface.
- `network-interface.owner-id` - The ID of the owner of the network interface.
- `network-interface.availability-zone` - The Availability Zone for the network interface.
- `network-interface.requester-id` - The requester ID for the network interface.
- `network-interface.requester-managed` - Indicates whether the network interface is being managed by AWS.
- `network-interface.status` - The status of the network interface (`available|in-use`).
- `network-interface.mac-address` - The MAC address of the network interface.
- `network-interface.private-dns-name` - The private DNS name of the network interface.
- `network-interface.source-dest-check` - Whether the network interface performs source/destination checking. A value of `true` means checking is enabled, and `false` means checking is disabled. The value must be `false` for the network interface to perform network address translation (NAT) in your VPC.
- `network-interface.group-id` - The ID of a security group associated with the network interface.
- `network-interface.group-name` - The name of a security group associated with the network interface.
- `network-interface.attachment.attachment-id` - The ID of the interface attachment.

- `network-interface.attachment.instance-id` - The ID of the instance to which the network interface is attached.
- `network-interface.attachment.instance-owner-id` - The owner ID of the instance to which the network interface is attached.
- `network-interface.addresses.private-ip-address` - The private IP address associated with the network interface.
- `network-interface.attachment.device-index` - The device index to which the network interface is attached.
- `network-interface.attachment.status` - The status of the attachment (`attaching` | `attached` | `detaching` | `detached`).
- `network-interface.attachment.attach-time` - The time that the network interface was attached to an instance.
- `network-interface.attachment.delete-on-termination` - Specifies whether the attachment is deleted when an instance is terminated.
- `network-interface.addresses.primary` - Specifies whether the IP address of the network interface is the primary private IP address.
- `network-interface.addresses.association.public-ip` - The ID of the association of an Elastic IP address with a network interface.
- `network-interface.addresses.association.ip-owner-id` - The owner ID of the private IP address associated with the network interface.
- `association.public-ip` - The address of the Elastic IP address bound to the network interface.
- `association.ip-owner-id` - The owner of the Elastic IP address associated with the network interface.
- `association.allocation-id` - The allocation ID returned when you allocated the Elastic IP address for your network interface.
- `association.association-id` - The association ID returned when the network interface was associated with an IP address.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

- **NextToken** (*string*) - The token to request the next page of results.
- **MaxResults** (*integer*) - The maximum number of results to return for the request in a single page. The remaining results of the initial request can be seen by sending another request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. You cannot specify this parameter and the `instance IDs` parameter in the same request.

Returns None

`class EC2.Waiter.InstanceRunning`

```
waiter = client.get_waiter('instance_running')
```

wait (***kwargs*)

Polls `EC2.Client.describe_instances()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(
    DryRun=True|False,
    InstanceIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceIds** (*list*) – One or more instance IDs.

Default: Describes all your instances.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `architecture` - The instance architecture (`i386|x86_64`).
 - `availability-zone` - The Availability Zone of the instance.
 - `block-device-mapping.attach-time` - The attach time for an EBS volume mapped to the instance, for example, `2010-09-15T17:15:20.000Z`.
 - `block-device-mapping.delete-on-termination` - A Boolean that indicates whether the EBS volume is deleted on instance termination.
 - `block-device-mapping.device-name` - The device name for the EBS volume (for example, `/dev/sdh` or `xvdh`).
 - `block-device-mapping.status` - The status for the EBS volume (`attaching|attached|detaching|detached`).
 - `block-device-mapping.volume-id` - The volume ID of the EBS volume.
 - `client-token` - The idempotency token you provided when you launched the instance.
 - `dns-name` - The public DNS name of the instance.
 - `group-id` - The ID of the security group for the instance. EC2-Classic only.
 - `group-name` - The name of the security group for the instance. EC2-Classic only.

- `hypervisor` - The hypervisor type of the instance (`ovm` | `xen`).
- `iam-instance-profile.arn` - The instance profile associated with the instance. Specified as an ARN.
- `image-id` - The ID of the image used to launch the instance.
- `instance-id` - The ID of the instance.
- `instance-lifecycle` - Indicates whether this is a Spot Instance (`spot`).
- `instance-state-code` - The state of the instance, as a 16-bit unsigned integer. The high byte is an opaque internal value and should be ignored. The low byte is set based on the state represented. The valid values are: 0 (pending), 16 (running), 32 (shutting-down), 48 (terminated), 64 (stopping), and 80 (stopped).
- `instance-state-name` - The state of the instance (`pending` | `running` | `shutting-down` | `terminated` | `stopping` | `stopped`).
- `instance-type` - The type of instance (for example, `t2.micro`).
- `instance.group-id` - The ID of the security group for the instance.
- `instance.group-name` - The name of the security group for the instance.
- `ip-address` - The public IP address of the instance.
- `kernel-id` - The kernel ID.
- `key-name` - The name of the key pair used when the instance was launched.
- `launch-index` - When launching multiple instances, this is the index for the instance in the launch group (for example, 0, 1, 2, and so on).
- `launch-time` - The time when the instance was launched.
- `monitoring-state` - Indicates whether monitoring is enabled for the instance (`disabled` | `enabled`).
- `owner-id` - The AWS account ID of the instance owner.
- `placement-group-name` - The name of the placement group for the instance.
- `platform` - The platform. Use `windows` if you have Windows instances; otherwise, leave blank.
- `private-dns-name` - The private DNS name of the instance.
- `private-ip-address` - The private IP address of the instance.
- `product-code` - The product code associated with the AMI used to launch the instance.
- `product-code.type` - The type of product code (`devpay` | `marketplace`).
- `ramdisk-id` - The RAM disk ID.
- `reason` - The reason for the current state of the instance (for example, shows “User Initiated [date]” when you stop or terminate the instance). Similar to the `state-reason-code` filter.
- `requester-id` - The ID of the entity that launched the instance on your behalf (for example, AWS Management Console, Auto Scaling, and so on).
- `reservation-id` - The ID of the instance’s reservation. A reservation ID is created any time you launch an instance. A reservation ID has a one-to-one relationship with an instance launch request, but can be associated with more than one instance if you launch multiple instances using the same launch request. For example, if you launch one instance, you’ll get one reservation ID. If you launch ten instances using the same launch request, you’ll also get one reservation ID.
- `root-device-name` - The name of the root device for the instance (for

- example, `/dev/sda1` or `/dev/xvda`).
- `root-device-type` - The type of root device that the instance uses (`ebs` | `instance-store`).
 - `source-dest-check` - Indicates whether the instance performs source/destination checking. A value of `true` means that checking is enabled, and `false` means checking is disabled. The value must be `false` for the instance to perform network address translation (NAT) in your VPC.
 - `spot-instance-request-id` - The ID of the Spot Instance request.
 - `state-reason-code` - The reason code for the state change.
 - `state-reason-message` - A message that describes the state change.
 - `subnet-id` - The ID of the subnet for the instance.
 - `tag :key =*value*` - The key/value combination of a tag assigned to the resource, where `tag :key` is the tag's key.
 - `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter "`tag-key=Purpose`" and the filter "`tag-value=X`", you get any resources assigned both the tag key `Purpose` (regardless of what the tag's value is), and the tag value `X` (regardless of what the tag's key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
 - `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
 - `tenancy` - The tenancy of an instance (`dedicated` | `default`).
 - `virtualization-type` - The virtualization type of the instance (`paravirtual` | `hvm`).
 - `vpc-id` - The ID of the VPC that the instance is running in.
 - `network-interface.description` - The description of the network interface.
 - `network-interface.subnet-id` - The ID of the subnet for the network interface.
 - `network-interface.vpc-id` - The ID of the VPC for the network interface.
 - `network-interface.network-interface.id` - The ID of the network interface.
 - `network-interface.owner-id` - The ID of the owner of the network interface.
 - `network-interface.availability-zone` - The Availability Zone for the network interface.
 - `network-interface.requester-id` - The requester ID for the network interface.
 - `network-interface.requester-managed` - Indicates whether the network interface is being managed by AWS.
 - `network-interface.status` - The status of the network interface (`available` | `in-use`).
 - `network-interface.mac-address` - The MAC address of the network interface.
 - `network-interface.private-dns-name` - The private DNS name of the network interface.
 - `network-interface.source-dest-check` - Whether the network interface performs source/destination checking. A value of `true` means checking is enabled, and `false` means checking is disabled. The value must be `false` for the network interface to perform network address translation (NAT) in your VPC.
 - `network-interface.group-id` - The ID of a security group asso-

ciated with the network interface.

- `network-interface.group-name` - The name of a security group associated with the network interface.
- `network-interface.attachment.attachment-id` - The ID of the interface attachment.
- `network-interface.attachment.instance-id` - The ID of the instance to which the network interface is attached.
- `network-interface.attachment.instance-owner-id` - The owner ID of the instance to which the network interface is attached.
- `network-interface.addresses.private-ip-address` - The private IP address associated with the network interface.
- `network-interface.attachment.device-index` - The device index to which the network interface is attached.
- `network-interface.attachment.status` - The status of the attachment (`attaching` | `attached` | `detaching` | `detached`).
- `network-interface.attachment.attach-time` - The time that the network interface was attached to an instance.
- `network-interface.attachment.delete-on-termination` - Specifies whether the attachment is deleted when an instance is terminated.
- `network-interface.addresses.primary` - Specifies whether the IP address of the network interface is the primary private IP address.
- `network-interface.addresses.association.public-ip` - The ID of the association of an Elastic IP address with a network interface.
- `network-interface.addresses.association.ip-owner-id` - The owner ID of the private IP address associated with the network interface.
- `association.public-ip` - The address of the Elastic IP address bound to the network interface.
- `association.ip-owner-id` - The owner of the Elastic IP address associated with the network interface.
- `association.allocation-id` - The allocation ID returned when you allocated the Elastic IP address for your network interface.
- `association.association-id` - The association ID returned when the network interface was associated with an IP address.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

- **NextToken** (*string*) - The token to request the next page of results.
- **MaxResults** (*integer*) - The maximum number of results to return for the request in a single page. The remaining results of the initial request can be seen by sending another request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. You cannot specify this parameter and the `instance IDs` parameter in the same request.

Returns None

class `EC2.Waiter.InstanceStatusOk`

```
waiter = client.get_waiter('instance_status_ok')
```

wait (***kwargs*)

Polls `EC2.Client.describe_instance_status()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(
    DryRun=True|False,
    InstanceIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    NextToken='string',
    MaxResults=123,
    IncludeAllInstances=True|False
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceIds** (*list*) – One or more instance IDs.

Default: Describes all your instances.

Constraints: Maximum 100 explicitly specified instance IDs.

– (*string*) –

- **Filters** (*list*) – One or more filters.
 - `availability-zone` - The Availability Zone of the instance.
 - `event.code` - The code for the scheduled event (`instance-reboot` | `system-reboot` | `system-maintenance` | `instance-retirement` | `instance-stop`).
 - `event.description` - A description of the event.
 - `event.not-after` - The latest end time for the scheduled event (for example, `2014-09-15T17:15:20.000Z`).
 - `event.not-before` - The earliest start time for the scheduled event (for example, `2014-09-15T17:15:20.000Z`).
 - `instance-state-code` - The code for the instance state, as a 16-bit unsigned integer. The high byte is an opaque internal value and should be ignored. The low byte is set based on the state represented. The valid values are 0 (pending), 16 (running), 32 (shutting-down), 48 (terminated), 64 (stopping), and 80 (stopped).

- `instance-state-name` - The state of the instance (pending | running | shutting-down | terminated | stopping | stopped).
- `instance-status.reachability` - Filters on instance status where the name is reachability (passed | failed | initializing | insufficient-data).
- `instance-status.status` - The status of the instance (ok | impaired | initializing | insufficient-data | not-applicable).
- `system-status.reachability` - Filters on system status where the name is reachability (passed | failed | initializing | insufficient-data).
- `system-status.status` - The system status of the instance (ok | impaired | initializing | insufficient-data | not-applicable).

- (*dict*) -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

- **NextToken** (*string*) - The token to retrieve the next page of results.
- **MaxResults** (*integer*) - The maximum number of results to return for the request in a single page. The remaining results of the initial request can be seen by sending another request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. You cannot specify this parameter and the `instance IDs` parameter in the same request.
- **IncludeAllInstances** (*boolean*) - When `true`, includes the health status for all instances. When `false`, includes the health status for running instances only.

Default: `false`

Returns None

class `EC2.Waiter.InstanceStopped`

```
waiter = client.get_waiter('instance_stopped')
```

wait (***kwargs*)

Polls `EC2.Client.describe_instances()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(  
    DryRun=True|False,  
    InstanceIds=[  
        'string',  
    ],  
    Filters=[
```

```

    {
        'Name': 'string',
        'Values': [
            'string',
        ]
    },
],
NextToken='string',
MaxResults=123
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceIds** (*list*) – One or more instance IDs.

Default: Describes all your instances.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `architecture` - The instance architecture (i386|x86_64).
 - `availability-zone` - The Availability Zone of the instance.
 - `block-device-mapping.attach-time` - The attach time for an EBS volume mapped to the instance, for example, 2010-09-15T17:15:20.000Z.
 - `block-device-mapping.delete-on-termination` - A Boolean that indicates whether the EBS volume is deleted on instance termination.
 - `block-device-mapping.device-name` - The device name for the EBS volume (for example, /dev/sdh or xvdh).
 - `block-device-mapping.status` - The status for the EBS volume (attaching|attached|detaching|detached).
 - `block-device-mapping.volume-id` - The volume ID of the EBS volume.
 - `client-token` - The idempotency token you provided when you launched the instance.
 - `dns-name` - The public DNS name of the instance.
 - `group-id` - The ID of the security group for the instance. EC2-Classic only.
 - `group-name` - The name of the security group for the instance. EC2-Classic only.
 - `hypervisor` - The hypervisor type of the instance (ovm|xen).
 - `iam-instance-profile.arn` - The instance profile associated with the instance. Specified as an ARN.
 - `image-id` - The ID of the image used to launch the instance.
 - `instance-id` - The ID of the instance.
 - `instance-lifecycle` - Indicates whether this is a Spot Instance (spot).
 - `instance-state-code` - The state of the instance, as a 16-bit unsigned integer. The high byte is an opaque internal value and should be ignored. The low byte is set based on the state represented. The valid values are: 0 (pending), 16 (running), 32 (shutting-down), 48 (terminated), 64 (stopping), and 80 (stopped).
 - `instance-state-name` - The state of the instance (pending |

- running|shutting-down|terminated|stopping|stopped
).
- `instance-type` - The type of instance (for example, `t2.micro`).
 - `instance.group-id` - The ID of the security group for the instance.
 - `instance.group-name` - The name of the security group for the instance.
 - `ip-address` - The public IP address of the instance.
 - `kernel-id` - The kernel ID.
 - `key-name` - The name of the key pair used when the instance was launched.
 - `launch-index` - When launching multiple instances, this is the index for the instance in the launch group (for example, 0, 1, 2, and so on).
 - `launch-time` - The time when the instance was launched.
 - `monitoring-state` - Indicates whether monitoring is enabled for the instance (`disabled|enabled`).
 - `owner-id` - The AWS account ID of the instance owner.
 - `placement-group-name` - The name of the placement group for the instance.
 - `platform` - The platform. Use `windows` if you have Windows instances; otherwise, leave blank.
 - `private-dns-name` - The private DNS name of the instance.
 - `private-ip-address` - The private IP address of the instance.
 - `product-code` - The product code associated with the AMI used to launch the instance.
 - `product-code.type` - The type of product code (`devpay | marketplace`).
 - `ramdisk-id` - The RAM disk ID.
 - `reason` - The reason for the current state of the instance (for example, shows “User Initiated [date]” when you stop or terminate the instance). Similar to the `state-reason-code` filter.
 - `requester-id` - The ID of the entity that launched the instance on your behalf (for example, AWS Management Console, Auto Scaling, and so on).
 - `reservation-id` - The ID of the instance’s reservation. A reservation ID is created any time you launch an instance. A reservation ID has a one-to-one relationship with an instance launch request, but can be associated with more than one instance if you launch multiple instances using the same launch request. For example, if you launch one instance, you’ll get one reservation ID. If you launch ten instances using the same launch request, you’ll also get one reservation ID.
 - `root-device-name` - The name of the root device for the instance (for example, `/dev/sda1` or `/dev/xvda`).
 - `root-device-type` - The type of root device that the instance uses (`ebs|instance-store`).
 - `source-dest-check` - Indicates whether the instance performs source/destination checking. A value of `true` means that checking is enabled, and `false` means checking is disabled. The value must be `false` for the instance to perform network address translation (NAT) in your VPC.
 - `spot-instance-request-id` - The ID of the Spot Instance request.
 - `state-reason-code` - The reason code for the state change.
 - `state-reason-message` - A message that describes the state change.
 - `subnet-id` - The ID of the subnet for the instance.
 - `tag :key =*value*` - The key/value combination of a tag assigned to the

resource, where `tag:key` is the tag's key.

- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter "`tag-key=Purpose`" and the filter "`tag-value=X`", you get any resources assigned both the tag key `Purpose` (regardless of what the tag's value is), and the tag value `X` (regardless of what the tag's key is). If you want to list only resources where `Purpose` is `X`, see the `tag:key=*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `tenancy` - The tenancy of an instance (`dedicated|default`).
- `virtualization-type` - The virtualization type of the instance (`paravirtual|hvm`).
- `vpc-id` - The ID of the VPC that the instance is running in.
- `network-interface.description` - The description of the network interface.
- `network-interface.subnet-id` - The ID of the subnet for the network interface.
- `network-interface.vpc-id` - The ID of the VPC for the network interface.
- `network-interface.network-interface.id` - The ID of the network interface.
- `network-interface.owner-id` - The ID of the owner of the network interface.
- `network-interface.availability-zone` - The Availability Zone for the network interface.
- `network-interface.requester-id` - The requester ID for the network interface.
- `network-interface.requester-managed` - Indicates whether the network interface is being managed by AWS.
- `network-interface.status` - The status of the network interface (`available|in-use`).
- `network-interface.mac-address` - The MAC address of the network interface.
- `network-interface-private-dns-name` - The private DNS name of the network interface.
- `network-interface.source-dest-check` - Whether the network interface performs source/destination checking. A value of `true` means checking is enabled, and `false` means checking is disabled. The value must be `false` for the network interface to perform network address translation (NAT) in your VPC.
- `network-interface.group-id` - The ID of a security group associated with the network interface.
- `network-interface.group-name` - The name of a security group associated with the network interface.
- `network-interface.attachment.attachment-id` - The ID of the interface attachment.
- `network-interface.attachment.instance-id` - The ID of the instance to which the network interface is attached.
- `network-interface.attachment.instance-owner-id` - The owner ID of the instance to which the network interface is attached.
- `network-interface.addresses.private-ip-address` - The private IP address associated with the network interface.
- `network-interface.attachment.device-index` - The device index to which the network interface is attached.

- `network-interface.attachment.status` - The status of the attachment (`attaching` | `attached` | `detaching` | `detached`).
- `network-interface.attachment.attach-time` - The time that the network interface was attached to an instance.
- `network-interface.attachment.delete-on-termination` - Specifies whether the attachment is deleted when an instance is terminated.
- `network-interface.addresses.primary` - Specifies whether the IP address of the network interface is the primary private IP address.
- `network-interface.addresses.association.public-ip` - The ID of the association of an Elastic IP address with a network interface.
- `network-interface.addresses.association.ip-owner-id` - The owner ID of the private IP address associated with the network interface.
- `association.public-ip` - The address of the Elastic IP address bound to the network interface.
- `association.ip-owner-id` - The owner of the Elastic IP address associated with the network interface.
- `association.allocation-id` - The allocation ID returned when you allocated the Elastic IP address for your network interface.
- `association.association-id` - The association ID returned when the network interface was associated with an IP address.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

- **NextToken** (*string*) - The token to request the next page of results.
- **MaxResults** (*integer*) - The maximum number of results to return for the request in a single page. The remaining results of the initial request can be seen by sending another request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. You cannot specify this parameter and the `instance IDs` parameter in the same request.

Returns None

class `EC2.Waiter.InstanceTerminated`

```
waiter = client.get_waiter('instance_terminated')
```

wait (***kwargs*)

Polls `EC2.Client.describe_instances()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```

waiter.wait(
    DryRun=True|False,
    InstanceIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    NextToken='string',
    MaxResults=123
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceIds** (*list*) – One or more instance IDs.

Default: Describes all your instances.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `architecture` - The instance architecture (i386|x86_64).
 - `availability-zone` - The Availability Zone of the instance.
 - `block-device-mapping.attach-time` - The attach time for an EBS volume mapped to the instance, for example, 2010-09-15T17:15:20.000Z.
 - `block-device-mapping.delete-on-termination` - A Boolean that indicates whether the EBS volume is deleted on instance termination.
 - `block-device-mapping.device-name` - The device name for the EBS volume (for example, /dev/sdh or xvdh).
 - `block-device-mapping.status` - The status for the EBS volume (attaching|attached|detaching|detached).
 - `block-device-mapping.volume-id` - The volume ID of the EBS volume.
 - `client-token` - The idempotency token you provided when you launched the instance.
 - `dns-name` - The public DNS name of the instance.
 - `group-id` - The ID of the security group for the instance. EC2-Classic only.
 - `group-name` - The name of the security group for the instance. EC2-Classic only.
 - `hypervisor` - The hypervisor type of the instance (ovm|xen).
 - `iam-instance-profile.arn` - The instance profile associated with the instance. Specified as an ARN.
 - `image-id` - The ID of the image used to launch the instance.
 - `instance-id` - The ID of the instance.
 - `instance-lifecycle` - Indicates whether this is a Spot Instance (spot).

- `instance-state-code` - The state of the instance, as a 16-bit unsigned integer. The high byte is an opaque internal value and should be ignored. The low byte is set based on the state represented. The valid values are: 0 (pending), 16 (running), 32 (shutting-down), 48 (terminated), 64 (stopping), and 80 (stopped).
- `instance-state-name` - The state of the instance (pending | running | shutting-down | terminated | stopping | stopped).
- `instance-type` - The type of instance (for example, `t2.micro`).
- `instance.group-id` - The ID of the security group for the instance.
- `instance.group-name` - The name of the security group for the instance.
- `ip-address` - The public IP address of the instance.
- `kernel-id` - The kernel ID.
- `key-name` - The name of the key pair used when the instance was launched.
- `launch-index` - When launching multiple instances, this is the index for the instance in the launch group (for example, 0, 1, 2, and so on).
- `launch-time` - The time when the instance was launched.
- `monitoring-state` - Indicates whether monitoring is enabled for the instance (disabled | enabled).
- `owner-id` - The AWS account ID of the instance owner.
- `placement-group-name` - The name of the placement group for the instance.
- `platform` - The platform. Use windows if you have Windows instances; otherwise, leave blank.
- `private-dns-name` - The private DNS name of the instance.
- `private-ip-address` - The private IP address of the instance.
- `product-code` - The product code associated with the AMI used to launch the instance.
- `product-code.type` - The type of product code (devpay | marketplace).
- `ramdisk-id` - The RAM disk ID.
- `reason` - The reason for the current state of the instance (for example, shows “User Initiated [date]” when you stop or terminate the instance). Similar to the state-reason-code filter.
- `requester-id` - The ID of the entity that launched the instance on your behalf (for example, AWS Management Console, Auto Scaling, and so on).
- `reservation-id` - The ID of the instance’s reservation. A reservation ID is created any time you launch an instance. A reservation ID has a one-to-one relationship with an instance launch request, but can be associated with more than one instance if you launch multiple instances using the same launch request. For example, if you launch one instance, you’ll get one reservation ID. If you launch ten instances using the same launch request, you’ll also get one reservation ID.
- `root-device-name` - The name of the root device for the instance (for example, `/dev/sda1` or `/dev/xvda`).
- `root-device-type` - The type of root device that the instance uses (ebs | instance-store).
- `source-dest-check` - Indicates whether the instance performs source/destination checking. A value of `true` means that checking is enabled, and `false` means checking is disabled. The value must be `false` for the instance to perform network address translation (NAT) in

your VPC.

- `spot-instance-request-id` - The ID of the Spot Instance request.
- `state-reason-code` - The reason code for the state change.
- `state-reason-message` - A message that describes the state change.
- `subnet-id` - The ID of the subnet for the instance.
- `tag :key =*value*` - The key/value combination of a tag assigned to the resource, where `tag :key` is the tag's key.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter "`tag-key=Purpose`" and the filter "`tag-value=X`", you get any resources assigned both the tag key `Purpose` (regardless of what the tag's value is), and the tag value `X` (regardless of what the tag's key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `tenancy` - The tenancy of an instance (`dedicated` | `default`).
- `virtualization-type` - The virtualization type of the instance (`paravirtual` | `hvm`).
- `vpc-id` - The ID of the VPC that the instance is running in.
- `network-interface.description` - The description of the network interface.
- `network-interface.subnet-id` - The ID of the subnet for the network interface.
- `network-interface.vpc-id` - The ID of the VPC for the network interface.
- `network-interface.network-interface.id` - The ID of the network interface.
- `network-interface.owner-id` - The ID of the owner of the network interface.
- `network-interface.availability-zone` - The Availability Zone for the network interface.
- `network-interface.requester-id` - The requester ID for the network interface.
- `network-interface.requester-managed` - Indicates whether the network interface is being managed by AWS.
- `network-interface.status` - The status of the network interface (`available` | `in-use`).
- `network-interface.mac-address` - The MAC address of the network interface.
- `network-interface.private-dns-name` - The private DNS name of the network interface.
- `network-interface.source-dest-check` - Whether the network interface performs source/destination checking. A value of `true` means checking is enabled, and `false` means checking is disabled. The value must be `false` for the network interface to perform network address translation (NAT) in your VPC.
- `network-interface.group-id` - The ID of a security group associated with the network interface.
- `network-interface.group-name` - The name of a security group associated with the network interface.
- `network-interface.attachment.attachment-id` - The ID of the interface attachment.
- `network-interface.attachment.instance-id` - The ID of the instance to which the network interface is attached.

- `network-interface.attachment.instance-owner-id` - The owner ID of the instance to which the network interface is attached.
- `network-interface.addresses.private-ip-address` - The private IP address associated with the network interface.
- `network-interface.attachment.device-index` - The device index to which the network interface is attached.
- `network-interface.attachment.status` - The status of the attachment (`attaching`|`attached`|`detaching`|`detached`).
- `network-interface.attachment.attach-time` - The time that the network interface was attached to an instance.
- `network-interface.attachment.delete-on-termination` - Specifies whether the attachment is deleted when an instance is terminated.
- `network-interface.addresses.primary` - Specifies whether the IP address of the network interface is the primary private IP address.
- `network-interface.addresses.association.public-ip` - The ID of the association of an Elastic IP address with a network interface.
- `network-interface.addresses.association.ip-owner-id` - The owner ID of the private IP address associated with the network interface.
- `association.public-ip` - The address of the Elastic IP address bound to the network interface.
- `association.ip-owner-id` - The owner of the Elastic IP address associated with the network interface.
- `association.allocation-id` - The allocation ID returned when you allocated the Elastic IP address for your network interface.
- `association.association-id` - The association ID returned when the network interface was associated with an IP address.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

- **NextToken** (*string*) - The token to request the next page of results.
- **MaxResults** (*integer*) - The maximum number of results to return for the request in a single page. The remaining results of the initial request can be seen by sending another request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. You cannot specify this parameter and the `instance IDs` parameter in the same request.

Returns None

class `EC2.Waiter.PasswordDataAvailable`

```
waiter = client.get_waiter('password_data_available')
```

wait (***kwargs*)

Polls `EC2.Client.get_password_data()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(
    DryRun=True|False,
    InstanceId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceId** (*string*) – [REQUIRED]

The ID of the Windows instance.

Returns None

class `EC2.Waiter.SnapshotCompleted`

```
waiter = client.get_waiter('snapshot_completed')
```

wait (***kwargs*)

Polls `EC2.Client.describe_snapshots()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(
    DryRun=True|False,
    SnapshotIds=[
        'string',
    ],
    OwnerIds=[
        'string',
    ],
    RestorableByUserIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If

you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **SnapshotIds** (*list*) – One or more snapshot IDs.

Default: Describes snapshots for which you have launch permissions.

– (*string*) –

- **OwnerIds** (*list*) – Returns the snapshots owned by the specified owner. Multiple owners can be specified.

– (*string*) –

- **RestorableByUserIds** (*list*) – One or more AWS accounts IDs that can create volumes from the snapshot.

– (*string*) –

- **Filters** (*list*) – One or more filters.

– `description` - A description of the snapshot.

– `owner-alias` - The AWS account alias (for example, `amazon`) that owns the snapshot.

– `owner-id` - The ID of the AWS account that owns the snapshot.

– `progress` - The progress of the snapshot, as a percentage (for example, 80%).

– `snapshot-id` - The snapshot ID.

– `start-time` - The time stamp when the snapshot was initiated.

– `status` - The status of the snapshot (`pending`|`completed`|`error`).

– `tag :key =*value*` - The key/value combination of a tag assigned to the resource.

– `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag’s value is), and the tag value `X` (regardless of what the tag’s key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.

– `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.

– `volume-id` - The ID of the volume the snapshot is for.

– `volume-size` - The size of the volume, in GiB.

– (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

- **NextToken** (*string*) – The `NextToken` value returned from a previous paginated `DescribeSnapshots` request where `MaxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `NextToken` value. This value is `null` when there are no more results to return.

- **MaxResults** (*integer*) – The maximum number of snapshot results returned by `DescribeSnapshots` in paginated output. When this parameter is used, `DescribeSnapshots` only returns `MaxResults` results in a single page

along with a `NextToken` response element. The remaining results of the initial request can be seen by sending another `DescribeSnapshots` request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. If this parameter is not used, then `DescribeSnapshots` returns all results. You cannot specify this parameter and the snapshot IDs parameter in the same request.

Returns None

class `EC2.Waiter.SpotInstanceRequestFulfilled`

```
waiter = client.get_waiter('spot_instance_request_fulfilled')
```

wait (***kwargs*)

Polls `EC2.Client.describe_spot_instance_requests()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(
    DryRun=True|False,
    SpotInstanceRequestIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SpotInstanceRequestIds** (*list*) – One or more Spot instance request IDs.
 - (*string*) –
- **Filters** (*list*) – One or more filters.
 - `availability-zone-group` - The Availability Zone group.
 - `create-time` - The time stamp when the Spot instance request was created.
 - `fault-code` - The fault code related to the request.
 - `fault-message` - The fault message related to the request.
 - `instance-id` - The ID of the instance that fulfilled the request.
 - `launch-group` - The Spot instance launch group.
 - `launch.block-device-mapping.delete-on-termination` - Indicates whether the Amazon EBS volume is deleted on instance termination.
 - `launch.block-device-mapping.device-name` - The device name for the Amazon EBS volume (for example, `/dev/sdh`).

- `launch.block-device-mapping.snapshot-id` - The ID of the snapshot used for the Amazon EBS volume.
- `launch.block-device-mapping.volume-size` - The size of the Amazon EBS volume, in GiB.
- `launch.block-device-mapping.volume-type` - The type of the Amazon EBS volume (`gp2` | `standard` | `io1`).
- `launch.group-id` - The security group for the instance.
- `launch.image-id` - The ID of the AMI.
- `launch.instance-type` - The type of instance (for example, `m1.small`).
- `launch.kernel-id` - The kernel ID.
- `launch.key-name` - The name of the key pair the instance launched with.
- `launch.monitoring-enabled` - Whether monitoring is enabled for the Spot instance.
- `launch.ramdisk-id` - The RAM disk ID.
- `network-interface.network-interface-id` - The ID of the network interface.
- `network-interface.device-index` - The index of the device for the network interface attachment on the instance.
- `network-interface.subnet-id` - The ID of the subnet for the instance.
- `network-interface.description` - A description of the network interface.
- `network-interface.private-ip-address` - The primary private IP address of the network interface.
- `network-interface.delete-on-termination` - Indicates whether the network interface is deleted when the instance is terminated.
- `network-interface.group-id` - The ID of the security group associated with the network interface.
- `network-interface.group-name` - The name of the security group associated with the network interface.
- `network-interface.addresses.primary` - Indicates whether the IP address is the primary private IP address.
- `product-description` - The product description associated with the instance (`Linux/UNIX` | `Windows`).
- `spot-instance-request-id` - The Spot instance request ID.
- `spot-price` - The maximum hourly price for any Spot instance launched to fulfill the request.
- `state` - The state of the Spot instance request (`open` | `active` | `closed` | `cancelled` | `failed`). Spot bid status information can help you track your Amazon EC2 Spot instance requests. For more information, see [Spot Bid Status](#) in the Amazon Elastic Compute Cloud User Guide.
- `status-code` - The short code describing the most recent evaluation of your Spot instance request.
- `status-message` - The message explaining the status of the Spot instance request.
- `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter `"tag-key=Purpose"` and the filter `"tag-value=X"`, you get any resources assigned both the tag key Purpose (regardless of what the tag's value is), and

the tag value X (regardless of what the tag's key is). If you want to list only resources where Purpose is X, see the `tag:key=*value*` filter.

- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `type` - The type of Spot instance request (`one-time` | `persistent`).
- `launched-availability-zone` - The Availability Zone in which the bid is launched.
- `valid-from` - The start date of the request.
- `valid-until` - The end date of the request.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

Returns None

class EC2.Waiter.**SubnetAvailable**

```
waiter = client.get_waiter('subnet_available')
```

wait (***kwargs*)

Polls `EC2.Client.describe_subnets()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(
    DryRun=True|False,
    SubnetIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
)
```

Parameters

- **DryRun** (*boolean*) - Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SubnetIds** (*list*) - One or more subnet IDs.

Default: Describes all your subnets.

- (*string*) -

- **Filters** (*list*) – One or more filters.
 - `availabilityZone` - The Availability Zone for the subnet. You can also use `availability-zone` as the filter name.
 - `available-ip-address-count` - The number of IP addresses in the subnet that are available.
 - `cidrBlock` - The CIDR block of the subnet. The CIDR block you specify must exactly match the subnet’s CIDR block for information to be returned for the subnet. You can also use `cidr` or `cidr-block` as the filter names.
 - `defaultForAz` - Indicates whether this is the default subnet for the Availability Zone. You can also use `default-for-az` as the filter name.
 - `state` - The state of the subnet (`pending` | `available`).
 - `subnet-id` - The ID of the subnet.
 - `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
 - `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag’s value is), and the tag value `X` (regardless of what the tag’s key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
 - `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
 - `vpc-id` - The ID of the VPC for the subnet.
 - (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

Returns None

class `EC2.Waiter.SystemStatusOk`

```
waiter = client.get_waiter('system_status_ok')
```

wait (***kwargs*)

Polls `EC2.Client.describe_instance_status()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(  
    DryRun=True|False,  
    InstanceIds=[  
        'string',  
    ],  
    Filters=[
```



```

    {
        'Name': 'string',
        'Values': [
            'string',
        ]
    },
],
NextToken='string',
MaxResults=123,
IncludeAllInstances=True|False
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceIds** (*list*) – One or more instance IDs.

Default: Describes all your instances.

Constraints: Maximum 100 explicitly specified instance IDs.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `availability-zone` - The Availability Zone of the instance.
 - `event.code` - The code for the scheduled event (`instance-reboot` | `system-reboot` | `system-maintenance` | `instance-retirement` | `instance-stop`).
 - `event.description` - A description of the event.
 - `event.not-after` - The latest end time for the scheduled event (for example, `2014-09-15T17:15:20.000Z`).
 - `event.not-before` - The earliest start time for the scheduled event (for example, `2014-09-15T17:15:20.000Z`).
 - `instance-state-code` - The code for the instance state, as a 16-bit unsigned integer. The high byte is an opaque internal value and should be ignored. The low byte is set based on the state represented. The valid values are 0 (pending), 16 (running), 32 (shutting-down), 48 (terminated), 64 (stopping), and 80 (stopped).
 - `instance-state-name` - The state of the instance (`pending` | `running` | `shutting-down` | `terminated` | `stopping` | `stopped`).
 - `instance-status.reachability` - Filters on instance status where the name is `reachability` (`passed` | `failed` | `initializing` | `insufficient-data`).
 - `instance-status.status` - The status of the instance (`ok` | `impaired` | `initializing` | `insufficient-data` | `not-applicable`).
 - `system-status.reachability` - Filters on system status where the name is `reachability` (`passed` | `failed` | `initializing` | `insufficient-data`).
 - `system-status.status` - The system status of the instance (`ok` | `impaired` | `initializing` | `insufficient-data` | `not-applicable`).
 - (*dict*) –

A filter name and value pair that is used to return a more specific list of

results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

- * **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

- * **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

- (*string*) –

- **NextToken** (*string*) – The token to retrieve the next page of results.
- **MaxResults** (*integer*) – The maximum number of results to return for the request in a single page. The remaining results of the initial request can be seen by sending another request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. You cannot specify this parameter and the `instance IDs` parameter in the same request.
- **IncludeAllInstances** (*boolean*) – When `true`, includes the health status for all instances. When `false`, includes the health status for running instances only.

Default: `false`

Returns `None`

class `EC2.Waiter.VolumeAvailable`

```
waiter = client.get_waiter('volume_available')
```

wait (***kwargs*)

Polls `EC2.Client.describe_volumes()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(
    DryRun=True|False,
    VolumeIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VolumeIds** (*list*) – One or more volume IDs.
 - (*string*) –
- **Filters** (*list*) – One or more filters.

- `attachment.attach-time` - The time stamp when the attachment initiated.
- `attachment.delete-on-termination` - Whether the volume is deleted on instance termination.
- `attachment.device` - The device name that is exposed to the instance (for example, `/dev/sda1`).
- `attachment.instance-id` - The ID of the instance the volume is attached to.
- `attachment.status` - The attachment state (`attaching` | `attached` | `detaching` | `detached`).
- `availability-zone` - The Availability Zone in which the volume was created.
- `create-time` - The time stamp when the volume was created.
- `encrypted` - The encryption status of the volume.
- `size` - The size of the volume, in GiB.
- `snapshot-id` - The snapshot from which the volume was created.
- `status` - The status of the volume (`creating` | `available` | `in-use` | `deleting` | `deleted` | `error`).
- `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag’s value is), and the tag value `X` (regardless of what the tag’s key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `volume-id` - The volume ID.
- `volume-type` - The Amazon EBS volume type. This can be `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, or `standard` for Magnetic volumes.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

- **NextToken** (*string*) - The `NextToken` value returned from a previous paginated `DescribeVolumes` request where `MaxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `NextToken` value. This value is `null` when there are no more results to return.
- **MaxResults** (*integer*) - The maximum number of volume results returned by `DescribeVolumes` in paginated output. When this parameter is used, `DescribeVolumes` only returns `MaxResults` results in a single page along with a `NextToken` response element. The remaining results of the initial request can be seen by sending another `DescribeVolumes` request with the returned `NextToken` value. This value can be between 5 and 1000; if

`MaxResults` is given a value larger than 1000, only 1000 results are returned. If this parameter is not used, then `DescribeVolumes` returns all results. You cannot specify this parameter and the `volume IDs` parameter in the same request.

Returns None

class `EC2.Waiter.VolumeDeleted`

```
waiter = client.get_waiter('volume_deleted')
```

wait (***kwargs*)

Polls `EC2.Client.describe_volumes()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(
    DryRun=True|False,
    VolumeIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VolumeIds** (*list*) – One or more volume IDs.
 - (*string*) –
- **Filters** (*list*) – One or more filters.
 - `attachment.attach-time` - The time stamp when the attachment initiated.
 - `attachment.delete-on-termination` - Whether the volume is deleted on instance termination.
 - `attachment.device` - The device name that is exposed to the instance (for example, `/dev/sda1`).
 - `attachment.instance-id` - The ID of the instance the volume is attached to.
 - `attachment.status` - The attachment state (`attaching` | `attached` | `detaching` | `detached`).
 - `availability-zone` - The Availability Zone in which the volume was created.
 - `create-time` - The time stamp when the volume was created.
 - `encrypted` - The encryption status of the volume.
 - `size` - The size of the volume, in GiB.

- `snapshot-id` - The snapshot from which the volume was created.
- `status` - The status of the volume (`creating`|`available`|`in-use`|`deleting`|`deleted`|`error`).
- `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag’s value is), and the tag value `X` (regardless of what the tag’s key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `volume-id` - The volume ID.
- `volume-type` - The Amazon EBS volume type. This can be `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, or `standard` for Magnetic volumes.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

- **NextToken** (*string*) - The `NextToken` value returned from a previous paginated `DescribeVolumes` request where `MaxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `NextToken` value. This value is `null` when there are no more results to return.
- **MaxResults** (*integer*) - The maximum number of volume results returned by `DescribeVolumes` in paginated output. When this parameter is used, `DescribeVolumes` only returns `MaxResults` results in a single page along with a `NextToken` response element. The remaining results of the initial request can be seen by sending another `DescribeVolumes` request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. If this parameter is not used, then `DescribeVolumes` returns all results. You cannot specify this parameter and the volume IDs parameter in the same request.

Returns None

class `EC2.Waiter.VolumeInUse`

```
waiter = client.get_waiter('volume_in_use')
```

wait (***kwargs*)

Polls `EC2.Client.describe_volumes()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(
    DryRun=True|False,
    VolumeIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VolumeIds** (*list*) – One or more volume IDs.
 - (*string*) –
- **Filters** (*list*) – One or more filters.
 - `attachment.attach-time` - The time stamp when the attachment initiated.
 - `attachment.delete-on-termination` - Whether the volume is deleted on instance termination.
 - `attachment.device` - The device name that is exposed to the instance (for example, `/dev/sda1`).
 - `attachment.instance-id` - The ID of the instance the volume is attached to.
 - `attachment.status` - The attachment state (`attaching` | `attached` | `detaching` | `detached`).
 - `availability-zone` - The Availability Zone in which the volume was created.
 - `create-time` - The time stamp when the volume was created.
 - `encrypted` - The encryption status of the volume.
 - `size` - The size of the volume, in GiB.
 - `snapshot-id` - The snapshot from which the volume was created.
 - `status` - The status of the volume (`creating` | `available` | `in-use` | `deleting` | `deleted` | `error`).
 - `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
 - `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag’s value is), and the tag value `X` (regardless of what the tag’s key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
 - `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
 - `volume-id` - The volume ID.

- `volume-type` - The Amazon EBS volume type. This can be `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, or `standard` for Magnetic volumes.

- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

- **NextToken** (*string*) - The `NextToken` value returned from a previous paginated `DescribeVolumes` request where `MaxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `NextToken` value. This value is `null` when there are no more results to return.
- **MaxResults** (*integer*) - The maximum number of volume results returned by `DescribeVolumes` in paginated output. When this parameter is used, `DescribeVolumes` only returns `MaxResults` results in a single page along with a `NextToken` response element. The remaining results of the initial request can be seen by sending another `DescribeVolumes` request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. If this parameter is not used, then `DescribeVolumes` returns all results. You cannot specify this parameter and the volume IDs parameter in the same request.

Returns None

class `EC2.Waiter.VpcAvailable`

```
waiter = client.get_waiter('vpc_available')
```

wait (***kwargs*)

Polls `EC2.Client.describe_vpcs()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(
    DryRun=True|False,
    VpcIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcIds** (*list*) – One or more VPC IDs.

Default: Describes all your VPCs.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `cidr` - The CIDR block of the VPC. The CIDR block you specify must exactly match the VPC's CIDR block for information to be returned for the VPC. Must contain the slash followed by one or two digits (for example, `/28`).
 - `dhcp-options-id` - The ID of a set of DHCP options.
 - `isDefault` - Indicates whether the VPC is the default VPC.
 - `state` - The state of the VPC (`pending` | `available`).
 - `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
 - `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag's value is), and the tag value `X` (regardless of what the tag's key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
 - `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
 - `vpc-id` - The ID of the VPC.
- (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

Returns None

class `EC2.Waiter.VpnConnectionAvailable`

```
waiter = client.get_waiter('vpn_connection_available')
```

wait (***kwargs*)

Polls `EC2.Client.describe_vpn_connections()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(  
    DryRun=True|False,  
    VpnConnectionIds=[
```



```

        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpnConnectionIds** (*list*) – One or more VPN connection IDs.

Default: Describes your VPN connections.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `customer-gateway-configuration` - The configuration information for the customer gateway.
 - `customer-gateway-id` - The ID of a customer gateway associated with the VPN connection.
 - `state` - The state of the VPN connection (`pending` | `available` | `deleting` | `deleted`).
 - `option.static-routes-only` - Indicates whether the connection has static routes only. Used for devices that do not support Border Gateway Protocol (BGP).
 - `route.destination-cidr-block` - The destination CIDR block. This corresponds to the subnet used in a customer data center.
 - `bgp-asn` - The BGP Autonomous System Number (ASN) associated with a BGP device.
 - `tag:key=*value*` - The key/value combination of a tag assigned to the resource.
 - `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag’s value is), and the tag value `X` (regardless of what the tag’s key is). If you want to list only resources where `Purpose` is `X`, see the `tag:key=*value*` filter.
 - `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
 - `type` - The type of VPN connection. Currently the only supported type is `ipsec.1`.
 - `vpn-connection-id` - The ID of the VPN connection.
 - `vpn-gateway-id` - The ID of a virtual private gateway associated with the VPN connection.
- (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

- * **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

- * **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

- (*string*) –

Returns None

class EC2.Waiter.VpnConnectionDeleted

```
waiter = client.get_waiter('vpn_connection_deleted')
```

wait (***kwargs*)

Polls `EC2.Client.describe_vpn_connections()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(
    DryRun=True|False,
    VpnConnectionIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpnConnectionIds** (*list*) – One or more VPN connection IDs.

Default: Describes your VPN connections.

- (*string*) –

- **Filters** (*list*) – One or more filters.
 - `customer-gateway-configuration` - The configuration information for the customer gateway.
 - `customer-gateway-id` - The ID of a customer gateway associated with the VPN connection.
 - `state` - The state of the VPN connection (`pending` | `available` | `deleting` | `deleted`).
 - `option.static-routes-only` - Indicates whether the connection has static routes only. Used for devices that do not support Border Gateway Protocol (BGP).
 - `route.destination-cidr-block` - The destination CIDR block. This corresponds to the subnet used in a customer data center.
 - `bgp-asn` - The BGP Autonomous System Number (ASN) associated with a BGP device.

- `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag’s value is), and the tag value `X` (regardless of what the tag’s key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `type` - The type of VPN connection. Currently the only supported type is `ipsec.1`.
- `vpn-connection-id` - The ID of the VPN connection.
- `vpn-gateway-id` - The ID of a virtual private gateway associated with the VPN connection.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

Returns None

Service Resource

class `EC2.ServiceResource`

A resource representing Amazon Elastic Compute Cloud (EC2):

```
import boto3
```

```
ec2 = boto3.resource('ec2')
```

These are the resource’s available actions:

- `create_dhcp_options()`
- `create_instances()`
- `create_internet_gateway()`
- `create_key_pair()`
- `create_network_acl()`
- `create_network_interface()`
- `create_placement_group()`
- `create_route_table()`
- `create_security_group()`
- `create_snapshot()`
- `create_subnet()`
- `create_tags()`
- `create_volume()`
- `create_vpc()`
- `create_vpc_peering_connection()`

- `disassociate_route_table()`
- `import_key_pair()`
- `register_image()`

These are the resource's available sub-resources:

- `DhcpOptions()`
- `Image()`
- `Instance()`
- `InternetGateway()`
- `KeyPair()`
- `NetworkAcl()`
- `NetworkInterface()`
- `PlacementGroup()`
- `RouteTable()`
- `RouteTableAssociation()`
- `SecurityGroup()`
- `Snapshot()`
- `Subnet()`
- `Tag()`
- `Volume()`
- `Vpc()`
- `VpcPeeringConnection()`

These are the resource's available collections:

- `dhcp_options_sets`
- `images`
- `instances`
- `internet_gateways`
- `key_pairs`
- `network_acls`
- `network_interfaces`
- `placement_groups`
- `route_tables`
- `security_groups`
- `snapshots`
- `subnets`
- `volumes`
- `vpc_peering_connections`
- `vpcs`

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

`create_dhcp_options` (***kwargs*)

Creates a set of DHCP options for your VPC. After creating the set, you must associate it with the VPC, causing all existing and new instances that you launch in the VPC to use this set of DHCP options. The following are the individual DHCP options you can specify. For more information about the options, see [RFC 2132](#).

- `domain-name-servers` - The IP addresses of up to four domain name servers, or `AmazonProvidedDNS`. The default DHCP option set specifies `AmazonProvidedDNS`. If specifying more than one domain name server, specify the IP addresses in a single parameter, separated by commas.
- `domain-name` - If you're using `AmazonProvidedDNS` in `us-east-1`, specify `ec2.internal`. If you're using `AmazonProvidedDNS` in another region, specify `region.compute.internal` (for example, `ap-northeast-1.compute.internal`). Otherwise, specify a domain name (for example, `MyCompany.com`). **Important** : Some Linux

operating systems accept multiple domain names separated by spaces. However, Windows and other Linux operating systems treat the value as a single domain, which results in unexpected behavior. If your DHCP options set is associated with a VPC that has instances with multiple operating systems, specify only one domain name.

- `ntp-servers` - The IP addresses of up to four Network Time Protocol (NTP) servers.
- `netbios-name-servers` - The IP addresses of up to four NetBIOS name servers.
- `netbios-node-type` - The NetBIOS node type (1, 2, 4, or 8). We recommend that you specify 2 (broadcast and multicast are not currently supported). For more information about these node types, see [RFC 2132](#).

Your VPC automatically starts out with a set of DHCP options that includes only a DNS server that we provide (AmazonProvidedDNS). If you create a set of options, and if your VPC has an Internet gateway, make sure to set the `domain-name-servers` option either to AmazonProvidedDNS or to a domain name server of your choice. For more information about DHCP options, see [DHCP Options Sets](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
dhcp_options = ec2.create_dhcp_options(
    DryRun=True|False,
    DhcpConfigurations=[
        {
            'Key': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **DhcpConfigurations** (*list*) – **[REQUIRED]**

A DHCP configuration option.

– (*dict*) –

- * **Key** (*string*) –
- * **Values** (*list*) –
- (*string*) –

Return type `ec2.DhcpOptions`

Returns DhcpOptions resource

create_instances (***kwargs*)

Launches the specified number of instances using an AMI for which you have permissions.

When you launch an instance, it enters the `pending` state. After the instance is ready for you, it enters the `running` state. To check the state of your instance, call `DescribeInstances`.

If you don't specify a security group when launching an instance, Amazon EC2 uses the default security group. For more information, see [Security Groups](#) in the *Amazon Elastic Compute Cloud User Guide*.

[EC2-VPC only accounts] If you don't specify a subnet in the request, we choose a default subnet from your default VPC for you.

[EC2-Classic accounts] If you're launching into EC2-Classic and you don't specify an Availability Zone, we choose one for you.

Linux instances have access to the public key of the key pair at boot. You can use this key to provide secure access to the instance. Amazon EC2 public images use this feature to provide secure access without passwords. For more information, see [Key Pairs](#) in the *Amazon Elastic Compute Cloud User Guide* .

You can provide optional user data when launching an instance. For more information, see [Instance Metadata](#) in the *Amazon Elastic Compute Cloud User Guide* .

If any of the AMIs have a product code attached for which the user has not subscribed, `RunInstances` fails.

T2 instance types can only be launched into a VPC. If you do not have a default VPC, or if you do not specify a subnet ID in the request, `RunInstances` fails.

For more information about troubleshooting, see [What To Do If An Instance Immediately Terminates](#) , and [Troubleshooting Connecting to Your Instance](#) in the *Amazon Elastic Compute Cloud User Guide* .

Request Syntax

```
instance = ec2.create_instances(  
    DryRun=True|False,  
    ImageId='string',  
    MinCount=123,  
    MaxCount=123,  
    KeyName='string',  
    SecurityGroups=[  
        'string',  
    ],  
    SecurityGroupIds=[  
        'string',  
    ],  
    UserData='string',  
    InstanceType='t1.micro'|'m1.small'|'m1.medium'|'m1.large'|'m1.xlarge'|'m3.medium'|'m3.l  
    Placement={  
        'AvailabilityZone': 'string',  
        'GroupName': 'string',  
        'Tenancy': 'default'|'dedicated'  
    },  
    KernelId='string',  
    RamdiskId='string',  
    BlockDeviceMappings=[  
        {  
            'VirtualName': 'string',  
            'DeviceName': 'string',  
            'Ebs': {  
                'SnapshotId': 'string',  
                'VolumeSize': 123,  
                'DeleteOnTermination': True|False,  
                'VolumeType': 'standard'|'io1'|'gp2',  
                'Iops': 123,  
                'Encrypted': True|False  
            },  
            'NoDevice': 'string'  
        },  
    ],  
    Monitoring={  
        'Enabled': True|False  
    },  
    SubnetId='string',
```

```

DisableApiTermination=True|False,
InstanceInitiatedShutdownBehavior='stop'|'terminate',
PrivateIpAddress='string',
ClientToken='string',
AdditionalInfo='string',
NetworkInterfaces=[
    {
        'NetworkInterfaceId': 'string',
        'DeviceIndex': 123,
        'SubnetId': 'string',
        'Description': 'string',
        'PrivateIpAddress': 'string',
        'Groups': [
            'string',
        ],
        'DeleteOnTermination': True|False,
        'PrivateIpAddresses': [
            {
                'PrivateIpAddress': 'string',
                'Primary': True|False
            },
        ],
        'SecondaryPrivateIpAddressCount': 123,
        'AssociatePublicIpAddress': True|False
    },
],
IamInstanceProfile={
    'Arn': 'string',
    'Name': 'string'
},
EbsOptimized=True|False
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **ImageId** (*string*) – **[REQUIRED]**

The ID of the AMI, which you can get by calling `DescribeImages`.

- **MinCount** (*integer*) – **[REQUIRED]**

The minimum number of instances to launch. If you specify a minimum that is more instances than Amazon EC2 can launch in the target Availability Zone, Amazon EC2 launches no instances.

Constraints: Between 1 and the maximum number you're allowed for the specified instance type. For more information about the default limits, and how to request an increase, see [How many instances can I run in Amazon EC2](#) in the Amazon EC2 General FAQ.

- **MaxCount** (*integer*) – **[REQUIRED]**

The maximum number of instances to launch. If you specify more instances than Amazon EC2 can launch in the target Availability Zone, Amazon EC2 launches the largest possible number of instances above `MinCount`.

Constraints: Between 1 and the maximum number you're allowed for the specified instance type. For more information about the default limits, and how to

request an increase, see [How many instances can I run in Amazon EC2](#) in the Amazon EC2 General FAQ.

- **KeyName** (*string*) – The name of the key pair. You can create a key pair using `CreateKeyPair` or `ImportKeyPair`.

Warning: If you do not specify a key pair, you can't connect to the instance unless you choose an AMI that is configured to allow users another way to log in.

- **SecurityGroups** (*list*) – [EC2-Classic, default VPC] One or more security group names. For a nondefault VPC, you must use security group IDs instead.

Default: Amazon EC2 uses the default security group.

– (*string*) –

- **SecurityGroupIds** (*list*) – One or more security group IDs. You can create a security group using `CreateSecurityGroup`.

Default: Amazon EC2 uses the default security group.

– (*string*) –

- **UserData** (*string*) – The Base64-encoded MIME user data for the instances.

This value will be base64 encoded automatically. Do not base64 encode this value prior to performing the operation.

- **InstanceType** (*string*) – The instance type. For more information, see [Instance Types](#) in the *Amazon Elastic Compute Cloud User Guide*.

Default: `m1.small`

- **Placement** (*dict*) – The placement for the instance.

– **AvailabilityZone** (*string*) –

The Availability Zone of the instance.

– **GroupName** (*string*) –

The name of the placement group the instance is in (for cluster compute instances).

– **Tenancy** (*string*) –

The tenancy of the instance (if the instance is running in a VPC). An instance with a tenancy of `dedicated` runs on single-tenant hardware.

- **KernelId** (*string*) – The ID of the kernel.

Warning: We recommend that you use PV-GRUB instead of kernels and RAM disks. For more information, see [PV-GRUB](#) in the *Amazon Elastic Compute Cloud User Guide*.

- **RamdiskId** (*string*) – The ID of the RAM disk.

Warning: We recommend that you use PV-GRUB instead of kernels and RAM disks. For more information, see [PV-GRUB](#) in the *Amazon Elastic Compute Cloud User Guide*.

- **BlockDeviceMappings** (*list*) – The block device mapping.

– (*dict*) –

Describes a block device mapping.

* **VirtualName** (*string*) –

The virtual device name (`ephemeral N`). Instance store volumes are numbered starting from 0. An instance type with 2 available instance store volumes can specify mappings for `ephemeral0` and

`ephemeral1`. The number of available instance store volumes depends on the instance type. After you connect to the instance, you must mount the volume.

Constraints: For M3 instances, you must specify instance store volumes in the block device mapping for the instance. When you launch an M3 instance, we ignore any instance store volumes specified in the block device mapping for the AMI.

* **DeviceName** (*string*) –

The device name exposed to the instance (for example, `/dev/sdh` or `xvdh`).

* **Ebs** (*dict*) –

Parameters used to automatically set up EBS volumes when the instance is launched.

· **SnapshotId** (*string*) –

The ID of the snapshot.

· **VolumeSize** (*integer*) –

The size of the volume, in GiB.

Constraints: 1–1024 for standard volumes, 1–16384 for gp2 volumes, and 4–16384 for io1 volumes. If you specify a snapshot, the volume size must be equal to or larger than the snapshot size.

Default: If you're creating the volume from a snapshot and don't specify a volume size, the default is the snapshot size.

· **DeleteOnTermination** (*boolean*) –

Indicates whether the EBS volume is deleted on instance termination.

· **VolumeType** (*string*) –

The volume type. `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, and `standard` for Magnetic volumes.

Default: `standard`

· **Iops** (*integer*) –

The number of I/O operations per second (IOPS) that the volume supports. For Provisioned IOPS (SSD) volumes, this represents the number of IOPS that are provisioned for the volume. For General Purpose (SSD) volumes, this represents the baseline performance of the volume and the rate at which the volume accumulates I/O credits for bursting. For more information on General Purpose (SSD) baseline performance, I/O credits, and bursting, see [Amazon EBS Volume Types](#) in the *Amazon Elastic Compute Cloud User Guide*.

Constraint: Range is 100 to 20000 for Provisioned IOPS (SSD) volumes and 3 to 10000 for General Purpose (SSD) volumes.

Condition: This parameter is required for requests to create `io1` volumes; it is not used in requests to create `standard` or `gp2` volumes.

- **Encrypted** (*boolean*) –
Indicates whether the EBS volume is encrypted. Encrypted Amazon EBS volumes may only be attached to instances that support Amazon EBS encryption.
- * **NoDevice** (*string*) –
Suppresses the specified device included in the block device mapping of the AMI.
- **Monitoring** (*dict*) – The monitoring for the instance.
 - **Enabled** (*boolean*) – [REQUIRED]
Indicates whether monitoring is enabled for the instance.
- **SubnetId** (*string*) – [EC2-VPC] The ID of the subnet to launch the instance into.
- **DisableApiTermination** (*boolean*) – If you set this parameter to `true`, you can't terminate the instance using the Amazon EC2 console, CLI, or API; otherwise, you can. If you set this parameter to `true` and then later want to be able to terminate the instance, you must first change the value of the `disableApiTermination` attribute to `false` using `ModifyInstanceAttribute`. Alternatively, if you set `InstanceInitiatedShutdownBehavior` to `terminate`, you can terminate the instance by running the shutdown command from the instance.
Default: `false`
- **InstanceInitiatedShutdownBehavior** (*string*) – Indicates whether an instance stops or terminates when you initiate shutdown from the instance (using the operating system command for system shutdown).
Default: `stop`
- **PrivateIpAddress** (*string*) – [EC2-VPC] The primary IP address. You must specify a value from the IP address range of the subnet.
Only one private IP address can be designated as primary. Therefore, you can't specify this parameter if `PrivateIpAddresses.n.Primary` is set to `true` and `PrivateIpAddresses.n.PrivateIpAddress` is set to an IP address.
Default: We select an IP address from the IP address range of the subnet.
- **ClientToken** (*string*) – Unique, case-sensitive identifier you provide to ensure the idempotency of the request. For more information, see [Ensuring Idempotency](#).
Constraints: Maximum 64 ASCII characters
- **AdditionalInfo** (*string*) – Reserved.
- **NetworkInterfaces** (*list*) – One or more network interfaces.
 - (*dict*) –
Describes a network interface.
 - * **NetworkInterfaceId** (*string*) –
The ID of the network interface.
 - * **DeviceIndex** (*integer*) –
The index of the device on the instance for the network interface attachment. If you are specifying a network interface in a `RunInstances` request, you must provide the device index.
 - * **SubnetId** (*string*) –
The ID of the subnet associated with the network string. Applies only if creating a network interface when launching an instance.

- * **Description** (*string*) –
The description of the network interface. Applies only if creating a network interface when launching an instance.
- * **PrivateIpAddress** (*string*) –
The private IP address of the network interface. Applies only if creating a network interface when launching an instance.
- * **Groups** (*list*) –
The IDs of the security groups for the network interface. Applies only if creating a network interface when launching an instance.
 - (*string*) –
- * **DeleteOnTermination** (*boolean*) –
If set to `true`, the interface is deleted when the instance is terminated. You can specify `true` only if creating a new network interface when launching an instance.
- * **PrivateIpAddresses** (*list*) –
One or more private IP addresses to assign to the network interface. Only one private IP address can be designated as primary.
 - (*dict*) –
Describes a secondary private IP address for a network interface.
 - **PrivateIpAddress** (*string*) – [REQUIRED]
The private IP addresses.
 - **Primary** (*boolean*) –
Indicates whether the private IP address is the primary private IP address. Only one IP address can be designated as primary.
- * **SecondaryPrivateIpAddressCount** (*integer*) –
The number of secondary private IP addresses. You can't specify this option and specify more than one private IP address using the private IP addresses option.
- * **AssociatePublicIpAddress** (*boolean*) –
Indicates whether to assign a public IP address to an instance you launch in a VPC. The public IP address can only be assigned to a network interface for `eth0`, and can only be assigned to a new network interface, not an existing one. You cannot specify more than one network interface in the request. If launching into a default subnet, the default value is `true`.
- **IamInstanceProfile** (*dict*) – The IAM instance profile.
 - **Arn** (*string*) –
The Amazon Resource Name (ARN) of the instance profile.
 - **Name** (*string*) –
The name of the instance profile.
- **EbsOptimized** (*boolean*) – Indicates whether the instance is optimized for EBS I/O. This optimization provides dedicated throughput to Amazon EBS and an optimized configuration stack to provide optimal EBS I/O performance. This optimization isn't available with all instance types. Additional usage charges apply when using an EBS-optimized instance.

Default: `false`

Return type `list(ec2.Instance)`

Returns A list of Instance resources

create_internet_gateway (***kwargs*)

Creates an Internet gateway for use with a VPC. After creating the Internet gateway, you attach it to a VPC using `AttachInternetGateway`.

For more information about your VPC and Internet gateway, see the [Amazon Virtual Private Cloud User Guide](#).

Request Syntax

```
internet_gateway = ec2.create_internet_gateway(  
    DryRun=True|False  
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type `ec2.InternetGateway`

Returns InternetGateway resource

create_key_pair (***kwargs*)

Creates a 2048-bit RSA key pair with the specified name. Amazon EC2 stores the public key and displays the private key for you to save to a file. The private key is returned as an unencrypted PEM encoded PKCS#8 private key. If a key with the specified name already exists, Amazon EC2 returns an error.

You can have up to five thousand key pairs per region.

The key pair returned to you is available only in the region in which you create it. To create a key pair that is available in all regions, use `ImportKeyPair`.

For more information about key pairs, see [Key Pairs](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
key_pair = ec2.create_key_pair(  
    DryRun=True|False,  
    KeyName='string'  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **KeyName** (*string*) – **[REQUIRED]**

A unique name for the key pair.

Constraints: Up to 255 ASCII characters

Return type `ec2.KeyPair`

Returns KeyPair resource

create_network_acl (***kwargs*)

Creates a network ACL in a VPC. Network ACLs provide an optional layer of security (in addition to security groups) for the instances in your VPC.

For more information about network ACLs, see [Network ACLs](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
network_acl = ec2.create_network_acl(
    DryRun=True|False,
    VpcId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcId** (*string*) – **[REQUIRED]**

The ID of the VPC.

Return type `ec2.NetworkAcl`

Returns NetworkAcl resource

create_network_interface (***kwargs*)

Creates a network interface in the specified subnet.

For more information about network interfaces, see [Elastic Network Interfaces](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
network_interface = ec2.create_network_interface(
    SubnetId='string',
    Description='string',
    PrivateIpAddress='string',
    Groups=[
        'string',
    ],
    PrivateIpAddresses=[
        {
            'PrivateIpAddress': 'string',
            'Primary': True|False
        },
    ],
    SecondaryPrivateIpAddressCount=123,
    DryRun=True|False
)
```

Parameters

- **SubnetId** (*string*) – **[REQUIRED]**
- **Description** (*string*) – A description for the network interface.
- **PrivateIpAddress** (*string*) – The primary private IP address of the network interface. If you don't specify an IP address, Amazon EC2 selects one for you from the subnet range. If you specify an IP address, you cannot indicate any IP addresses specified in `privateIpAddresses` as primary (only one IP address can be designated as primary).
- **Groups** (*list*) – The IDs of one or more security groups.
 - (*string*) –
- **PrivateIpAddresses** (*list*) – One or more private IP addresses.
 - (*dict*) –

Describes a secondary private IP address for a network interface.

- * **PrivateIpAddress** (*string*) – [REQUIRED]

The private IP addresses.

- * **Primary** (*boolean*) –

Indicates whether the private IP address is the primary private IP address. Only one IP address can be designated as primary.

- **SecondaryPrivateIpAddressCount** (*integer*) – The number of secondary private IP addresses to assign to a network interface. When you specify a number of secondary IP addresses, Amazon EC2 selects these IP addresses within the subnet range. You can't specify this option and specify more than one private IP address using `privateIpAddresses`.

The number of IP addresses you can assign to a network interface varies by instance type. For more information, see [Private IP Addresses Per ENI Per Instance Type](#) in the *Amazon Elastic Compute Cloud User Guide*.

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type `ec2.NetworkInterface`

Returns `NetworkInterface` resource

create_placement_group (***kwargs*)

Creates a placement group that you launch cluster instances into. You must give the group a name that's unique within the scope of your account.

For more information about placement groups and cluster instances, see [Cluster Instances](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
placement_group = ec2.create_placement_group(  
    DryRun=True|False,  
    GroupName='string',  
    Strategy='cluster'  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **GroupName** (*string*) – [REQUIRED]

A name for the placement group.

Constraints: Up to 255 ASCII characters

- **Strategy** (*string*) – [REQUIRED]

The placement strategy.

Return type `ec2.PlacementGroup`

Returns `PlacementGroup` resource

create_route_table (***kwargs*)

Creates a route table for the specified VPC. After you create a route table, you can add routes and associate the table with a subnet.

For more information about route tables, see [Route Tables](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
route_table = ec2.create_route_table(
    DryRun=True|False,
    VpcId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcId** (*string*) – **[REQUIRED]**

The ID of the VPC.

Return type `ec2.RouteTable`

Returns `RouteTable` resource

create_security_group (***kwargs*)

Creates a security group.

A security group is for use with instances either in the EC2-Classic platform or in a specific VPC. For more information, see [Amazon EC2 Security Groups](#) in the *Amazon Elastic Compute Cloud User Guide* and [Security Groups for Your VPC](#) in the *Amazon Virtual Private Cloud User Guide*.

Warning: EC2-Classic: You can have up to 500 security groups.
EC2-VPC: You can create up to 100 security groups per VPC.

When you create a security group, you specify a friendly name of your choice. You can have a security group for use in EC2-Classic with the same name as a security group for use in a VPC. However, you can't have two security groups for use in EC2-Classic with the same name or two security groups for use in a VPC with the same name.

You have a default security group for use in EC2-Classic and a default security group for use in your VPC. If you don't specify a security group when you launch an instance, the instance is launched into the appropriate default security group. A default security group includes a default rule that grants instances unrestricted network access to each other.

You can add or remove rules from your security groups using `AuthorizeSecurityGroupIngress`, `AuthorizeSecurityGroupEgress`, `RevokeSecurityGroupIngress`, and `RevokeSecurityGroupEgress`.

Request Syntax

```
security_group = ec2.create_security_group(
    DryRun=True|False,
    GroupName='string',
    Description='string',
    VpcId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **GroupName** (*string*) – **[REQUIRED]**

The name of the security group.

Constraints: Up to 255 characters in length

Constraints for EC2-Classic: ASCII characters

Constraints for EC2-VPC: a-z, A-Z, 0-9, spaces, and . _ - : / () # , @ [] + = ; { } ! \$ *

- **Description** (*string*) – [REQUIRED]

A description for the security group. This is informational only.

Constraints: Up to 255 characters in length

Constraints for EC2-Classic: ASCII characters

Constraints for EC2-VPC: a-z, A-Z, 0-9, spaces, and . _ - : / () # , @ [] + = ; { } ! \$ *

- **VpcId** (*string*) – [EC2-VPC] The ID of the VPC. Required for EC2-VPC.

Return type `ec2.SecurityGroup`

Returns SecurityGroup resource

create_snapshot (***kwargs*)

Creates a snapshot of an EBS volume and stores it in Amazon S3. You can use snapshots for backups, to make copies of EBS volumes, and to save data before shutting down an instance.

When a snapshot is created, any AWS Marketplace product codes that are associated with the source volume are propagated to the snapshot.

You can take a snapshot of an attached volume that is in use. However, snapshots only capture data that has been written to your EBS volume at the time the snapshot command is issued; this may exclude any data that has been cached by any applications or the operating system. If you can pause any file systems on the volume long enough to take a snapshot, your snapshot should be complete. However, if you cannot pause all file writes to the volume, you should unmount the volume from within the instance, issue the snapshot command, and then remount the volume to ensure a consistent and complete snapshot. You may remount and use your volume while the snapshot status is `pending`.

To create a snapshot for EBS volumes that serve as root devices, you should stop the instance before taking the snapshot.

Snapshots that are taken from encrypted volumes are automatically encrypted. Volumes that are created from encrypted snapshots are also automatically encrypted. Your encrypted volumes and any associated snapshots always remain protected.

For more information, see [Amazon Elastic Block Store](#) and [Amazon EBS Encryption](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
snapshot = ec2.create_snapshot(  
    DryRun=True|False,  
    VolumeId='string',  
    Description='string'  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VolumeId** (*string*) – [REQUIRED]
The ID of the EBS volume.
- **Description** (*string*) – A description for the snapshot.

Return type `ec2.Snapshot`

Returns Snapshot resource

create_subnet (***kwargs*)

Creates a subnet in an existing VPC.

When you create each subnet, you provide the VPC ID and the CIDR block you want for the subnet. After you create a subnet, you can't change its CIDR block. The subnet's CIDR block can be the same as the VPC's CIDR block (assuming you want only a single subnet in the VPC), or a subset of the VPC's CIDR block. If you create more than one subnet in a VPC, the subnets' CIDR blocks must not overlap. The smallest subnet (and VPC) you can create uses a /28 netmask (16 IP addresses), and the largest uses a /16 netmask (65,536 IP addresses).

Warning: AWS reserves both the first four and the last IP address in each subnet's CIDR block. They're not available for use.

If you add more than one subnet to a VPC, they're set up in a star topology with a logical router in the middle.

If you launch an instance in a VPC using an Amazon EBS-backed AMI, the IP address doesn't change if you stop and restart the instance (unlike a similar instance launched outside a VPC, which gets a new IP address when restarted). It's therefore possible to have a subnet with no running instances (they're all stopped), but no remaining IP addresses available.

For more information about subnets, see [Your VPC and Subnets](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
subnet = ec2.create_subnet(
    DryRun=True|False,
    VpcId='string',
    CidrBlock='string',
    AvailabilityZone='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcId** (*string*) – **[REQUIRED]**
The ID of the VPC.
- **CidrBlock** (*string*) – **[REQUIRED]**
The network range for the subnet, in CIDR notation. For example, `10.0.0.0/24`.
- **AvailabilityZone** (*string*) – The Availability Zone for the subnet.
Default: Amazon EC2 selects one for you (recommended).

Return type `ec2.Subnet`

Returns Subnet resource

create_tags (***kwargs*)

Adds or overwrites one or more tags for the specified Amazon EC2 resource or resources. Each resource can have a maximum of 10 tags. Each tag consists of a key and optional value. Tag keys must be unique per resource.

For more information about tags, see [Tagging Your Resources](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
tag = ec2.create_tags(  
    DryRun=True|False,  
    Resources=[  
        'string',  
    ],  
    Tags=[  
        {  
            'Key': 'string',  
            'Value': 'string'  
        },  
    ]  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Resources** (*list*) – **[REQUIRED]**

The IDs of one or more resources to tag. For example, `ami-1a2b3c4d`.

– (*string*) –

- **Tags** (*list*) – **[REQUIRED]**

One or more tags. The `value` parameter is required, but if you don't want the tag to have a value, specify the parameter with no value, and we set the value to an empty string.

– (*dict*) –

Describes a tag.

* **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

* **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

Return type `list(ec2.Tag)`

Returns A list of Tag resources

create_volume (***kwargs*)

Creates an EBS volume that can be attached to an instance in the same Availability Zone. The volume is created in the regional endpoint that you send the HTTP request to. For more information see [Regions and Endpoints](#).

You can create a new empty volume or restore a volume from an EBS snapshot. Any AWS Marketplace product codes from the snapshot are propagated to the volume.

You can create encrypted volumes with the `Encrypted` parameter. Encrypted volumes may only be attached to instances that support Amazon EBS encryption. Volumes that are created from encrypted

snapshots are also automatically encrypted. For more information, see [Amazon EBS Encryption](#) in the *Amazon Elastic Compute Cloud User Guide*.

For more information, see [Creating or Restoring an Amazon EBS Volume](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
volume = ec2.create_volume(
    DryRun=True|False,
    Size=123,
    SnapshotId='string',
    AvailabilityZone='string',
    VolumeType='standard'|'io1'|'gp2',
    Iops=123,
    Encrypted=True|False,
    KmsKeyId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Size** (*integer*) – The size of the volume, in GiBs.

Constraints: 1-1024 for standard volumes, 1-16384 for gp2 volumes, and 4-16384 for io1 volumes. If you specify a snapshot, the volume size must be equal to or larger than the snapshot size.

Default: If you're creating the volume from a snapshot and don't specify a volume size, the default is the snapshot size.

- **SnapshotId** (*string*) – The snapshot from which to create the volume.
- **AvailabilityZone** (*string*) – **[REQUIRED]**

The Availability Zone in which to create the volume. Use `DescribeAvailabilityZones` to list the Availability Zones that are currently available to you.

- **VolumeType** (*string*) – The volume type. This can be `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, or `standard` for Magnetic volumes.

Default: `standard`

- **Iops** (*integer*) – Only valid for Provisioned IOPS (SSD) volumes. The number of I/O operations per second (IOPS) to provision for the volume, with a maximum ratio of 30 IOPS/GiB.

Constraint: Range is 100 to 20000 for Provisioned IOPS (SSD) volumes

- **Encrypted** (*boolean*) – Specifies whether the volume should be encrypted. Encrypted Amazon EBS volumes may only be attached to instances that support Amazon EBS encryption. Volumes that are created from encrypted snapshots are automatically encrypted. There is no way to create an encrypted volume from an unencrypted snapshot or vice versa. If your AMI uses encrypted volumes, you can only launch it on supported instance types. For more information, see [Amazon EBS Encryption](#) in the *Amazon Elastic Compute Cloud User Guide*.
- **KmsKeyId** (*string*) – The full ARN of the AWS Key Management Service (AWS KMS) customer master key (CMK) to use when creating the encrypted volume. This parameter is only required if you want to use a non-default CMK; if this

parameter is not specified, the default CMK for EBS is used. The ARN contains the `arn:aws:kms` namespace, followed by the region of the CMK, the AWS account ID of the CMK owner, the `key` namespace, and then the CMK ID. For example, `arn:aws:kms:us-east-1:012345678910:key/abcd1234-a123-456a-a12b-a123b4cd56ef`. If a `KmsKeyId` is specified, the `Encrypted` flag must also be set.

Return type `ec2.Volume`

Returns Volume resource

create_vpc (***kwargs*)

Creates a VPC with the specified CIDR block.

The smallest VPC you can create uses a /28 netmask (16 IP addresses), and the largest uses a /16 netmask (65,536 IP addresses). To help you decide how big to make your VPC, see [Your VPC and Subnets](#) in the *Amazon Virtual Private Cloud User Guide*.

By default, each instance you launch in the VPC has the default DHCP options, which includes only a default DNS server that we provide (AmazonProvidedDNS). For more information about DHCP options, see [DHCP Options Sets](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
vpc = ec2.create_vpc(
    DryRun=True|False,
    CidrBlock='string',
    InstanceTenancy='default'|'dedicated'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **CidrBlock** (*string*) – **[REQUIRED]**

The network range for the VPC, in CIDR notation. For example, `10.0.0.0/16`.
- **InstanceTenancy** (*string*) – The supported tenancy options for instances launched into the VPC. A value of `default` means that instances can be launched with any tenancy; a value of `dedicated` means all instances launched into the VPC are launched as dedicated tenancy instances regardless of the tenancy assigned to the instance at launch. Dedicated tenancy instances run on single-tenant hardware.

Default: `default`

Return type `ec2.Vpc`

Returns Vpc resource

create_vpc_peering_connection (***kwargs*)

Requests a VPC peering connection between two VPCs: a requester VPC that you own and a peer VPC with which to create the connection. The peer VPC can belong to another AWS account. The requester VPC and peer VPC cannot have overlapping CIDR blocks.

The owner of the peer VPC must accept the peering request to activate the peering connection. The VPC peering connection request expires after 7 days, after which it cannot be accepted or rejected.

A `CreateVpcPeeringConnection` request between VPCs with overlapping CIDR blocks results in the VPC peering connection having a status of `failed`.

Request Syntax

```

vpc_peering_connection = ec2.create_vpc_peering_connection(
    DryRun=True|False,
    VpcId='string',
    PeerVpcId='string',
    PeerOwnerId='string'
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcId** (*string*) – The ID of the requester VPC.
- **PeerVpcId** (*string*) – The ID of the VPC with which you are creating the VPC peering connection.
- **PeerOwnerId** (*string*) – The AWS account ID of the owner of the peer VPC.

Default: Your AWS account ID

Return type `ec2.VpcPeeringConnection`

Returns `VpcPeeringConnection` resource

`disassociate_route_table` (**kwargs)

Disassociates a subnet from a route table.

After you perform this action, the subnet no longer uses the routes in the route table. Instead, it uses the routes in the VPC's main route table. For more information about route tables, see [Route Tables](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```

response = ec2.disassociate_route_table(
    DryRun=True|False,
    AssociationId='string'
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **AssociationId** (*string*) – **[REQUIRED]**

The association ID representing the current association between the route table and subnet.

Returns `None`

`import_key_pair` (**kwargs)

Imports the public key from an RSA key pair that you created with a third-party tool. Compare this with `CreateKeyPair`, in which AWS creates the key pair and gives the keys to you (AWS keeps a copy of the public key). With `ImportKeyPair`, you create the key pair and give AWS just the public key. The private key is never transferred between you and AWS.

For more information about key pairs, see [Key Pairs](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
key_pair = ec2.import_key_pair(
    DryRun=True|False,
    KeyName='string',
    PublicKeyMaterial=b'bytes'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **KeyName** (*string*) – **[REQUIRED]**
A unique name for the key pair.
- **PublicKeyMaterial** (*bytes*) – **[REQUIRED]**
The public key. You must base64 encode the public key material before sending it to AWS.

Return type `ec2.KeyPair`

Returns KeyPair resource

register_image (***kwargs*)

Registers an AMI. When you're creating an AMI, this is the final step you must complete before you can launch an instance from the AMI. For more information about creating AMIs, see [Creating Your Own AMIs](#) in the *Amazon Elastic Compute Cloud User Guide*.

Note: For Amazon EBS-backed instances, `CreateImage` creates and registers the AMI in a single request, so you don't have to register the AMI yourself.

You can also use `RegisterImage` to create an Amazon EBS-backed Linux AMI from a snapshot of a root device volume. For more information, see [Launching an Instance from a Snapshot](#) in the *Amazon Elastic Compute Cloud User Guide*.

Warning: Some Linux distributions, such as Red Hat Enterprise Linux (RHEL) and SUSE Linux Enterprise Server (SLES), use the EC2 `billingProduct` code associated with an AMI to verify subscription status for package updates. Creating an AMI from an EBS snapshot does not maintain this billing code, and subsequent instances launched from such an AMI will not be able to connect to package update infrastructure.

Similarly, although you can create a Windows AMI from a snapshot, you can't successfully launch an instance from the AMI.

To create Windows AMIs or to create AMIs for Linux operating systems that must retain AMI billing codes to work properly, see `CreateImage`.

If needed, you can deregister an AMI at any time. Any modifications you make to an AMI backed by an instance store volume invalidates its registration. If you make changes to an image, deregister the previous image and register the new image.

Note: You can't register an image where a secondary (non-root) snapshot has AWS Marketplace product codes.

Request Syntax

```
image = ec2.register_image(
    DryRun=True|False,
```

```

ImageLocation='string',
Name='string',
Description='string',
Architecture='i386'|'x86_64',
KernelId='string',
RamdiskId='string',
RootDeviceName='string',
BlockDeviceMappings=[
    {
        'VirtualName': 'string',
        'DeviceName': 'string',
        'Ebs': {
            'SnapshotId': 'string',
            'VolumeSize': 123,
            'DeleteOnTermination': True|False,
            'VolumeType': 'standard'|'io1'|'gp2',
            'Iops': 123,
            'Encrypted': True|False
        },
        'NoDevice': 'string'
    },
],
VirtualizationType='string',
SriovNetSupport='string'
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **ImageLocation** (*string*) – The full path to your AMI manifest in Amazon S3 storage.
- **Name** (*string*) – **[REQUIRED]**

A name for your AMI.

Constraints: 3-128 alphanumeric characters, parentheses (()), square brackets ([]), spaces (), periods (.), slashes (/), dashes (-), single quotes ('), at-signs (@), or underscores(_)

- **Description** (*string*) – A description for your AMI.
- **Architecture** (*string*) – The architecture of the AMI.

Default: For Amazon EBS-backed AMIs, `i386`. For instance store-backed AMIs, the architecture specified in the manifest file.

- **KernelId** (*string*) – The ID of the kernel.
- **RamdiskId** (*string*) – The ID of the RAM disk.
- **RootDeviceName** (*string*) – The name of the root device (for example, `/dev/sda1`, or `/dev/xvda`).
- **BlockDeviceMappings** (*list*) – One or more block device mapping entries.

– (*dict*) –

Describes a block device mapping.

* **VirtualName** (*string*) –

The virtual device name (ephemeral N). Instance store volumes are numbered starting from 0. An instance type with 2 available instance store volumes can specify mappings for `ephemeral0` and

`ephemeral1`. The number of available instance store volumes depends on the instance type. After you connect to the instance, you must mount the volume.

Constraints: For M3 instances, you must specify instance store volumes in the block device mapping for the instance. When you launch an M3 instance, we ignore any instance store volumes specified in the block device mapping for the AMI.

* **DeviceName** (*string*) –

The device name exposed to the instance (for example, `/dev/sdh` or `xvdh`).

* **Ebs** (*dict*) –

Parameters used to automatically set up EBS volumes when the instance is launched.

· **SnapshotId** (*string*) –

The ID of the snapshot.

· **VolumeSize** (*integer*) –

The size of the volume, in GiB.

Constraints: 1–1024 for standard volumes, 1–16384 for gp2 volumes, and 4–16384 for io1 volumes. If you specify a snapshot, the volume size must be equal to or larger than the snapshot size.

Default: If you're creating the volume from a snapshot and don't specify a volume size, the default is the snapshot size.

· **DeleteOnTermination** (*boolean*) –

Indicates whether the EBS volume is deleted on instance termination.

· **VolumeType** (*string*) –

The volume type. `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, and `standard` for Magnetic volumes.

Default: `standard`

· **Iops** (*integer*) –

The number of I/O operations per second (IOPS) that the volume supports. For Provisioned IOPS (SSD) volumes, this represents the number of IOPS that are provisioned for the volume. For General Purpose (SSD) volumes, this represents the baseline performance of the volume and the rate at which the volume accumulates I/O credits for bursting. For more information on General Purpose (SSD) baseline performance, I/O credits, and bursting, see [Amazon EBS Volume Types](#) in the *Amazon Elastic Compute Cloud User Guide*.

Constraint: Range is 100 to 20000 for Provisioned IOPS (SSD) volumes and 3 to 10000 for General Purpose (SSD) volumes.

Condition: This parameter is required for requests to create `io1` volumes; it is not used in requests to create `standard` or `gp2` volumes.

- **Encrypted** (*boolean*) –

Indicates whether the EBS volume is encrypted. Encrypted Amazon EBS volumes may only be attached to instances that support Amazon EBS encryption.

- * **NoDevice** (*string*) –

Suppresses the specified device included in the block device mapping of the AMI.

- **VirtualizationType** (*string*) – The type of virtualization.

Default: `paravirtual`

- **SriovNetSupport** (*string*) – Set to `simple` to enable enhanced networking for the AMI and any instances that you launch from the AMI.

There is no way to disable enhanced networking at this time.

This option is supported only for HVM AMIs. Specifying this option with a PV AMI can make instances launched from the AMI unreachable.

Return type `ec2.Image`

Returns Image resource

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

DhcpOptions (*id*)

Creates a DhcpOptions resource.:

```
dhcp_options = ec2.DhcpOptions('id')
```

Parameters **id** (*string*) – The DhcpOptions's id identifier. This **must** be set.

Return type `EC2.DhcpOptions`

Returns A DhcpOptions resource

Image (*id*)

Creates a Image resource.:

```
image = ec2.Image('id')
```

Parameters **id** (*string*) – The Image's id identifier. This **must** be set.

Return type `EC2.Image`

Returns A Image resource

Instance (*id*)

Creates a Instance resource.:

```
instance = ec2.Instance('id')
```

Parameters **id** (*string*) – The Instance's id identifier. This **must** be set.

Return type `EC2.Instance`

Returns A Instance resource

InternetGateway (*id*)

Creates a InternetGateway resource.:

```
internet_gateway = ec2.InternetGateway('id')
```

Parameters **id** (*string*) – The InternetGateway's id identifier. This **must** be set.

Return type `EC2.InternetGateway`

Returns A InternetGateway resource

KeyPair (*name*)

Creates a KeyPair resource.:

```
key_pair = ec2.KeyPair('name')
```

Parameters *name* (*string*) – The KeyPair’s name identifier. This **must** be set.

Return type `EC2.KeyPair`

Returns A KeyPair resource

NetworkAcl (*id*)

Creates a NetworkAcl resource.:

```
network_acl = ec2.NetworkAcl('id')
```

Parameters *id* (*string*) – The NetworkAcl’s id identifier. This **must** be set.

Return type `EC2.NetworkAcl`

Returns A NetworkAcl resource

NetworkInterface (*id*)

Creates a NetworkInterface resource.:

```
network_interface = ec2.NetworkInterface('id')
```

Parameters *id* (*string*) – The NetworkInterface’s id identifier. This **must** be set.

Return type `EC2.NetworkInterface`

Returns A NetworkInterface resource

PlacementGroup (*name*)

Creates a PlacementGroup resource.:

```
placement_group = ec2.PlacementGroup('name')
```

Parameters *name* (*string*) – The PlacementGroup’s name identifier. This **must** be set.

Return type `EC2.PlacementGroup`

Returns A PlacementGroup resource

RouteTable (*id*)

Creates a RouteTable resource.:

```
route_table = ec2.RouteTable('id')
```

Parameters *id* (*string*) – The RouteTable’s id identifier. This **must** be set.

Return type `EC2.RouteTable`

Returns A RouteTable resource

RouteTableAssociation (*id*)

Creates a RouteTableAssociation resource.:

```
route_table_association = ec2.RouteTableAssociation('id')
```

Parameters *id* (*string*) – The RouteTableAssociation’s id identifier. This **must** be set.

Return type `EC2.RouteTableAssociation`

Returns A RouteTableAssociation resource

SecurityGroup (*id*)

Creates a SecurityGroup resource.:

```
security_group = ec2.SecurityGroup('id')
```

Parameters *id* (*string*) – The SecurityGroup’s id identifier. This **must** be set.

Return type `EC2.SecurityGroup`

Returns A SecurityGroup resource

Snapshot (*id*)

Creates a Snapshot resource.:

```
snapshot = ec2.Snapshot('id')
```

Parameters *id* (*string*) – The Snapshot’s id identifier. This **must** be set.

Return type `EC2.Snapshot`

Returns A Snapshot resource

Subnet (*id*)

Creates a Subnet resource.:

```
subnet = ec2.Subnet('id')
```

Parameters *id* (*string*) – The Subnet’s id identifier. This **must** be set.

Return type `EC2.Subnet`

Returns A Subnet resource

Tag (*resource_id*, *key*, *value*)

Creates a Tag resource.:

```
tag = ec2.Tag('resource_id', 'key', 'value')
```

Parameters

- **resource_id** (*string*) – The Tag’s resource_id identifier. This **must** be set.
- **key** (*string*) – The Tag’s key identifier. This **must** be set.
- **value** (*string*) – The Tag’s value identifier. This **must** be set.

Return type `EC2.Tag`

Returns A Tag resource

Volume (*id*)

Creates a Volume resource.:

```
volume = ec2.Volume('id')
```

Parameters *id* (*string*) – The Volume’s id identifier. This **must** be set.

Return type `EC2.Volume`

Returns A Volume resource

Vpc (*id*)

Creates a Vpc resource.:

```
vpc = ec2.Vpc('id')
```

Parameters *id* (*string*) – The Vpc’s id identifier. This **must** be set.

Return type `EC2.Vpc`

Returns A Vpc resource

VpcPeeringConnection (*id*)

Creates a VpcPeeringConnection resource.:

```
vpc_peering_connection = ec2.VpcPeeringConnection('id')
```

Parameters *id* (*string*) – The VpcPeeringConnection’s id identifier. This **must** be set.

Return type `EC2.VpcPeeringConnection`

Returns A VpcPeeringConnection resource

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

dhcp_options_sets**all** ()

Creates an iterable of all DhcpOptions resources in the collection.

Request Syntax

```
dhcp_options_iterator = ec2.dhcp_options_sets.all()
```

Return type `list(ec2.DhcpOptions)`

Returns A list of DhcpOptions resources

filter (**kwargs)

Creates an iterable of all DhcpOptions resources in the collection filtered by kwargs passed to method.

Request Syntax

```
dhcp_options_iterator = ec2.dhcp_options_sets.filter(
    DryRun=True|False,
    DhcpOptionsIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **DhcpOptionsIds** (*list*) – The IDs of one or more DHCP options sets.

Default: Describes all your DHCP options sets.

– (*string*) –

- **Filters** (*list*) – One or more filters.
 - `dhcp-options-id` - The ID of a set of DHCP options.

- `key` - The key for one of the options (for example, `domain-name`).
- `value` - The value for one of the options.
- `tag :key=*value*` - The key/value combination of a tag assigned to the resource.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag’s value is), and the tag value `X` (regardless of what the tag’s key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key=*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

Return type `list(ec2.DhcpOptions)`

Returns A list of `DhcpOptions` resources

limit (***kwargs*)

Creates an iterable up to a specified amount of `DhcpOptions` resources in the collection.

Request Syntax

```
dhcp_options_iterator = ec2.dhcp_options_sets.limit(
    count=123
)
```

Parameters `count` (*integer*) - The limit to the number of resources in the iterable.

Return type `list(ec2.DhcpOptions)`

Returns A list of `DhcpOptions` resources

page_size (***kwargs*)

Creates an iterable of all `DhcpOptions` resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
dhcp_options_iterator = ec2.dhcp_options_sets.page_size(
    count=123
)
```

Parameters `count` (*integer*) - The number of items returned by each service call

Return type `list(ec2.DhcpOptions)`

Returns A list of `DhcpOptions` resources

images

all()

Creates an iterable of all Image resources in the collection.

Request Syntax

```
image_iterator = ec2.images.all()
```

Return type list(ec2.Image)

Returns A list of Image resources

filter(kwargs)**

Creates an iterable of all Image resources in the collection filtered by kwargs passed to method.

Request Syntax

```
image_iterator = ec2.images.filter(
    DryRun=True|False,
    ImageIds=[
        'string',
    ],
    Owners=[
        'string',
    ],
    ExecutableUsers=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **ImageIds** (*list*) – One or more image IDs.

Default: Describes all images available to you.

– (*string*) –

- **Owners** (*list*) – Filters the images by the owner. Specify an AWS account ID, `amazon` (owner is Amazon), `aws-marketplace` (owner is AWS Marketplace), `self` (owner is the sender of the request). Omitting this option returns all images for which you have launch permissions, regardless of ownership.
– (*string*) –
- **ExecutableUsers** (*list*) – Scopes the images by users with explicit launch permissions. Specify an AWS account ID, `self` (the sender of the request), or `all` (public AMIs).
– (*string*) –
- **Filters** (*list*) – One or more filters.
– `architecture` - The image architecture (`i386` | `x86_64`).

- `block-device-mapping.delete-on-termination` - A Boolean value that indicates whether the Amazon EBS volume is deleted on instance termination.
- `block-device-mapping.device-name` - The device name for the EBS volume (for example, `/dev/sdh`).
- `block-device-mapping.snapshot-id` - The ID of the snapshot used for the EBS volume.
- `block-device-mapping.volume-size` - The volume size of the EBS volume, in GiB.
- `block-device-mapping.volume-type` - The volume type of the EBS volume (`gp2` | `standard` | `io1`).
- `description` - The description of the image (provided during image creation).
- `hypervisor` - The hypervisor type (`ovm` | `xen`).
- `image-id` - The ID of the image.
- `image-type` - The image type (`machine` | `kernel` | `ramdisk`).
- `is-public` - A Boolean that indicates whether the image is public.
- `kernel-id` - The kernel ID.
- `manifest-location` - The location of the image manifest.
- `name` - The name of the AMI (provided during image creation).
- `owner-alias` - The AWS account alias (for example, `amazon`).
- `owner-id` - The AWS account ID of the image owner.
- `platform` - The platform. To only list Windows-based AMIs, use `windows`.
- `product-code` - The product code.
- `product-code.type` - The type of the product code (`devpay` | `marketplace`).
- `ramdisk-id` - The RAM disk ID.
- `root-device-name` - The name of the root device volume (for example, `/dev/sda1`).
- `root-device-type` - The type of the root device volume (`ebs` | `instance-store`).
- `state` - The state of the image (`available` | `pending` | `failed`).
- `state-reason-code` - The reason code for the state change.
- `state-reason-message` - The message for the state change.
- `tag :key=*value*` - The key/value combination of a tag assigned to the resource.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the tag-value filter. For example, if you use both the filter “tag-key=Purpose” and the filter “tag-value=X”, you get any resources assigned both the tag key Purpose (regardless of what the tag’s value is), and the tag value X (regardless of what the tag’s key is). If you want to list only resources where Purpose is X, see the `tag :key=*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the tag-key filter.
- `virtualization-type` - The virtualization type (`paravirtual` | `hvm`).
- *(dict)* -

A filter name and value pair that is used to return a more specific list

of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

- * **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

- * **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

- (*string*) –

Return type `list(ec2.Image)`

Returns A list of Image resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Image resources in the collection.

Request Syntax

```
image_iterator = ec2.images.limit(  
    count=123  
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(ec2.Image)`

Returns A list of Image resources

page_size (***kwargs*)

Creates an iterable of all Image resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
image_iterator = ec2.images.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(ec2.Image)`

Returns A list of Image resources

instances

all ()

Creates an iterable of all Instance resources in the collection.

Request Syntax

```
instance_iterator = ec2.instances.all()
```

Return type `list(ec2.Instance)`

Returns A list of Instance resources

create_tags (***kwargs*)

Adds or overwrites one or more tags for the specified Amazon EC2 resource or resources. Each resource can have a maximum of 10 tags. Each tag consists of a key and optional value. Tag keys must be unique per resource.

For more information about tags, see [Tagging Your Resources](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax


```

response = ec2.instances.create_tags(
    DryRun=True|False,
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Tags** (*list*) – **[REQUIRED]**

One or more tags. The `value` parameter is required, but if you don't want the tag to have a value, specify the parameter with no value, and we set the value to an empty string.

– (*dict*) –

Describes a tag.

* **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

* **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

Returns None

filter (***kwargs*)

Creates an iterable of all Instance resources in the collection filtered by kwargs passed to method.

Request Syntax

```

instance_iterator = ec2.instances.filter(
    DryRun=True|False,
    InstanceIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    NextToken='string',
    MaxResults=123
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceIds** (*list*) – One or more instance IDs.

Default: Describes all your instances.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `architecture` - The instance architecture (`i386` | `x86_64`).
 - `availability-zone` - The Availability Zone of the instance.
 - `block-device-mapping.attach-time` - The attach time for an EBS volume mapped to the instance, for example, `2010-09-15T17:15:20.000Z`.
 - `block-device-mapping.delete-on-termination` - A Boolean that indicates whether the EBS volume is deleted on instance termination.
 - `block-device-mapping.device-name` - The device name for the EBS volume (for example, `/dev/sdh` or `xvdh`).
 - `block-device-mapping.status` - The status for the EBS volume (`attaching` | `attached` | `detaching` | `detached`).
 - `block-device-mapping.volume-id` - The volume ID of the EBS volume.
 - `client-token` - The idempotency token you provided when you launched the instance.
 - `dns-name` - The public DNS name of the instance.
 - `group-id` - The ID of the security group for the instance. EC2-Classic only.
 - `group-name` - The name of the security group for the instance. EC2-Classic only.
 - `hypervisor` - The hypervisor type of the instance (`ovm` | `xen`).
 - `iam-instance-profile.arn` - The instance profile associated with the instance. Specified as an ARN.
 - `image-id` - The ID of the image used to launch the instance.
 - `instance-id` - The ID of the instance.
 - `instance-lifecycle` - Indicates whether this is a Spot Instance (`spot`).
 - `instance-state-code` - The state of the instance, as a 16-bit unsigned integer. The high byte is an opaque internal value and should be ignored. The low byte is set based on the state represented. The valid values are: 0 (pending), 16 (running), 32 (shutting-down), 48 (terminated), 64 (stopping), and 80 (stopped).
 - `instance-state-name` - The state of the instance (`pending` | `running` | `shutting-down` | `terminated` | `stopping` | `stopped`).
 - `instance-type` - The type of instance (for example, `t2.micro`).
 - `instance.group-id` - The ID of the security group for the instance.
 - `instance.group-name` - The name of the security group for the instance.
 - `ip-address` - The public IP address of the instance.
 - `kernel-id` - The kernel ID.

- `key-name` - The name of the key pair used when the instance was launched.
- `launch-index` - When launching multiple instances, this is the index for the instance in the launch group (for example, 0, 1, 2, and so on).
- `launch-time` - The time when the instance was launched.
- `monitoring-state` - Indicates whether monitoring is enabled for the instance (`disabled` | `enabled`).
- `owner-id` - The AWS account ID of the instance owner.
- `placement-group-name` - The name of the placement group for the instance.
- `platform` - The platform. Use windows if you have Windows instances; otherwise, leave blank.
- `private-dns-name` - The private DNS name of the instance.
- `private-ip-address` - The private IP address of the instance.
- `product-code` - The product code associated with the AMI used to launch the instance.
- `product-code.type` - The type of product code (`devpay` | `marketplace`).
- `ramdisk-id` - The RAM disk ID.
- `reason` - The reason for the current state of the instance (for example, shows “User Initiated [date]” when you stop or terminate the instance). Similar to the `state-reason-code` filter.
- `requester-id` - The ID of the entity that launched the instance on your behalf (for example, AWS Management Console, Auto Scaling, and so on).
- `reservation-id` - The ID of the instance’s reservation. A reservation ID is created any time you launch an instance. A reservation ID has a one-to-one relationship with an instance launch request, but can be associated with more than one instance if you launch multiple instances using the same launch request. For example, if you launch one instance, you’ll get one reservation ID. If you launch ten instances using the same launch request, you’ll also get one reservation ID.
- `root-device-name` - The name of the root device for the instance (for example, `/dev/sda1` or `/dev/xvda`).
- `root-device-type` - The type of root device that the instance uses (`ebs` | `instance-store`).
- `source-dest-check` - Indicates whether the instance performs source/destination checking. A value of `true` means that checking is enabled, and `false` means checking is disabled. The value must be `false` for the instance to perform network address translation (NAT) in your VPC.
- `spot-instance-request-id` - The ID of the Spot Instance request.
- `state-reason-code` - The reason code for the state change.
- `state-reason-message` - A message that describes the state change.
- `subnet-id` - The ID of the subnet for the instance.
- `tag :key=*value*` - The key/value combination of a tag assigned to the resource, where `tag :key` is the tag’s key.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you

get any resources assigned both the tag key Purpose (regardless of what the tag's value is), and the tag value X (regardless of what the tag's key is). If you want to list only resources where Purpose is X, see the tag `:key=*value*` filter.

- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `tenancy` - The tenancy of an instance (`dedicated` | `default`).
- `virtualization-type` - The virtualization type of the instance (`paravirtual` | `hvm`).
- `vpc-id` - The ID of the VPC that the instance is running in.
- `network-interface.description` - The description of the network interface.
- `network-interface.subnet-id` - The ID of the subnet for the network interface.
- `network-interface.vpc-id` - The ID of the VPC for the network interface.
- `network-interface.network-interface.id` - The ID of the network interface.
- `network-interface.owner-id` - The ID of the owner of the network interface.
- `network-interface.availability-zone` - The Availability Zone for the network interface.
- `network-interface.requester-id` - The requester ID for the network interface.
- `network-interface.requester-managed` - Indicates whether the network interface is being managed by AWS.
- `network-interface.status` - The status of the network interface (`available` | `in-use`).
- `network-interface.mac-address` - The MAC address of the network interface.
- `network-interface-private-dns-name` - The private DNS name of the network interface.
- `network-interface.source-dest-check` - Whether the network interface performs source/destination checking. A value of `true` means checking is enabled, and `false` means checking is disabled. The value must be `false` for the network interface to perform network address translation (NAT) in your VPC.
- `network-interface.group-id` - The ID of a security group associated with the network interface.
- `network-interface.group-name` - The name of a security group associated with the network interface.
- `network-interface.attachment.attachment-id` - The ID of the interface attachment.
- `network-interface.attachment.instance-id` - The ID of the instance to which the network interface is attached.
- `network-interface.attachment.instance-owner-id` - The owner ID of the instance to which the network interface is attached.
- `network-interface.addresses.private-ip-address` - The private IP address associated with the network interface.
- `network-interface.attachment.device-index` - The device index to which the network interface is attached.
- `network-interface.attachment.status` - The status

- of the attachment (attaching | attached | detaching | detached).
- `network-interface.attachment.attach-time` - The time that the network interface was attached to an instance.
- `network-interface.attachment.delete-on-termination` - Specifies whether the attachment is deleted when an instance is terminated.
- `network-interface.addresses.primary` - Specifies whether the IP address of the network interface is the primary private IP address.
- `network-interface.addresses.association.public-ip` - The ID of the association of an Elastic IP address with a network interface.
- `network-interface.addresses.association.ip-owner-id` - The owner ID of the private IP address associated with the network interface.
- `association.public-ip` - The address of the Elastic IP address bound to the network interface.
- `association.ip-owner-id` - The owner of the Elastic IP address associated with the network interface.
- `association.allocation-id` - The allocation ID returned when you allocated the Elastic IP address for your network interface.
- `association.association-id` - The association ID returned when the network interface was associated with an IP address.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

- **NextToken** (*string*) - The token to request the next page of results.
- **MaxResults** (*integer*) - The maximum number of results to return for the request in a single page. The remaining results of the initial request can be seen by sending another request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. You cannot specify this parameter and the `instance IDs` parameter in the same request.

Return type `list(ec2.Instance)`

Returns A list of Instance resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Instance resources in the collection.

Request Syntax

```
instance_iterator = ec2.instances.limit(
    count=123
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(ec2.Instance)`

Returns A list of Instance resources

monitor (***kwargs*)

Enables monitoring for a running instance. For more information about monitoring instances, see [Monitoring Your Instances and Volumes](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = ec2.instances.monitor(  
    DryRun=True|False,  
  
)
```

Parameters `DryRun` (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type `dict`

Returns

Response Syntax

```
{  
    'InstanceMonitorings': [  
        {  
            'InstanceId': 'string',  
            'Monitoring': {  
                'State': 'disabled'|'disabling'|'enabled'|'pending'  
            }  
        },  
    ]  
}
```

Response Structure

- (*dict*) –

- **InstanceMonitorings** (*list*) –

Monitoring information for one or more instances.

- * (*dict*) –

Describes the monitoring information of the instance.

- **InstanceId** (*string*) –

The ID of the instance.

- **Monitoring** (*dict*) –

The monitoring information.

- **State** (*string*) –

Indicates whether monitoring is enabled for the instance.

page_size (***kwargs*)

Creates an iterable of all Instance resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
instance_iterator = ec2.instances.page_size(
    count=123
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(ec2.Instance)`

Returns A list of Instance resources

reboot (***kwargs*)

Requests a reboot of one or more instances. This operation is asynchronous; it only queues a request to reboot the specified instances. The operation succeeds if the instances are valid and belong to you. Requests to reboot terminated instances are ignored.

If a Linux/Unix instance does not cleanly shut down within four minutes, Amazon EC2 performs a hard reboot.

For more information about troubleshooting, see [Getting Console Output and Rebooting Instances](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = ec2.instances.reboot(
    DryRun=True|False,
)
```

Parameters `DryRun` (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns None

start (***kwargs*)

Starts an Amazon EBS-backed AMI that you've previously stopped.

Instances that use Amazon EBS volumes as their root devices can be quickly stopped and started. When an instance is stopped, the compute resources are released and you are not billed for hourly instance usage. However, your root partition Amazon EBS volume remains, continues to persist your data, and you are charged for Amazon EBS volume usage. You can restart your instance at any time. Each time you transition an instance from stopped to started, Amazon EC2 charges a full instance hour, even if transitions happen multiple times within a single hour.

Before stopping an instance, make sure it is in a state from which it can be restarted. Stopping an instance does not preserve data stored in RAM.

Performing this operation on an instance that uses an instance store as its root device returns an error.

For more information, see [Stopping Instances](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = ec2.instances.start(
    AdditionalInfo='string',
    DryRun=True|False
)
```

Parameters

- **AdditionalInfo** (*string*) – Reserved.

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns

Response Syntax

```
{
  'StartingInstances': [
    {
      'InstanceId': 'string',
      'CurrentState': {
        'Code': 123,
        'Name': 'pending' | 'running' | 'shutting-down' | 'terminated' | 'stop
      },
      'PreviousState': {
        'Code': 123,
        'Name': 'pending' | 'running' | 'shutting-down' | 'terminated' | 'stop
      }
    },
  ],
}
```

Response Structure

- (*dict*) –
 - **StartingInstances** (*list*) –

Information about one or more started instances.

* (*dict*) –

Describes an instance state change.

- **InstanceId** (*string*) –
The ID of the instance.
- **CurrentState** (*dict*) –
The current state of the instance.
- **Code** (*integer*) –
The low byte represents the state. The high byte is an opaque internal value and should be ignored.
 - 0: pending
 - 16: running
 - 32: shutting-down
 - 48: terminated
 - 64: stopping
 - 80: stopped
- **Name** (*string*) –
The current state of the instance.
- **PreviousState** (*dict*) –
The previous state of the instance.
- **Code** (*integer*) –
The low byte represents the state. The high byte is an opaque internal value and should be ignored.
 - 0: pending

- 16: running
- 32: shutting-down
- 48: terminated
- 64: stopping
- 80: stopped
- **Name** (*string*) –

The current state of the instance.

stop (***kwargs*)

Stops an Amazon EBS-backed instance. Each time you transition an instance from stopped to started, Amazon EC2 charges a full instance hour, even if transitions happen multiple times within a single hour.

You can't start or stop Spot Instances.

Instances that use Amazon EBS volumes as their root devices can be quickly stopped and started. When an instance is stopped, the compute resources are released and you are not billed for hourly instance usage. However, your root partition Amazon EBS volume remains, continues to persist your data, and you are charged for Amazon EBS volume usage. You can restart your instance at any time.

Before stopping an instance, make sure it is in a state from which it can be restarted. Stopping an instance does not preserve data stored in RAM.

Performing this operation on an instance that uses an instance store as its root device returns an error.

You can stop, start, and terminate EBS-backed instances. You can only terminate instance store-backed instances. What happens to an instance differs if you stop it or terminate it. For example, when you stop an instance, the root device and any other devices attached to the instance persist. When you terminate an instance, the root device and any other devices attached during the instance launch are automatically deleted. For more information about the differences between stopping and terminating instances, see [Instance Lifecycle](#) in the *Amazon Elastic Compute Cloud User Guide*.

For more information about troubleshooting, see [Troubleshooting Stopping Your Instance](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = ec2.instances.stop(
    DryRun=True|False,
    Force=True|False
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Force** (*boolean*) – Forces the instances to stop. The instances do not have an opportunity to flush file system caches or file system metadata. If you use this option, you must perform file system check and repair procedures. This option is not recommended for Windows instances.

Default: false

Return type dict

Returns

Response Syntax

```
{
  'StoppingInstances': [
    {
      'InstanceId': 'string',
      'CurrentState': {
        'Code': 123,
        'Name': 'pending' | 'running' | 'shutting-down' | 'terminated' | 'stop
      },
      'PreviousState': {
        'Code': 123,
        'Name': 'pending' | 'running' | 'shutting-down' | 'terminated' | 'stop
      }
    },
  ],
}
```

Response Structure

- (*dict*) –
 - **StoppingInstances** (*list*) –

Information about one or more stopped instances.

* (*dict*) –

Describes an instance state change.

- **InstanceId** (*string*) –

The ID of the instance.

- **CurrentState** (*dict*) –

The current state of the instance.

- **Code** (*integer*) –

The low byte represents the state. The high byte is an opaque internal value and should be ignored.

- 0: pending
- 16: running
- 32: shutting-down
- 48: terminated
- 64: stopping
- 80: stopped
- **Name** (*string*) –

The current state of the instance.

- **PreviousState** (*dict*) –

The previous state of the instance.

- **Code** (*integer*) –

The low byte represents the state. The high byte is an opaque internal value and should be ignored.

- 0: pending
- 16: running
- 32: shutting-down
- 48: terminated
- 64: stopping
- 80: stopped
- **Name** (*string*) –

The current state of the instance.

terminate (**kwargs)

Shuts down one or more instances. This operation is idempotent; if you terminate an instance more than once, each call succeeds.

Terminated instances remain visible after termination (for approximately one hour).

By default, Amazon EC2 deletes all EBS volumes that were attached when the instance launched. Volumes attached after instance launch continue running.

You can stop, start, and terminate EBS-backed instances. You can only terminate instance store-backed instances. What happens to an instance differs if you stop it or terminate it. For example, when you stop an instance, the root device and any other devices attached to the instance persist. When you terminate an instance, any attached EBS volumes with the `DeleteOnTermination` block device mapping parameter set to `true` are automatically deleted. For more information about the differences between stopping and terminating instances, see [Instance Lifecycle](#) in the *Amazon Elastic Compute Cloud User Guide*.

For more information about troubleshooting, see [Troubleshooting Terminating Your Instance](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = ec2.instances.terminate(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns**Response Syntax**

```
{
    'TerminatingInstances': [
        {
            'InstanceId': 'string',
            'CurrentState': {
                'Code': 123,
                'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stopped'|'stopping',
            },
            'PreviousState': {
                'Code': 123,
                'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stopped'|'stopping',
            },
        },
    ],
}
```

Response Structure

- (*dict*) –
 - **TerminatingInstances** (*list*) –

Information about one or more terminated instances.

- * (*dict*) –

Describes an instance state change.

- **InstanceId** (*string*) –
The ID of the instance.
- **CurrentState** (*dict*) –
The current state of the instance.
- **Code** (*integer*) –
The low byte represents the state. The high byte is an opaque internal value and should be ignored.
 - 0: pending
 - 16: running
 - 32: shutting-down
 - 48: terminated
 - 64: stopping
 - 80: stopped
- **Name** (*string*) –
The current state of the instance.
- **PreviousState** (*dict*) –
The previous state of the instance.
- **Code** (*integer*) –
The low byte represents the state. The high byte is an opaque internal value and should be ignored.
 - 0: pending
 - 16: running
 - 32: shutting-down
 - 48: terminated
 - 64: stopping
 - 80: stopped
- **Name** (*string*) –
The current state of the instance.

unmonitor (***kwargs*)

Disables monitoring for a running instance. For more information about monitoring instances, see [Monitoring Your Instances and Volumes](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = ec2.instances.unmonitor(  
    DryRun=True|False,  
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns

Response Syntax

```
{  
    'InstanceMonitorings': [  
        {
```

```

        'InstanceId': 'string',
        'Monitoring': {
            'State': 'disabled'|'disabling'|'enabled'|'pending'
        }
    },
]
}

```

Response Structure

- (*dict*) –

– **InstanceMonitorings** (*list*) –

Monitoring information for one or more instances.

- * (*dict*) –

Describes the monitoring information of the instance.

- **InstanceId** (*string*) –

The ID of the instance.

- **Monitoring** (*dict*) –

The monitoring information.

- **State** (*string*) –

Indicates whether monitoring is enabled for the instance.

internet_gateways

all()

Creates an iterable of all InternetGateway resources in the collection.

Request Syntax

```
internet_gateway_iterator = ec2.internet_gateways.all()
```

Return type list(ec2.InternetGateway)

Returns A list of InternetGateway resources

filter(**kwargs)

Creates an iterable of all InternetGateway resources in the collection filtered by kwargs passed to method.

Request Syntax

```

internet_gateway_iterator = ec2.internet_gateways.filter(
    DryRun=True|False,
    InternetGatewayIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InternetGatewayIds** (*list*) – One or more Internet gateway IDs.

Default: Describes all your Internet gateways.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `attachment.state` - The current state of the attachment between the gateway and the VPC (`available`). Present only if a VPC is attached.
 - `attachment.vpc-id` - The ID of an attached VPC.
 - `internet-gateway-id` - The ID of the Internet gateway.
 - `tag:key=*value*` - The key/value combination of a tag assigned to the resource.
 - `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “tag-key=Purpose” and the filter “tag-value=X”, you get any resources assigned both the tag key Purpose (regardless of what the tag’s value is), and the tag value X (regardless of what the tag’s key is). If you want to list only resources where Purpose is X, see the `tag:key=*value*` filter.
 - `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

Return type `list(ec2.InternetGateway)`

Returns A list of InternetGateway resources

limit (***kwargs*)

Creates an iterable up to a specified amount of InternetGateway resources in the collection.

Request Syntax

```
internet_gateway_iterator = ec2.internet_gateways.limit(  
    count=123  
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(ec2.InternetGateway)`

Returns A list of InternetGateway resources

page_size (***kwargs*)

Creates an iterable of all InternetGateway resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
internet_gateway_iterator = ec2.internet_gateways.page_size(
    count=123
)
```

Parameters *count* (*integer*) – The number of items returned by each service call

Return type `list(ec2.InternetGateway)`

Returns A list of InternetGateway resources

key_pairs

all()

Creates an iterable of all KeyPair resources in the collection.

Request Syntax

```
key_pair_iterator = ec2.key_pairs.all()
```

Return type `list(ec2.KeyPair)`

Returns A list of KeyPair resources

filter (***kwargs*)

Creates an iterable of all KeyPair resources in the collection filtered by kwargs passed to method.

Request Syntax

```
key_pair_iterator = ec2.key_pairs.filter(
    DryRun=True|False,
    KeyNames=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **KeyNames** (*list*) – One or more key pair names.

Default: Describes all your key pairs.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `fingerprint` - The fingerprint of the key pair.
 - `key-name` - The name of the key pair.
 - (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

Return type `list(ec2.KeyPair)`

Returns A list of KeyPair resources

limit (***kwargs*)

Creates an iterable up to a specified amount of KeyPair resources in the collection.

Request Syntax

```
key_pair_iterator = ec2.key_pairs.limit(  
    count=123  
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(ec2.KeyPair)`

Returns A list of KeyPair resources

page_size (***kwargs*)

Creates an iterable of all KeyPair resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
key_pair_iterator = ec2.key_pairs.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(ec2.KeyPair)`

Returns A list of KeyPair resources

network_acls

all ()

Creates an iterable of all NetworkAcl resources in the collection.

Request Syntax

```
network_acl_iterator = ec2.network_acls.all()
```

Return type `list(ec2.NetworkAcl)`

Returns A list of NetworkAcl resources

filter (***kwargs*)

Creates an iterable of all NetworkAcl resources in the collection filtered by kwargs passed to method.

Request Syntax


```

network_acl_iterator = ec2.network_acls.filter(
    DryRun=True|False,
    NetworkAclIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **NetworkAclIds** (*list*) – One or more network ACL IDs.

Default: Describes all your network ACLs.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `association.association-id` - The ID of an association ID for the ACL.
 - `association.network-acl-id` - The ID of the network ACL involved in the association.
 - `association.subnet-id` - The ID of the subnet involved in the association.
 - `default` - Indicates whether the ACL is the default network ACL for the VPC.
 - `entry.cidr` - The CIDR range specified in the entry.
 - `entry.egress` - Indicates whether the entry applies to egress traffic.
 - `entry.icmp.code` - The ICMP code specified in the entry, if any.
 - `entry.icmp.type` - The ICMP type specified in the entry, if any.
 - `entry.port-range.from` - The start of the port range specified in the entry.
 - `entry.port-range.to` - The end of the port range specified in the entry.
 - `entry.protocol` - The protocol specified in the entry (`tcp` | `udp` | `icmp` or a protocol number).
 - `entry.rule-action` - Allows or denies the matching traffic (`allow` | `deny`).
 - `entry.rule-number` - The number of an entry (in other words, rule) in the ACL's set of entries.
 - `network-acl-id` - The ID of the network ACL.
 - `tag :key=*value*` - The key/value combination of a tag assigned to the resource.
 - `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use

both the filter “tag-key=Purpose” and the filter “tag-value=X”, you get any resources assigned both the tag key Purpose (regardless of what the tag’s value is), and the tag value X (regardless of what the tag’s key is). If you want to list only resources where Purpose is X, see the `tag:key=*value*` filter.

- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `vpc-id` - The ID of the VPC for the network ACL.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

Return type `list(ec2.NetworkAcl)`

Returns A list of NetworkAcl resources

limit (***kwargs*)

Creates an iterable up to a specified amount of NetworkAcl resources in the collection.

Request Syntax

```
network_acl_iterator = ec2.network_acls.limit(  
    count=123  
)
```

Parameters `count` (*integer*) - The limit to the number of resources in the iterable.

Return type `list(ec2.NetworkAcl)`

Returns A list of NetworkAcl resources

page_size (***kwargs*)

Creates an iterable of all NetworkAcl resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
network_acl_iterator = ec2.network_acls.page_size(  
    count=123  
)
```

Parameters `count` (*integer*) - The number of items returned by each service call

Return type `list(ec2.NetworkAcl)`

Returns A list of NetworkAcl resources

network_interfaces

all ()

Creates an iterable of all NetworkInterface resources in the collection.

Request Syntax

```
network_interface_iterator = ec2.network_interfaces.all()
```

Return type `list(ec2.NetworkInterface)`

Returns A list of NetworkInterface resources

filter (***kwargs*)

Creates an iterable of all NetworkInterface resources in the collection filtered by kwargs passed to method.

Request Syntax

```
network_interface_iterator = ec2.network_interfaces.filter(
    DryRun=True|False,
    NetworkInterfaceIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **NetworkInterfaceIds** (*list*) – One or more network interface IDs.

Default: Describes all your network interfaces.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `addresses.private-ip-address` - The private IP addresses associated with the network interface.
 - `addresses.primary` - Whether the private IP address is the primary IP address associated with the network interface.
 - `addresses.association.public-ip` - The association ID returned when the network interface was associated with the Elastic IP address.
 - `addresses.association.owner-id` - The owner ID of the addresses associated with the network interface.
 - `association.association-id` - The association ID returned when the network interface was associated with an IP address.
 - `association.allocation-id` - The allocation ID returned when you allocated the Elastic IP address for your network interface.
 - `association.ip-owner-id` - The owner of the Elastic IP address associated with the network interface.
 - `association.public-ip` - The address of the Elastic IP address bound to the network interface.

- `association.public-dns-name` - The public DNS name for the network interface.
- `attachment.attachment-id` - The ID of the interface attachment.
- `attachment.instance-id` - The ID of the instance to which the network interface is attached.
- `attachment.instance-owner-id` - The owner ID of the instance to which the network interface is attached.
- `attachment.device-index` - The device index to which the network interface is attached.
- `attachment.status` - The status of the attachment (`attaching`|`attached`|`detaching`|`detached`).
- `attachment.attach.time` - The time that the network interface was attached to an instance.
- `attachment.delete-on-termination` - Indicates whether the attachment is deleted when an instance is terminated.
- `availability-zone` - The Availability Zone of the network interface.
- `description` - The description of the network interface.
- `group-id` - The ID of a security group associated with the network interface.
- `group-name` - The name of a security group associated with the network interface.
- `mac-address` - The MAC address of the network interface.
- `network-interface-id` - The ID of the network interface.
- `owner-id` - The AWS account ID of the network interface owner.
- `private-ip-address` - The private IP address or addresses of the network interface.
- `private-dns-name` - The private DNS name of the network interface.
- `requester-id` - The ID of the entity that launched the instance on your behalf (for example, AWS Management Console, Auto Scaling, and so on).
- `requester-managed` - Indicates whether the network interface is being managed by an AWS service (for example, AWS Management Console, Auto Scaling, and so on).
- `source-check` - Indicates whether the network interface performs source/destination checking. A value of `true` means checking is enabled, and `false` means checking is disabled. The value must be `false` for the network interface to perform Network Address Translation (NAT) in your VPC.
- `status` - The status of the network interface. If the network interface is not attached to an instance, the status is `available`; if a network interface is attached to an instance the status is `in-use`.
- `subnet-id` - The ID of the subnet for the network interface.
- `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag’s value is), and the tag value `X` (regardless of what the tag’s key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.

- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `vpc-id` - The ID of the VPC for the network interface.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

Return type `list(ec2.NetworkInterface)`

Returns A list of `NetworkInterface` resources

limit (***kwargs*)

Creates an iterable up to a specified amount of `NetworkInterface` resources in the collection.

Request Syntax

```
network_interface_iterator = ec2.network_interfaces.limit(
    count=123
)
```

Parameters `count` (*integer*) - The limit to the number of resources in the iterable.

Return type `list(ec2.NetworkInterface)`

Returns A list of `NetworkInterface` resources

page_size (***kwargs*)

Creates an iterable of all `NetworkInterface` resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
network_interface_iterator = ec2.network_interfaces.page_size(
    count=123
)
```

Parameters `count` (*integer*) - The number of items returned by each service call

Return type `list(ec2.NetworkInterface)`

Returns A list of `NetworkInterface` resources

placement_groups

all ()

Creates an iterable of all `PlacementGroup` resources in the collection.

Request Syntax

```
placement_group_iterator = ec2.placement_groups.all()
```

Return type `list(ec2.PlacementGroup)`

Returns A list of `PlacementGroup` resources

filter (**kwargs)

Creates an iterable of all PlacementGroup resources in the collection filtered by kwargs passed to method.

Request Syntax

```
placement_group_iterator = ec2.placement_groups.filter(
    DryRun=True|False,
    GroupNames=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **GroupNames** (*list*) – One or more placement group names.

Default: Describes all your placement groups, or only those otherwise specified.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `group-name` - The name of the placement group.
 - `state` - The state of the placement group (`pending` | `available` | `deleting` | `deleted`).
 - `strategy` - The strategy of the placement group (`cluster`).
- (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

- * **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

- * **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

- (*string*) –

Return type `list(ec2.PlacementGroup)`

Returns A list of PlacementGroup resources

limit (**kwargs)

Creates an iterable up to a specified amount of PlacementGroup resources in the collection.

Request Syntax

```
placement_group_iterator = ec2.placement_groups.limit(
    count=123
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(ec2.PlacementGroup)`

Returns A list of PlacementGroup resources

page_size (***kwargs*)

Creates an iterable of all PlacementGroup resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
placement_group_iterator = ec2.placement_groups.page_size(
    count=123
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(ec2.PlacementGroup)`

Returns A list of PlacementGroup resources

route_tables

all ()

Creates an iterable of all RouteTable resources in the collection.

Request Syntax

```
route_table_iterator = ec2.route_tables.all()
```

Return type `list(ec2.RouteTable)`

Returns A list of RouteTable resources

filter (***kwargs*)

Creates an iterable of all RouteTable resources in the collection filtered by kwargs passed to method.

Request Syntax

```
route_table_iterator = ec2.route_tables.filter(
    DryRun=True|False,
    RouteTableIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **RouteTableIds** (*list*) – One or more route table IDs.

Default: Describes all your route tables.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `association.route-table-association-id` - The ID of an association ID for the route table.
 - `association.route-table-id` - The ID of the route table involved in the association.
 - `association.subnet-id` - The ID of the subnet involved in the association.
 - `association.main` - Indicates whether the route table is the main route table for the VPC.
 - `route-table-id` - The ID of the route table.
 - `route.destination-cidr-block` - The CIDR range specified in a route in the table.
 - `route.destination-prefix-list-id` - The ID (prefix) of the AWS service specified in a route in the table.
 - `route.gateway-id` - The ID of a gateway specified in a route in the table.
 - `route.instance-id` - The ID of an instance specified in a route in the table.
 - `route.origin` - Describes how the route was created. `CreateRouteTable` indicates that the route was automatically created when the route table was created; `CreateRoute` indicates that the route was manually added to the route table; `EnableVgwRoutePropagation` indicates that the route was propagated by route propagation.
 - `route.state` - The state of a route in the route table (`active` | `blackhole`). The blackhole state indicates that the route's target isn't available (for example, the specified gateway isn't attached to the VPC, the specified NAT instance has been terminated, and so on).
 - `route.vpc-peering-connection-id` - The ID of a VPC peering connection specified in a route in the table.
 - `tag:key=*value*` - The key/value combination of a tag assigned to the resource.
 - `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “tag-key=Purpose” and the filter “tag-value=X”, you get any resources assigned both the tag key Purpose (regardless of what the tag's value is), and the tag value X (regardless of what the tag's key is). If you want to list only resources where Purpose is X, see the `tag:key=*value*` filter.
 - `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
 - `vpc-id` - The ID of the VPC for the route table.
- (*dict*) –

A filter name and value pair that is used to return a more specific list

of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

Return type `list(ec2.RouteTable)`

Returns A list of RouteTable resources

limit (***kwargs*)

Creates an iterable up to a specified amount of RouteTable resources in the collection.

Request Syntax

```
route_table_iterator = ec2.route_tables.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(ec2.RouteTable)`

Returns A list of RouteTable resources

page_size (***kwargs*)

Creates an iterable of all RouteTable resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
route_table_iterator = ec2.route_tables.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(ec2.RouteTable)`

Returns A list of RouteTable resources

security_groups

all ()

Creates an iterable of all SecurityGroup resources in the collection.

Request Syntax

```
security_group_iterator = ec2.security_groups.all()
```

Return type `list(ec2.SecurityGroup)`

Returns A list of SecurityGroup resources

filter (***kwargs*)

Creates an iterable of all SecurityGroup resources in the collection filtered by kwargs passed to method.

Request Syntax

```
security_group_iterator = ec2.security_groups.filter(  
    DryRun=True|False,  
    GroupNames=[  
        'string',  
    ],  
    GroupIds=[  
        'string',  
    ],  
    Filters=[  
        {  
            'Name': 'string',  
            'Values': [  
                'string',  
            ]  
        },  
    ],  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **GroupNames** (*list*) – [EC2-Classic and default VPC only] One or more security group names. You can specify either the security group name or the security group ID. For security groups in a nondefault VPC, use the `group-name` filter to describe security groups by name.

Default: Describes all your security groups.

– (*string*) –

- **GroupIds** (*list*) – One or more security group IDs. Required for security groups in a nondefault VPC.

Default: Describes all your security groups.

– (*string*) –

- **Filters** (*list*) – One or more filters. If using multiple filters for rules, the results include security groups for which any combination of rules - not necessarily a single rule - match all filters.

- `description` - The description of the security group.
- `egress.ip-permission.prefix-list-id` - The ID (prefix) of the AWS service to which the security group allows access.
- `group-id` - The ID of the security group.
- `group-name` - The name of the security group.
- `ip-permission.cidr` - A CIDR range that has been granted permission.
- `ip-permission.from-port` - The start of port range for the TCP and UDP protocols, or an ICMP type number.
- `ip-permission.group-id` - The ID of a security group that has been granted permission.
- `ip-permission.group-name` - The name of a security group that has been granted permission.
- `ip-permission.protocol` - The IP protocol for the permission (`tcp` | `udp` | `icmp` or a protocol number).
- `ip-permission.to-port` - The end of port range for the TCP and UDP protocols, or an ICMP code.
- `ip-permission.user-id` - The ID of an AWS account that

has been granted permission.

- `owner-id` - The AWS account ID of the owner of the security group.
- `tag-key` - The key of a tag assigned to the security group.
- `tag-value` - The value of a tag assigned to the security group.
- `vpc-id` - The ID of the VPC specified when the security group was created.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

Return type `list(ec2.SecurityGroup)`

Returns A list of SecurityGroup resources

limit (***kwargs*)

Creates an iterable up to a specified amount of SecurityGroup resources in the collection.

Request Syntax

```
security_group_iterator = ec2.security_groups.limit(
    count=123
)
```

Parameters `count` (*integer*) - The limit to the number of resources in the iterable.

Return type `list(ec2.SecurityGroup)`

Returns A list of SecurityGroup resources

page_size (***kwargs*)

Creates an iterable of all SecurityGroup resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
security_group_iterator = ec2.security_groups.page_size(
    count=123
)
```

Parameters `count` (*integer*) - The number of items returned by each service call

Return type `list(ec2.SecurityGroup)`

Returns A list of SecurityGroup resources

snapshots

all ()

Creates an iterable of all Snapshot resources in the collection.

Request Syntax

```
snapshot_iterator = ec2.snapshots.all()
```

Return type `list(ec2.Snapshot)`

Returns A list of Snapshot resources

filter (***kwargs*)

Creates an iterable of all Snapshot resources in the collection filtered by kwargs passed to method.

Request Syntax

```
snapshot_iterator = ec2.snapshots.filter(  
    DryRun=True|False,  
    SnapshotIds=[  
        'string',  
    ],  
    OwnerIds=[  
        'string',  
    ],  
    RestorableByUserIds=[  
        'string',  
    ],  
    Filters=[  
        {  
            'Name': 'string',  
            'Values': [  
                'string',  
            ]  
        },  
    ],  
    NextToken='string',  
    MaxResults=123  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SnapshotIds** (*list*) – One or more snapshot IDs.

Default: Describes snapshots for which you have launch permissions.

– (*string*) –

- **OwnerIds** (*list*) – Returns the snapshots owned by the specified owner. Multiple owners can be specified.

– (*string*) –

- **RestorableByUserIds** (*list*) – One or more AWS accounts IDs that can create volumes from the snapshot.

– (*string*) –

- **Filters** (*list*) – One or more filters.

– *description* - A description of the snapshot.

– *owner-alias* - The AWS account alias (for example, amazon) that owns the snapshot.

– *owner-id* - The ID of the AWS account that owns the snapshot.

– *progress* - The progress of the snapshot, as a percentage (for example, 80%).

– *snapshot-id* - The snapshot ID.

– *start-time* - The time stamp when the snapshot was initiated.

– *status* - The status of the snapshot (*pending* | *completed* | *error*).

- `tag :key=*value*` - The key/value combination of a tag assigned to the resource.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “tag-key=Purpose” and the filter “tag-value=X”, you get any resources assigned both the tag key Purpose (regardless of what the tag’s value is), and the tag value X (regardless of what the tag’s key is). If you want to list only resources where Purpose is X, see the `tag :key=*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `volume-id` - The ID of the volume the snapshot is for.
- `volume-size` - The size of the volume, in GiB.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

- **NextToken** (*string*) - The `NextToken` value returned from a previous paginated `DescribeSnapshots` request where `MaxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `NextToken` value. This value is `null` when there are no more results to return.
- **MaxResults** (*integer*) - The maximum number of snapshot results returned by `DescribeSnapshots` in paginated output. When this parameter is used, `DescribeSnapshots` only returns `MaxResults` results in a single page along with a `NextToken` response element. The remaining results of the initial request can be seen by sending another `DescribeSnapshots` request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. If this parameter is not used, then `DescribeSnapshots` returns all results. You cannot specify this parameter and the snapshot IDs parameter in the same request.

Return type `list(ec2.Snapshot)`

Returns A list of Snapshot resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Snapshot resources in the collection.

Request Syntax

```
snapshot_iterator = ec2.snapshots.limit(
    count=123
)
```

Parameters `count` (*integer*) - The limit to the number of resources in the iterable.

Return type `list(ec2.Snapshot)`

Returns A list of Snapshot resources

page_size (**kwargs)

Creates an iterable of all Snapshot resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
snapshot_iterator = ec2.snapshots.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type list(ec2.Snapshot)

Returns A list of Snapshot resources

subnets**all** ()

Creates an iterable of all Subnet resources in the collection.

Request Syntax

```
subnet_iterator = ec2.subnets.all()
```

Return type list(ec2.Subnet)

Returns A list of Subnet resources

filter (**kwargs)

Creates an iterable of all Subnet resources in the collection filtered by kwargs passed to method.

Request Syntax

```
subnet_iterator = ec2.subnets.filter(  
    DryRun=True|False,  
    SubnetIds=[  
        'string',  
    ],  
    Filters=[  
        {  
            'Name': 'string',  
            'Values': [  
                'string',  
            ],  
        },  
    ],  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SubnetIds** (*list*) – One or more subnet IDs.

Default: Describes all your subnets.

– (*string*) –

- **Filters** (*list*) – One or more filters.
 - `availabilityZone` - The Availability Zone for the subnet. You can also use `availability-zone` as the filter name.

- `available-ip-address-count` - The number of IP addresses in the subnet that are available.
- `cidrBlock` - The CIDR block of the subnet. The CIDR block you specify must exactly match the subnet's CIDR block for information to be returned for the subnet. You can also use `cidr` or `cidr-block` as the filter names.
- `defaultForAz` - Indicates whether this is the default subnet for the Availability Zone. You can also use `default-for-az` as the filter name.
- `state` - The state of the subnet (`pending` | `available`).
- `subnet-id` - The ID of the subnet.
- `tag :key=*value*` - The key/value combination of a tag assigned to the resource.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “tag-key=Purpose” and the filter “tag-value=X”, you get any resources assigned both the tag key Purpose (regardless of what the tag's value is), and the tag value X (regardless of what the tag's key is). If you want to list only resources where Purpose is X, see the `tag :key=*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `vpc-id` - The ID of the VPC for the subnet.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

Return type `list(ec2.Subnet)`

Returns A list of Subnet resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Subnet resources in the collection.

Request Syntax

```
subnet_iterator = ec2.subnets.limit(
    count=123
)
```

Parameters `count` (*integer*) - The limit to the number of resources in the iterable.

Return type `list(ec2.Subnet)`

Returns A list of Subnet resources

page_size (***kwargs*)

Creates an iterable of all Subnet resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
subnet_iterator = ec2.subnets.page_size(
    count=123
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(ec2.Subnet)`

Returns A list of Subnet resources

volumes

all()

Creates an iterable of all Volume resources in the collection.

Request Syntax

```
volume_iterator = ec2.volumes.all()
```

Return type `list(ec2.Volume)`

Returns A list of Volume resources

filter (***kwargs*)

Creates an iterable of all Volume resources in the collection filtered by kwargs passed to method.

Request Syntax

```
volume_iterator = ec2.volumes.filter(
    DryRun=True|False,
    VolumeIds=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VolumeIds** (*list*) – One or more volume IDs.
 - (*string*) –
- **Filters** (*list*) – One or more filters.
 - `attachment.attach-time` - The time stamp when the attachment initiated.
 - `attachment.delete-on-termination` - Whether the volume is deleted on instance termination.
 - `attachment.device` - The device name that is exposed to the instance (for example, `/dev/sda1`).

- `attachment.instance-id` - The ID of the instance the volume is attached to.
- `attachment.status` - The attachment state (`attaching` | `attached` | `detaching` | `detached`).
- `availability-zone` - The Availability Zone in which the volume was created.
- `create-time` - The time stamp when the volume was created.
- `encrypted` - The encryption status of the volume.
- `size` - The size of the volume, in GiB.
- `snapshot-id` - The snapshot from which the volume was created.
- `status` - The status of the volume (`creating` | `available` | `in-use` | `deleting` | `deleted` | `error`).
- `tag :key=*value*` - The key/value combination of a tag assigned to the resource.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “tag-key=Purpose” and the filter “tag-value=X”, you get any resources assigned both the tag key Purpose (regardless of what the tag’s value is), and the tag value X (regardless of what the tag’s key is). If you want to list only resources where Purpose is X, see the `tag :key=*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `volume-id` - The volume ID.
- `volume-type` - The Amazon EBS volume type. This can be `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, or `standard` for Magnetic volumes.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

- **NextToken** (*string*) - The `NextToken` value returned from a previous paginated `DescribeVolumes` request where `MaxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `NextToken` value. This value is `null` when there are no more results to return.
- **MaxResults** (*integer*) - The maximum number of volume results returned by `DescribeVolumes` in paginated output. When this parameter is used, `DescribeVolumes` only returns `MaxResults` results in a single page along with a `NextToken` response element. The remaining results of the initial request can be seen by sending another `DescribeVolumes` request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. If this parameter is not used, then `DescribeVolumes` returns all results. You cannot specify this parameter and the volume IDs parameter in the same request.

Return type `list(ec2.Volume)`

Returns A list of Volume resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Volume resources in the collection.

Request Syntax

```
volume_iterator = ec2.volumes.limit(  
    count=123  
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type list(ec2.Volume)

Returns A list of Volume resources

page_size (***kwargs*)

Creates an iterable of all Volume resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
volume_iterator = ec2.volumes.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type list(ec2.Volume)

Returns A list of Volume resources

vpc_peering_connections

all ()

Creates an iterable of all VpcPeeringConnection resources in the collection.

Request Syntax

```
vpc_peering_connection_iterator = ec2.vpc_peering_connections.all()
```

Return type list(ec2.VpcPeeringConnection)

Returns A list of VpcPeeringConnection resources

filter (***kwargs*)

Creates an iterable of all VpcPeeringConnection resources in the collection filtered by kwargs passed to method.

Request Syntax

```
vpc_peering_connection_iterator = ec2.vpc_peering_connections.filter(  
    DryRun=True|False,  
    VpcPeeringConnectionIds=[  
        'string',  
    ],  
    Filters=[  
        {  
            'Name': 'string',  
            'Values': [  
                'string',  
            ],  
        },  
    ],
```

)]

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcPeeringConnectionIds** (*list*) – One or more VPC peering connection IDs.

Default: Describes all your VPC peering connections.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `accepter-vpc-info.cidr-block` - The CIDR block of the peer VPC.
 - `accepter-vpc-info.owner-id` - The AWS account ID of the owner of the peer VPC.
 - `accepter-vpc-info.vpc-id` - The ID of the peer VPC.
 - `expiration-time` - The expiration date and time for the VPC peering connection.
 - `requester-vpc-info.cidr-block` - The CIDR block of the requester's VPC.
 - `requester-vpc-info.owner-id` - The AWS account ID of the owner of the requester VPC.
 - `requester-vpc-info.vpc-id` - The ID of the requester VPC.
 - `status-code` - The status of the VPC peering connection (`pending-acceptance` | `failed` | `expired` | `provisioning` | `active` | `deleted` | `rejected`).
 - `status-message` - A message that provides more information about the status of the VPC peering connection, if applicable.
 - `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
 - `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag's value is), and the tag value `X` (regardless of what the tag's key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
 - `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
 - `vpc-peering-connection-id` - The ID of the VPC peering connection.
- (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

Return type `list(ec2.VpcPeeringConnection)`

Returns A list of `VpcPeeringConnection` resources

limit (***kwargs*)

Creates an iterable up to a specified amount of `VpcPeeringConnection` resources in the collection.

Request Syntax

```
vpc_peering_connection_iterator = ec2.vpc_peering_connections.limit(  
    count=123  
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(ec2.VpcPeeringConnection)`

Returns A list of `VpcPeeringConnection` resources

page_size (***kwargs*)

Creates an iterable of all `VpcPeeringConnection` resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
vpc_peering_connection_iterator = ec2.vpc_peering_connections.page_size(  
    count=123  
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(ec2.VpcPeeringConnection)`

Returns A list of `VpcPeeringConnection` resources

vpcs

all ()

Creates an iterable of all `Vpc` resources in the collection.

Request Syntax

```
vpc_iterator = ec2.vpcs.all()
```

Return type `list(ec2.Vpc)`

Returns A list of `Vpc` resources

filter (***kwargs*)

Creates an iterable of all `Vpc` resources in the collection filtered by `kwargs` passed to method.

Request Syntax

```
vpc_iterator = ec2.vpcs.filter(  
    DryRun=True|False,  
    VpcIds=[  
        'string',  
    ],  
    Filters=[  
        {  
            'Name': 'string',  
            'Values': [  
                'string',  
            ]  
        },  
    ]  
)
```

```

    ]
    },
]
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcIds** (*list*) – One or more VPC IDs.

Default: Describes all your VPCs.

- (*string*) –
- **Filters** (*list*) – One or more filters.
 - `cidr` - The CIDR block of the VPC. The CIDR block you specify must exactly match the VPC's CIDR block for information to be returned for the VPC. Must contain the slash followed by one or two digits (for example, `/28`).
 - `dhcp-options-id` - The ID of a set of DHCP options.
 - `isDefault` - Indicates whether the VPC is the default VPC.
 - `state` - The state of the VPC (`pending` | `available`).
 - `tag :key=*value*` - The key/value combination of a tag assigned to the resource.
 - `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag's value is), and the tag value `X` (regardless of what the tag's key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key=*value*` filter.
 - `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
 - `vpc-id` - The ID of the VPC.
- (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

Return type `list(ec2.Vpc)`

Returns A list of Vpc resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Vpc resources in the collection.

Request Syntax

```
vpc_iterator = ec2.vpcs.limit(
    count=123
)
```

```
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(ec2.Vpc)`

Returns A list of Vpc resources

page_size (***kwargs*)

Creates an iterable of all Vpc resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
vpc_iterator = ec2.vpcs.page_size(  
    count=123  
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(ec2.Vpc)`

Returns A list of Vpc resources

DhcpOptions

class `EC2.DhcpOptions` (*id*)

A resource representing an Amazon Elastic Compute Cloud (EC2) DhcpOptions:

```
import boto3
```

```
ec2 = boto3.resource('ec2')  
dhcp_options = ec2.DhcpOptions('id')
```

Parameters `id` (*string*) – The DhcpOptions’s id identifier. This **must** be set.

These are the resource’s available identifiers:

- `id`

These are the resource’s available attributes:

- `dhcp_configurations`
- `dhcp_options_id`
- `tags`

These are the resource’s available actions:

- `associate_with_vpc()`
- `create_tags()`
- `delete()`
- `load()`
- `reload()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

`id`

(*string*) The DhcpOptions’s id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

dhcp_configurations*(list)*

One or more DHCP options in the set.

dhcp_options_id*(string)*

The ID of the set of DHCP options.

tags*(list)*

Any tags assigned to the DHCP options set.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

associate_with_vpc (***kwargs*)

Associates a set of DHCP options (that you've previously created) with the specified VPC, or associates no DHCP options with the VPC.

After you associate the options with the VPC, any existing instances and all new instances that you launch in that VPC use the options. You don't need to restart or relaunch the instances. They automatically pick up the changes within a few hours, depending on how frequently the instance renews its DHCP lease. You can explicitly renew the lease using the operating system on the instance.

For more information, see [DHCP Options Sets](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = dhcp_options.associate_with_vpc(
    DryRun=True|False,
    VpcId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcId** (*string*) – **[REQUIRED]**

The ID of the VPC.

Returns None**create_tags** (***kwargs*)

Adds or overwrites one or more tags for the specified Amazon EC2 resource or resources. Each resource can have a maximum of 10 tags. Each tag consists of a key and optional value. Tag keys must be unique per resource.

For more information about tags, see [Tagging Your Resources](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
tag = dhcp_options.create_tags(
    DryRun=True|False,
    Tags=[
        {
```

```
        'Key': 'string',
        'Value': 'string'
    },
]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **Tags** (*list*) – **[REQUIRED]**

One or more tags. The `value` parameter is required, but if you don't want the tag to have a value, specify the parameter with no value, and we set the value to an empty string.

– (*dict*) –

Describes a tag.

* **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

* **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

Return type `list(ec2.Tag)`

Returns A list of Tag resources

delete (***kwargs*)

Deletes the specified set of DHCP options. You must disassociate the set of DHCP options before you can delete it. You can disassociate the set of DHCP options by associating either a new set of options or the default set of options with the VPC.

Request Syntax

```
response = dhcp_options.delete(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns `None`

load ()

Calls `ec2.Client.describe_dhcp_options()` to update the attributes of the `DhcpOptions` resource

Request Syntax

```
dhcp_options.load()
```


Returns None

reload()

Calls `ec2.Client.describe_dhcp_options()` to update the attributes of the `DhcpOptions` resource

Request Syntax

```
dhcp_options.reload()
```

Returns None

Image

class `EC2.Image(id)`

A resource representing an Amazon Elastic Compute Cloud (EC2) Image:

```
import boto3
```

```
ec2 = boto3.resource('ec2')
image = ec2.Image('id')
```

Parameters `id` (*string*) – The Image’s id identifier. This **must** be set.

These are the resource’s available identifiers:

- `id`

These are the resource’s available attributes:

- `architecture`
- `block_device_mappings`
- `creation_date`
- `description`
- `hypervisor`
- `image_id`
- `image_location`
- `image_owner_alias`
- `image_type`
- `kernel_id`
- `name`
- `owner_id`
- `platform`
- `product_codes`
- `public`
- `ramdisk_id`
- `root_device_name`
- `root_device_type`
- `sriov_net_support`
- `state`
- `state_reason`
- `tags`
- `virtualization_type`

These are the resource’s available actions:

- `create_tags()`
- `deregister()`
- `describe_attribute()`
- `load()`

- `modify_attribute()`
- `reload()`
- `reset_attribute()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

id

(*string*) The Image's id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

architecture

(*string*)

The architecture of the image.

block_device_mappings

(*list*)

Any block device mapping entries.

creation_date

(*string*)

The date and time the image was created.

description

(*string*)

The description of the AMI that was provided during image creation.

hypervisor

(*string*)

The hypervisor type of the image.

image_id

(*string*)

The ID of the AMI.

image_location

(*string*)

The location of the AMI.

image_owner_alias

(*string*)

The AWS account alias (for example, `amazon`, `self`) or the AWS account ID of the AMI owner.

image_type

(*string*)

The type of image.

kernel_id

(*string*)

The kernel associated with the image, if any. Only applicable for machine images.

name
(string)
The name of the AMI that was provided during image creation.

owner_id
(string)
The AWS account ID of the image owner.

platform
(string)
The value is `Windows` for Windows AMIs; otherwise blank.

product_codes
(list)
Any product codes associated with the AMI.

public
(boolean)
Indicates whether the image has public launch permissions. The value is `true` if this image has public launch permissions or `false` if it has only implicit and explicit launch permissions.

ramdisk_id
(string)
The RAM disk associated with the image, if any. Only applicable for machine images.

root_device_name
(string)
The device name of the root device (for example, `/dev/sda1` or `/dev/xvda`).

root_device_type
(string)
The type of root device used by the AMI. The AMI can use an EBS volume or an instance store volume.

sriov_net_support
(string)
Specifies whether enhanced networking is enabled.

state
(string)
The current state of the AMI. If the state is `available`, the image is successfully registered and can be used to launch an instance.

state_reason
(dict)
The reason for the state change.

tags
(list)
Any tags assigned to the image.

virtualization_type
(string)
The type of virtualization of the AMI.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

create_tags (**kwargs)

Adds or overwrites one or more tags for the specified Amazon EC2 resource or resources. Each resource can have a maximum of 10 tags. Each tag consists of a key and optional value. Tag keys must be unique per resource.

For more information about tags, see [Tagging Your Resources](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
tag = image.create_tags(
    DryRun=True|False,
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Tags** (*list*) – **[REQUIRED]**

One or more tags. The `value` parameter is required, but if you don't want the tag to have a value, specify the parameter with no value, and we set the value to an empty string.

– (*dict*) –

Describes a tag.

* **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

* **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

Return type `list(ec2.Tag)`

Returns A list of Tag resources

deregister (**kwargs)

Deregisters the specified AMI. After you deregister an AMI, it can't be used to launch new instances.

This command does not delete the AMI.

Request Syntax

```
response = image.deregister(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns `None`

describe_attribute (***kwargs*)

Describes the specified attribute of the specified AMI. You can specify only one attribute at a time.

Request Syntax

```
response = image.describe_attribute(
    DryRun=True|False,
    Attribute='description'|'kernel'|'ramdisk'|'launchPermission'|'productCodes'|'blockDevi
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Attribute** (*string*) – **[REQUIRED]**

The AMI attribute.

Note : Depending on your account privileges, the `blockDeviceMapping` attribute may return a `Client.AuthFailure` error. If this happens, use `DescribeImages` to get information about the block device mapping for the AMI.

Return type `dict`

Returns

Response Syntax

```
{
  'ImageId': 'string',
  'LaunchPermissions': [
    {
      'UserId': 'string',
      'Group': 'all'
    },
  ],
  'ProductCodes': [
    {
      'ProductCodeId': 'string',
      'ProductCodeType': 'devpay'|'marketplace'
    },
  ],
  'KernelId': {
    'Value': 'string'
  },
  'RamdiskId': {
    'Value': 'string'
  },
}
```

```
'Description': {
    'Value': 'string'
},
'SriovNetSupport': {
    'Value': 'string'
},
'BlockDeviceMappings': [
    {
        'VirtualName': 'string',
        'DeviceName': 'string',
        'Ebs': {
            'SnapshotId': 'string',
            'VolumeSize': 123,
            'DeleteOnTermination': True|False,
            'VolumeType': 'standard'|'io1'|'gp2',
            'Iops': 123,
            'Encrypted': True|False
        },
        'NoDevice': 'string'
    },
    ]
}
```

Response Structure

- (*dict*) –

Information about the image attribute.

- **ImageId** (*string*) –

The ID of the AMI.

- **LaunchPermissions** (*list*) –

One or more launch permissions.

- * (*dict*) –

Describes a launch permission.

- **UserId** (*string*) –

The AWS account ID.

- **Group** (*string*) –

The name of the group.

- **ProductCodes** (*list*) –

One or more product codes.

- * (*dict*) –

Describes a product code.

- **ProductCodeId** (*string*) –

The product code.

- **ProductCodeType** (*string*) –

The type of product code.

- **KernelId** (*dict*) –

The kernel ID.

- * **Value** (*string*) –

Valid values are case-sensitive and vary by action.

– **RamdiskId** (*dict*) –

The RAM disk ID.

* **Value** (*string*) –

Valid values are case-sensitive and vary by action.

– **Description** (*dict*) –

A description for the AMI.

* **Value** (*string*) –

Valid values are case-sensitive and vary by action.

– **SriovNetSupport** (*dict*) –

The value to use for a resource attribute.

* **Value** (*string*) –

Valid values are case-sensitive and vary by action.

– **BlockDeviceMappings** (*list*) –

One or more block device mapping entries.

* (*dict*) –

Describes a block device mapping.

· **VirtualName** (*string*) –

The virtual device name (ephemeral N). Instance store volumes are numbered starting from 0. An instance type with 2 available instance store volumes can specify mappings for `ephemeral0` and `ephemeral1`. The number of available instance store volumes depends on the instance type. After you connect to the instance, you must mount the volume.

Constraints: For M3 instances, you must specify instance store volumes in the block device mapping for the instance. When you launch an M3 instance, we ignore any instance store volumes specified in the block device mapping for the AMI.

· **DeviceName** (*string*) –

The device name exposed to the instance (for example, `/dev/sdh` or `xvdh`).

· **Ebs** (*dict*) –

Parameters used to automatically set up EBS volumes when the instance is launched.

· **SnapshotId** (*string*) –

The ID of the snapshot.

· **VolumeSize** (*integer*) –

The size of the volume, in GiB.

Constraints: 1–1024 for standard volumes, 1–16384 for `gp2` volumes, and 4–16384 for `io1` volumes. If you specify a snapshot, the volume size must be equal to or larger than the snapshot size.

Default: If you’re creating the volume from a snapshot and don’t specify a volume size, the default is the snapshot size.

· **DeleteOnTermination** (*boolean*) –

Indicates whether the EBS volume is deleted on instance termination.

- **VolumeType** (*string*) –

The volume type. `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, and `standard` for Magnetic volumes.

Default: `standard`

- **Iops** (*integer*) –

The number of I/O operations per second (IOPS) that the volume supports. For Provisioned IOPS (SSD) volumes, this represents the number of IOPS that are provisioned for the volume. For General Purpose (SSD) volumes, this represents the baseline performance of the volume and the rate at which the volume accumulates I/O credits for bursting. For more information on General Purpose (SSD) baseline performance, I/O credits, and bursting, see [Amazon EBS Volume Types](#) in the *Amazon Elastic Compute Cloud User Guide*.

Constraint: Range is 100 to 20000 for Provisioned IOPS (SSD) volumes and 3 to 10000 for General Purpose (SSD) volumes.

Condition: This parameter is required for requests to create `io1` volumes; it is not used in requests to create `standard` or `gp2` volumes.

- **Encrypted** (*boolean*) –

Indicates whether the EBS volume is encrypted. Encrypted Amazon EBS volumes may only be attached to instances that support Amazon EBS encryption.

- **NoDevice** (*string*) –

Suppresses the specified device included in the block device mapping of the AMI.

load()

Calls `ec2.Client.describe_images()` to update the attributes of the Image resource

Request Syntax

```
image.load()
```

Returns None

modify_attribute(kwargs)**

Modifies the specified attribute of the specified AMI. You can specify only one attribute at a time.

Note: AWS Marketplace product codes cannot be modified. Images with an AWS Marketplace product code cannot be made public.

Request Syntax

```
response = image.modify_attribute(  
    DryRun=True|False,  
    Attribute='string',  
    OperationType='add'|'remove',
```



```

    UserIds=[
        'string',
    ],
    UserGroups=[
        'string',
    ],
    ProductCodes=[
        'string',
    ],
    Value='string',
    LaunchPermission={
        'Add': [
            {
                'UserId': 'string',
                'Group': 'all'
            },
        ],
        'Remove': [
            {
                'UserId': 'string',
                'Group': 'all'
            },
        ]
    },
    Description={
        'Value': 'string'
    }
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Attribute** (*string*) – The name of the attribute to modify.
- **OperationType** (*string*) – The operation type.
- **UserIds** (*list*) – One or more AWS account IDs. This is only valid when modifying the `launchPermission` attribute.
 - (*string*) –
- **UserGroups** (*list*) – One or more user groups. This is only valid when modifying the `launchPermission` attribute.
 - (*string*) –
- **ProductCodes** (*list*) – One or more product codes. After you add a product code to an AMI, it can't be removed. This is only valid when modifying the `productCodes` attribute.
 - (*string*) –
- **Value** (*string*) – The value of the attribute being modified. This is only valid when modifying the `description` attribute.
- **LaunchPermission** (*dict*) – A launch permission modification.
 - **Add** (*list*) –

The AWS account ID to add to the list of launch permissions for the AMI.

* (*dict*) –

Describes a launch permission.

· **UserId** (*string*) –

- The AWS account ID.
 - **Group** (*string*) –
 - The name of the group.
 - **Remove** (*list*) –
 - The AWS account ID to remove from the list of launch permissions for the AMI.
 - * (*dict*) –
 - Describes a launch permission.
 - **UserId** (*string*) –
 - The AWS account ID.
 - **Group** (*string*) –
 - The name of the group.
 - **Description** (*dict*) – A description for the AMI.
 - **Value** (*string*) –
 - Valid values are case-sensitive and vary by action.

Returns None

reload()

Calls `ec2.Client.describe_images()` to update the attributes of the Image resource

Request Syntax

```
image.reload()
```

Returns None

reset_attribute (***kwargs*)

Resets an attribute of an AMI to its default value.

Note: The `productCodes` attribute can't be reset.

Request Syntax

```
response = image.reset_attribute(  
    DryRun=True|False,  
    Attribute='launchPermission'  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Attribute** (*string*) – **[REQUIRED]**
 - The attribute to reset (currently you can only reset the launch permission attribute).

Returns None

Instance

class EC2.Instance (*id*)

A resource representing an Amazon Elastic Compute Cloud (EC2) Instance:

```
import boto3
```

```
ec2 = boto3.resource('ec2')
instance = ec2.Instance('id')
```

Parameters `id` (*string*) – The Instance’s id identifier. This **must** be set.

These are the resource’s available identifiers:

- `id`

These are the resource’s available attributes:

- `ami_launch_index`
- `architecture`
- `block_device_mappings`
- `client_token`
- `ebs_optimized`
- `hypervisor`
- `iam_instance_profile`
- `image_id`
- `instance_id`
- `instance_lifecycle`
- `instance_type`
- `kernel_id`
- `key_name`
- `launch_time`
- `monitoring`
- `network_interfaces`
- `placement`
- `platform`
- `private_dns_name`
- `private_ip_address`
- `product_codes`
- `public_dns_name`
- `public_ip_address`
- `ramdisk_id`
- `root_device_name`
- `root_device_type`
- `security_groups`
- `source_dest_check`
- `spot_instance_request_id`
- `sriov_net_support`
- `state`
- `state_reason`
- `state_transition_reason`
- `subnet_id`
- `tags`
- `virtualization_type`
- `vpc_id`

These are the resource’s available references:

- `image`
- `key_pair`
- `placement_group`
- `subnet`
- `vpc`

These are the resource’s available actions:

- `attach_classic_link_vpc()`
- `attach_volume()`
- `console_output()`
- `create_image()`
- `create_tags()`
- `describe_attribute()`
- `detach_classic_link_vpc()`
- `detach_volume()`
- `load()`
- `modify_attribute()`
- `monitor()`
- `password_data()`
- `reboot()`
- `reload()`
- `report_status()`
- `reset_attribute()`
- `reset_kernel()`
- `reset_ramdisk()`
- `reset_source_dest_check()`
- `start()`
- `stop()`
- `terminate()`
- `unmonitor()`

These are the resource's available collections:

- `volumes`

These are the resource's available waiters:

- `wait_until_exists()`
- `wait_until_running()`
- `wait_until_stopped()`
- `wait_until_terminated()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

id

(*string*) The Instance's id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

ami_launch_index

(*integer*)

The AMI launch index, which can be used to find this instance in the launch group.

architecture

(*string*)

The architecture of the image.

block_device_mappings

(*list*)

Any block device mapping entries for the instance.

client_token*(string)*

The idempotency token you provided when you launched the instance.

ebs_optimized*(boolean)*

Indicates whether the instance is optimized for EBS I/O. This optimization provides dedicated throughput to Amazon EBS and an optimized configuration stack to provide optimal I/O performance. This optimization isn't available with all instance types. Additional usage charges apply when using an EBS Optimized instance.

hypervisor*(string)*

The hypervisor type of the instance.

iam_instance_profile*(dict)*

The IAM instance profile associated with the instance.

image_id*(string)*

The ID of the AMI used to launch the instance.

instance_id*(string)*

The ID of the instance.

instance_lifecycle*(string)*

Indicates whether this is a Spot Instance.

instance_type*(string)*

The instance type.

kernel_id*(string)*

The kernel associated with this instance.

key_name*(string)*

The name of the key pair, if this instance was launched with an associated key pair.

launch_time*(datetime)*

The time the instance was launched.

monitoring*(dict)*

The monitoring information for the instance.

network_interfaces*(list)*

[EC2-VPC] One or more network interfaces for the instance.

placement

(dict)

The location where the instance launched.

platform

(string)

The value is `Windows` for Windows instances; otherwise blank.

private_dns_name

(string)

The private DNS name assigned to the instance. This DNS name can only be used inside the Amazon EC2 network. This name is not available until the instance enters the `running` state.

private_ip_address

(string)

The private IP address assigned to the instance.

product_codes

(list)

The product codes attached to this instance.

public_dns_name

(string)

The public DNS name assigned to the instance. This name is not available until the instance enters the `running` state.

public_ip_address

(string)

The public IP address assigned to the instance.

ramdisk_id

(string)

The RAM disk associated with this instance.

root_device_name

(string)

The root device name (for example, `/dev/sda1` or `/dev/xvda`).

root_device_type

(string)

The root device type used by the AMI. The AMI can use an EBS volume or an instance store volume.

security_groups

(list)

One or more security groups for the instance.

source_dest_check

(boolean)

Specifies whether to enable an instance launched in a VPC to perform NAT. This controls whether source/destination checking is enabled on the instance. A value of `true` means checking is enabled, and `false` means checking is disabled. The value must be `false` for the instance to perform NAT. For more information, see [NAT Instances](#) in the *Amazon Virtual Private Cloud User Guide*.

spot_instance_request_id*(string)*

The ID of the Spot Instance request.

sriov_net_support*(string)*

Specifies whether enhanced networking is enabled.

state*(dict)*

The current state of the instance.

state_reason*(dict)*

The reason for the most recent state transition.

state_transition_reason*(string)*

The reason for the most recent state transition. This might be an empty string.

subnet_id*(string)*

The ID of the subnet in which the instance is running.

tags*(list)*

Any tags assigned to the instance.

virtualization_type*(string)*

The virtualization type of the instance.

vpc_id*(string)*

The ID of the VPC in which the instance is running.

References

References are related resource instances that have a belongs-to relationship. For more information about references refer to the [Resources Introduction Guide](#).

image*(Image)* The related image if set, otherwise None.**key_pair***(KeyPair)* The related key_pair if set, otherwise None.**placement_group***(PlacementGroup)* The related placement_group if set, otherwise None.**subnet***(Subnet)* The related subnet if set, otherwise None.**vpc***(Vpc)* The related vpc if set, otherwise None.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

attach_classic_link_vpc (**kwargs)

Links an EC2-Classic instance to a ClassicLink-enabled VPC through one or more of the VPC's security groups. You cannot link an EC2-Classic instance to more than one VPC at a time. You can only link an instance that's in the `running` state. An instance is automatically unlinked from a VPC when it's stopped - you can link it to the VPC again when you restart it.

After you've linked an instance, you cannot change the VPC security groups that are associated with it. To change the security groups, you must first unlink the instance, and then link it again.

Linking your instance to a VPC is sometimes referred to as *attaching* your instance.

Request Syntax

```
response = instance.attach_classic_link_vpc(
    DryRun=True|False,
    VpcId='string',
    Groups=[
        'string',
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcId** (*string*) – **[REQUIRED]**
The ID of a ClassicLink-enabled VPC.
- **Groups** (*list*) – **[REQUIRED]**
The ID of one or more of the VPC's security groups. You cannot specify security groups from a different VPC.
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'Return': True|False
}
```

Response Structure

- (*dict*) –
 - **Return** (*boolean*) –
Returns `true` if the request succeeds; otherwise, it returns an error.

attach_volume (**kwargs)

Attaches an EBS volume to a running or stopped instance and exposes it to the instance with the specified device name.

Encrypted EBS volumes may only be attached to instances that support Amazon EBS encryption. For more information, see [Amazon EBS Encryption](#) in the *Amazon Elastic Compute Cloud User Guide*.

For a list of supported device names, see [Attaching an EBS Volume to an Instance](#) . Any device names that aren't reserved for instance store volumes can be used for EBS volumes. For more information, see [Amazon EC2 Instance Store](#) in the *Amazon Elastic Compute Cloud User Guide* .

If a volume has an AWS Marketplace product code:

- The volume can be attached only to a stopped instance.
- AWS Marketplace product codes are copied from the volume to the instance.
- You must be subscribed to the product.
- The instance type and operating system of the instance must support the product. For example, you can't detach a volume from a Windows instance and attach it to a Linux instance.

For an overview of the AWS Marketplace, see [Introducing AWS Marketplace](#) .

For more information about EBS volumes, see [Attaching Amazon EBS Volumes](#) in the *Amazon Elastic Compute Cloud User Guide* .

Request Syntax

```
response = instance.attach_volume(
    DryRun=True|False,
    VolumeId='string',
    Device='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation` . Otherwise, it is `UnauthorizedOperation` .
- **VolumeId** (*string*) – **[REQUIRED]**
The ID of the EBS volume. The volume and instance must be within the same Availability Zone.
- **Device** (*string*) – **[REQUIRED]**
The device name to expose to the instance (for example, `/dev/sdh` or `xvdh`).

Return type dict

Returns

Response Syntax

```
{
    'VolumeId': 'string',
    'InstanceId': 'string',
    'Device': 'string',
    'State': 'attaching'|'attached'|'detaching'|'detached',
    'AttachTime': datetime(2015, 1, 1),
    'DeleteOnTermination': True|False
}
```

Response Structure

- (*dict*) –
Information about the volume attachment.
 - **VolumeId** (*string*) –
The ID of the volume.
 - **InstanceId** (*string*) –
The ID of the instance.

- **Device** (*string*) –
The device name.
- **State** (*string*) –
The attachment state of the volume.
- **AttachTime** (*datetime*) –
The time stamp when the attachment initiated.
- **DeleteOnTermination** (*boolean*) –
Indicates whether the EBS volume is deleted on instance termination.

console_output (***kwargs*)

Gets the console output for the specified instance.

Instances do not have a physical monitor through which you can view their console output. They also lack physical controls that allow you to power up, reboot, or shut them down. To allow these actions, we provide them through the Amazon EC2 API and command line interface.

Instance console output is buffered and posted shortly after instance boot, reboot, and termination. Amazon EC2 preserves the most recent 64 KB output which is available for at least one hour after the most recent post.

For Linux instances, the instance console output displays the exact console output that would normally be displayed on a physical monitor attached to a computer. This output is buffered because the instance produces it and then posts it to a store where the instance’s owner can retrieve it.

For Windows instances, the instance console output includes output from the EC2Config service.

Request Syntax

```
response = instance.console_output(  
    DryRun=True|False,  
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns**Response Syntax**

```
{  
    'InstanceId': 'string',  
    'Timestamp': datetime(2015, 1, 1),  
    'Output': 'string'  
}
```

Response Structure

- (*dict*) –
 - **InstanceId** (*string*) –
The ID of the instance.
 - **Timestamp** (*datetime*) –
The time the output was last updated.

– **Output** (*string*) –

The console output, Base64 encoded.

create_image (***kwargs*)

Creates an Amazon EBS-backed AMI from an Amazon EBS-backed instance that is either running or stopped.

If you customized your instance with instance store volumes or EBS volumes in addition to the root device volume, the new AMI contains block device mapping information for those volumes. When you launch an instance from this new AMI, the instance automatically launches with those additional volumes.

For more information, see [Creating Amazon EBS-Backed Linux AMIs](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
image = instance.create_image(
    DryRun=True|False,
    Name='string',
    Description='string',
    NoReboot=True|False,
    BlockDeviceMappings=[
        {
            'VirtualName': 'string',
            'DeviceName': 'string',
            'Ebs': {
                'SnapshotId': 'string',
                'VolumeSize': 123,
                'DeleteOnTermination': True|False,
                'VolumeType': 'standard'|'io1'|'gp2',
                'Iops': 123,
                'Encrypted': True|False
            },
            'NoDevice': 'string'
        },
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Name** (*string*) – **[REQUIRED]**
A name for the new image.

Constraints: 3-128 alphanumeric characters, parentheses (()), square brackets ([]), spaces (), periods (.), slashes (/), dashes (-), single quotes (‘), at-signs (@), or underscores(_)
- **Description** (*string*) – A description for the new image.
- **NoReboot** (*boolean*) – By default, this parameter is set to `false`, which means Amazon EC2 attempts to shut down the instance cleanly before image creation and then reboots the instance. When the parameter is set to `true`, Amazon EC2 doesn't shut down the instance before creating the image. When this option is used, file system integrity on the created image can't be guaranteed.
- **BlockDeviceMappings** (*list*) – Information about one or more block device mappings.

– (*dict*) –

Describes a block device mapping.

* **VirtualName** (*string*) –

The virtual device name (ephemeral N). Instance store volumes are numbered starting from 0. An instance type with 2 available instance store volumes can specify mappings for `ephemeral0` and `ephemeral1`. The number of available instance store volumes depends on the instance type. After you connect to the instance, you must mount the volume.

Constraints: For M3 instances, you must specify instance store volumes in the block device mapping for the instance. When you launch an M3 instance, we ignore any instance store volumes specified in the block device mapping for the AMI.

* **DeviceName** (*string*) –

The device name exposed to the instance (for example, `/dev/sdh` or `xvdh`).

* **Ebs** (*dict*) –

Parameters used to automatically set up EBS volumes when the instance is launched.

· **SnapshotId** (*string*) –

The ID of the snapshot.

· **VolumeSize** (*integer*) –

The size of the volume, in GiB.

Constraints: 1–1024 for standard volumes, 1–16384 for `gp2` volumes, and 4–16384 for `io1` volumes. If you specify a snapshot, the volume size must be equal to or larger than the snapshot size.

Default: If you’re creating the volume from a snapshot and don’t specify a volume size, the default is the snapshot size.

· **DeleteOnTermination** (*boolean*) –

Indicates whether the EBS volume is deleted on instance termination.

· **VolumeType** (*string*) –

The volume type. `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, and `standard` for Magnetic volumes.

Default: `standard`

· **Iops** (*integer*) –

The number of I/O operations per second (IOPS) that the volume supports. For Provisioned IOPS (SSD) volumes, this represents the number of IOPS that are provisioned for the volume. For General Purpose (SSD) volumes, this represents the baseline performance of the volume and the rate at which the volume accumulates I/O credits for bursting. For more information on General Purpose (SSD) baseline performance, I/O credits, and bursting, see [Amazon EBS Volume Types](#) in the *Amazon Elastic Compute Cloud User Guide*.

Constraint: Range is 100 to 20000 for Provisioned IOPS (SSD) volumes and 3 to 10000 for General Purpose (SSD) volumes.

Condition: This parameter is required for requests to create `io1` volumes; it is not used in requests to create standard or `gp2` volumes.

- **Encrypted** (*boolean*) –

Indicates whether the EBS volume is encrypted. Encrypted Amazon EBS volumes may only be attached to instances that support Amazon EBS encryption.

- * **NoDevice** (*string*) –

Suppresses the specified device included in the block device mapping of the AMI.

Return type `ec2.Image`

Returns Image resource

create_tags (***kwargs*)

Adds or overwrites one or more tags for the specified Amazon EC2 resource or resources. Each resource can have a maximum of 10 tags. Each tag consists of a key and optional value. Tag keys must be unique per resource.

For more information about tags, see [Tagging Your Resources](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
tag = instance.create_tags(
    DryRun=True|False,
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Tags** (*list*) – **[REQUIRED]**

One or more tags. The `value` parameter is required, but if you don't want the tag to have a value, specify the parameter with no value, and we set the value to an empty string.

– (*dict*) –

Describes a tag.

- * **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

- * **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

Return type `list(ec2.Tag)`

Returns A list of Tag resources

describe_attribute (***kwargs*)

Describes the specified attribute of the specified instance. You can specify only one attribute at a time. Valid attribute values are: `instanceType` | `kernel` | `ramdisk` | `userData` | `disableApiTermination` | `instanceInitiatedShutdownBehavior` | `rootDeviceName` | `blockDeviceMapping` | `productCodes` | `sourceDestCheck` | `groupSet` | `efsOptimized` | `sriovNetSupport`

Request Syntax

```
response = instance.describe_attribute(
    DryRun=True|False,
    Attribute='instanceType'|'kernel'|'ramdisk'|'userData'|'disableApiTermination'|'instanceInitiatedShutdownBehavior'|'rootDeviceName'|'blockDeviceMapping'|'productCodes'|'sourceDestCheck'|'groupSet'|'efsOptimized'|'sriovNetSupport'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Attribute** (*string*) – **[REQUIRED]**

The instance attribute.

Return type `dict`

Returns

Response Syntax

```
{
  'InstanceId': 'string',
  'InstanceType': {
    'Value': 'string'
  },
  'KernelId': {
    'Value': 'string'
  },
  'RamdiskId': {
    'Value': 'string'
  },
  'UserData': {
    'Value': 'string'
  },
  'DisableApiTermination': {
    'Value': True|False
  },
  'InstanceInitiatedShutdownBehavior': {
    'Value': 'string'
  },
  'RootDeviceName': {
    'Value': 'string'
  },
  'BlockDeviceMappings': [
```

```

    {
        'DeviceName': 'string',
        'Ebs': {
            'VolumeId': 'string',
            'Status': 'attaching'|'attached'|'detaching'|'detached',
            'AttachTime': datetime(2015, 1, 1),
            'DeleteOnTermination': True|False
        },
    },
],
'ProductCodes': [
    {
        'ProductCodeId': 'string',
        'ProductCodeType': 'devpay'|'marketplace'
    },
],
'EbsOptimized': {
    'Value': True|False
},
'SriovNetSupport': {
    'Value': 'string'
},
'SourceDestCheck': {
    'Value': True|False
},
'Groups': [
    {
        'GroupName': 'string',
        'GroupId': 'string'
    },
]
}

```

Response Structure

- (*dict*) –

Information about the instance attribute.

- **InstanceId** (*string*) –
The ID of the instance.
- **InstanceType** (*dict*) –
The instance type.
* **Value** (*string*) –
Valid values are case-sensitive and vary by action.
- **KernelId** (*dict*) –
The kernel ID.
* **Value** (*string*) –
Valid values are case-sensitive and vary by action.
- **RamdiskId** (*dict*) –
The RAM disk ID.
* **Value** (*string*) –
Valid values are case-sensitive and vary by action.
- **UserData** (*dict*) –
The Base64-encoded MIME user data.

- * **Value** (*string*) –

Valid values are case-sensitive and vary by action.

- **DisableApiTermination** (*dict*) –

If the value is `true`, you can't terminate the instance through the Amazon EC2 console, CLI, or API; otherwise, you can.

- * **Value** (*boolean*) –

Valid values are `true` or `false`.

- **InstanceInitiatedShutdownBehavior** (*dict*) –

Indicates whether an instance stops or terminates when you initiate shutdown from the instance (using the operating system command for system shutdown).

- * **Value** (*string*) –

Valid values are case-sensitive and vary by action.

- **RootDeviceName** (*dict*) –

The name of the root device (for example, `/dev/sda1` or `/dev/xvda`).

- * **Value** (*string*) –

Valid values are case-sensitive and vary by action.

- **BlockDeviceMappings** (*list*) –

The block device mapping of the instance.

- * (*dict*) –

Describes a block device mapping.

- **DeviceName** (*string*) –

The device name exposed to the instance (for example, `/dev/sdh` or `xvdh`).

- **Ebs** (*dict*) –

Parameters used to automatically set up EBS volumes when the instance is launched.

- **VolumeId** (*string*) –

The ID of the EBS volume.

- **Status** (*string*) –

The attachment state.

- **AttachTime** (*datetime*) –

The time stamp when the attachment initiated.

- **DeleteOnTermination** (*boolean*) –

Indicates whether the volume is deleted on instance termination.

- **ProductCodes** (*list*) –

A list of product codes.

- * (*dict*) –

Describes a product code.

- **ProductCodeId** (*string*) –

The product code.

- **ProductCodeType** (*string*) –
The type of product code.
- **EbsOptimized** (*dict*) –
Indicates whether the instance is optimized for EBS I/O.
* **Value** (*boolean*) –
Valid values are `true` or `false`.
- **SriovNetSupport** (*dict*) –
The value to use for a resource attribute.
* **Value** (*string*) –
Valid values are case-sensitive and vary by action.
- **SourceDestCheck** (*dict*) –
Indicates whether source/destination checking is enabled. A value of `true` means checking is enabled, and `false` means checking is disabled. This value must be `false` for a NAT instance to perform NAT.
* **Value** (*boolean*) –
Valid values are `true` or `false`.
- **Groups** (*list*) –
The security groups associated with the instance.
* (*dict*) –
Describes a security group.
 - **GroupName** (*string*) –
The name of the security group.
 - **GroupId** (*string*) –
The ID of the security group.

detach_classic_link_vpc (***kwargs*)

Unlinks (detaches) a linked EC2-Classical instance from a VPC. After the instance has been unlinked, the VPC security groups are no longer associated with it. An instance is automatically unlinked from a VPC when it's stopped.

Request Syntax

```
response = instance.detach_classic_link_vpc(
    DryRun=True|False,
    VpcId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcId** (*string*) – **[REQUIRED]**
The ID of the VPC to which the instance is linked.

Return type dict

Returns

Response Syntax

```
{
    'Return': True|False
}
```

Response Structure

- (dict) –
 - **Return** (boolean) –

Returns `true` if the request succeeds; otherwise, it returns an error.

`detach_volume` (**kwargs)

Detaches an EBS volume from an instance. Make sure to unmount any file systems on the device within your operating system before detaching the volume. Failure to do so results in the volume being stuck in a busy state while detaching.

If an Amazon EBS volume is the root device of an instance, it can't be detached while the instance is running. To detach the root volume, stop the instance first.

When a volume with an AWS Marketplace product code is detached from an instance, the product code is no longer associated with the instance.

For more information, see [Detaching an Amazon EBS Volume](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = instance.detach_volume(
    DryRun=True|False,
    VolumeId='string',
    Device='string',
    Force=True|False
)
```

Parameters

- **DryRun** (boolean) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VolumeId** (string) – **[REQUIRED]**
The ID of the volume.
- **Device** (string) – The device name.
- **Force** (boolean) – Forces detachment if the previous detachment attempt did not occur cleanly (for example, logging into an instance, unmounting the volume, and detaching normally). This option can lead to data loss or a corrupted file system. Use this option only as a last resort to detach a volume from a failed instance. The instance won't have an opportunity to flush file system caches or file system metadata. If you use this option, you must perform file system check and repair procedures.

Return type dict

Returns

Response Syntax

```
{
    'VolumeId': 'string',
    'InstanceId': 'string',
    'Device': 'string',
}
```

```

        'State': 'attaching'|'attached'|'detaching'|'detached',
        'AttachTime': datetime(2015, 1, 1),
        'DeleteOnTermination': True|False
    }

```

Response Structure

- *(dict)* –

Information about the volume attachment.

- **VolumeId** (*string*) –

The ID of the volume.

- **InstanceId** (*string*) –

The ID of the instance.

- **Device** (*string*) –

The device name.

- **State** (*string*) –

The attachment state of the volume.

- **AttachTime** (*datetime*) –

The time stamp when the attachment initiated.

- **DeleteOnTermination** (*boolean*) –

Indicates whether the EBS volume is deleted on instance termination.

load()

Calls `ec2.Client.describe_instances()` to update the attributes of the Instance resource

Request Syntax

```
instance.load()
```

Returns None

modify_attribute(**kwargs)

Modifies the specified attribute of the specified instance. You can specify only one attribute at a time.

To modify some attributes, the instance must be stopped. For more information, see [Modifying Attributes of a Stopped Instance](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```

response = instance.modify_attribute(
    DryRun=True|False,
    Attribute='instanceType'|'kernel'|'ramdisk'|'userData'|'disableApiTermination'|'instance
    Value='string',
    BlockDeviceMappings=[
        {
            'DeviceName': 'string',
            'Ebs': {
                'VolumeId': 'string',
                'DeleteOnTermination': True|False
            },
            'VirtualName': 'string',
            'NoDevice': 'string'
        },
    ],
    SourceDestCheck={

```

```
        'Value': True|False
    },
    DisableApiTermination={
        'Value': True|False
    },
    InstanceType={
        'Value': 'string'
    },
    Kernel={
        'Value': 'string'
    },
    Ramdisk={
        'Value': 'string'
    },
    UserData={
        'Value': b'bytes'
    },
    InstanceInitiatedShutdownBehavior={
        'Value': 'string'
    },
    Groups=[
        'string',
    ],
    EbsOptimized={
        'Value': True|False
    },
    SrioNetSupport={
        'Value': 'string'
    }
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Attribute** (*string*) – The name of the attribute.
- **Value** (*string*) – A new value for the attribute. Use only with the `kernel`, `ramdisk`, `userData`, `disableApiTermination`, or `instanceInitiatedShutdownBehavior` attribute.
- **BlockDeviceMappings** (*list*) – Modifies the `DeleteOnTermination` attribute for volumes that are currently attached. The volume must be owned by the caller. If no value is specified for `DeleteOnTermination`, the default is `true` and the volume is deleted when the instance is terminated.

To add instance store volumes to an Amazon EBS-backed instance, you must add them when you launch the instance. For more information, see [Updating the Block Device Mapping when Launching an Instance](#) in the *Amazon Elastic Compute Cloud User Guide*.

– (*dict*) –

Describes a block device mapping entry.

- * **DeviceName** (*string*) –

The device name exposed to the instance (for example, `/dev/sdh` or `xvdh`).

- * **Ebs** (*dict*) –

Parameters used to automatically set up EBS volumes when the instance is launched.

- **VolumeId** (*string*) –

The ID of the EBS volume.

- **DeleteOnTermination** (*boolean*) –

Indicates whether the volume is deleted on instance termination.

- * **VirtualName** (*string*) –

The virtual device name.

- * **NoDevice** (*string*) –

suppress the specified device included in the block device mapping.

- **SourceDestCheck** (*dict*) – Specifies whether source/destination checking is enabled. A value of `true` means that checking is enabled, and `false` means checking is disabled. This value must be `false` for a NAT instance to perform NAT.

- **Value** (*boolean*) –

Valid values are `true` or `false`.

- **DisableApiTermination** (*dict*) – If the value is `true`, you can't terminate the instance using the Amazon EC2 console, CLI, or API; otherwise, you can. You cannot use this parameter for Spot Instances.

- **Value** (*boolean*) –

Valid values are `true` or `false`.

- **InstanceType** (*dict*) – Changes the instance type to the specified value. For more information, see [Instance Types](#). If the instance type is not valid, the error returned is `InvalidInstanceAttributeValue`.

- **Value** (*string*) –

Valid values are case-sensitive and vary by action.

- **Kernel** (*dict*) – Changes the instance's kernel to the specified value. We recommend that you use PV-GRUB instead of kernels and RAM disks. For more information, see [PV-GRUB](#).

- **Value** (*string*) –

Valid values are case-sensitive and vary by action.

- **Ramdisk** (*dict*) – Changes the instance's RAM disk to the specified value. We recommend that you use PV-GRUB instead of kernels and RAM disks. For more information, see [PV-GRUB](#).

- **Value** (*string*) –

Valid values are case-sensitive and vary by action.

- **UserData** (*dict*) – Changes the instance's user data to the specified value.

- **Value** (*bytes*) –

- **InstanceInitiatedShutdownBehavior** (*dict*) – Specifies whether an instance stops or terminates when you initiate shutdown from the instance (using the operating system command for system shutdown).

- **Value** (*string*) –

Valid values are case-sensitive and vary by action.

- **Groups** (*list*) – [EC2-VPC] Changes the security groups of the instance. You must specify at least one security group, even if it's just the default security group for the VPC. You must specify the security group ID, not the security group name.

- (*string*) –

- **EbsOptimized** (*dict*) – Specifies whether the instance is optimized for EBS I/O. This optimization provides dedicated throughput to Amazon EBS and an optimized configuration stack to provide optimal EBS I/O performance. This optimization isn't available with all instance types. Additional usage charges apply when using an EBS Optimized instance.

- **Value** (*boolean*) –

Valid values are `true` or `false`.

- **SriovNetSupport** (*dict*) – Set to `simple` to enable enhanced networking for the instance.

There is no way to disable enhanced networking at this time.

This option is supported only for HVM instances. Specifying this option with a PV instance can make it unreachable.

- **Value** (*string*) –

Valid values are case-sensitive and vary by action.

Returns None

monitor (***kwargs*)

Enables monitoring for a running instance. For more information about monitoring instances, see [Monitoring Your Instances and Volumes](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = instance.monitor(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns

Response Syntax

```
{
  'InstanceMonitorings': [
    {
      'InstanceId': 'string',
      'Monitoring': {
        'State': 'disabled'|'disabling'|'enabled'|'pending'
      }
    },
  ]
}
```

Response Structure

- (*dict*) –
 - **InstanceMonitorings** (*list*) –

Monitoring information for one or more instances.

- * (*dict*) –

Describes the monitoring information of the instance.

- **InstanceId** (*string*) –
The ID of the instance.
- **Monitoring** (*dict*) –
The monitoring information.
- **State** (*string*) –
Indicates whether monitoring is enabled for the instance.

password_data (***kwargs*)

Retrieves the encrypted administrator password for an instance running Windows.

The Windows password is generated at boot if the EC2Config service plugin, `Ec2SetPassword`, is enabled. This usually only happens the first time an AMI is launched, and then `Ec2SetPassword` is automatically disabled. The password is not generated for rebundled AMIs unless `Ec2SetPassword` is enabled before bundling.

The password is encrypted using the key pair that you specified when you launched the instance. You must provide the corresponding key pair file.

Password generation and encryption takes a few moments. We recommend that you wait up to 15 minutes after launching an instance before trying to retrieve the generated password.

Request Syntax

```
response = instance.password_data(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns

Response Syntax

```
{
    'InstanceId': 'string',
    'Timestamp': datetime(2015, 1, 1),
    'PasswordData': 'string'
}
```

Response Structure

- (*dict*) –
 - **InstanceId** (*string*) –
The ID of the Windows instance.
 - **Timestamp** (*datetime*) –
The time the data was last updated.
 - **PasswordData** (*string*) –
The password of the instance.

reboot (***kwargs*)

Requests a reboot of one or more instances. This operation is asynchronous; it only queues a request

to reboot the specified instances. The operation succeeds if the instances are valid and belong to you. Requests to reboot terminated instances are ignored.

If a Linux/Unix instance does not cleanly shut down within four minutes, Amazon EC2 performs a hard reboot.

For more information about troubleshooting, see [Getting Console Output and Rebooting Instances](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = instance.reboot(  
    DryRun=True|False,  
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns `None`

reload()

Calls `ec2.Client.describe_instances()` to update the attributes of the Instance resource

Request Syntax

```
instance.reload()
```

Returns `None`

report_status (***kwargs*)

Submits feedback about the status of an instance. The instance must be in the `running` state. If your experience with the instance differs from the instance status returned by `DescribeInstanceState`, use `ReportInstanceState` to report your experience with the instance. Amazon EC2 collects this information to improve the accuracy of status checks.

Use of this action does not change the value returned by `DescribeInstanceState`.

Request Syntax

```
response = instance.report_status(  
    DryRun=True|False,  
    Status='ok'|'impaired',  
    StartTime=datetime(2015, 1, 1),  
    EndTime=datetime(2015, 1, 1),  
    ReasonCodes=[  
        'instance-stuck-in-state'|'unresponsive'|'not-accepting-credentials'|'password-not-  
    ],  
    Description='string'  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Status** (*string*) – **[REQUIRED]**

The status of all instances listed.

- **StartTime** (*datetime*) – The time at which the reported instance health state began.
- **EndTime** (*datetime*) – The time at which the reported instance health state ended.
- **ReasonCodes** (*list*) – [REQUIRED]

One or more reason codes that describes the health state of your instance.

- `instance-stuck-in-state` : My instance is stuck in a state.
- `unresponsive` : My instance is unresponsive.
- `not-accepting-credentials` : My instance is not accepting my credentials.
- `password-not-available` : A password is not available for my instance.
- `performance-network` : My instance is experiencing performance problems which I believe are network related.
- `performance-instance-store` : My instance is experiencing performance problems which I believe are related to the instance stores.
- `performance-ebs-volume` : My instance is experiencing performance problems which I believe are related to an EBS volume.
- `performance-other` : My instance is experiencing performance problems.
- `other` : [explain using the description parameter]
- (*string*) –

- **Description** (*string*) – Descriptive text about the health state of your instance.

Returns None

reset_attribute (***kwargs*)

Resets an attribute of an instance to its default value. To reset the `kernel` or `ramdisk`, the instance must be in a stopped state. To reset the `SourceDestCheck`, the instance can be either running or stopped.

The `SourceDestCheck` attribute controls whether source/destination checking is enabled. The default value is `true`, which means checking is enabled. This value must be `false` for a NAT instance to perform NAT. For more information, see [NAT Instances](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = instance.reset_attribute(
    DryRun=True|False,
    Attribute='instanceType'|'kernel'|'ramdisk'|'userData'|'disableApiTermination'|'instanc
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Attribute** (*string*) – [REQUIRED]

The attribute to reset.

Returns None

reset_kernel (***kwargs*)

Resets an attribute of an instance to its default value. To reset the `kernel` or `ramdisk`, the instance must be in a stopped state. To reset the `SourceDestCheck`, the instance can be either running or stopped.

The `SourceDestCheck` attribute controls whether source/destination checking is enabled. The default value is `true`, which means checking is enabled. This value must be `false` for a NAT instance to perform NAT. For more information, see [NAT Instances](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = instance.reset_kernel(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns None

reset_ramdisk (**kwargs)

Resets an attribute of an instance to its default value. To reset the `kernel` or `ramdisk`, the instance must be in a stopped state. To reset the `SourceDestCheck`, the instance can be either running or stopped.

The `SourceDestCheck` attribute controls whether source/destination checking is enabled. The default value is `true`, which means checking is enabled. This value must be `false` for a NAT instance to perform NAT. For more information, see [NAT Instances](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = instance.reset_ramdisk(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns None

reset_source_dest_check (**kwargs)

Resets an attribute of an instance to its default value. To reset the `kernel` or `ramdisk`, the instance must be in a stopped state. To reset the `SourceDestCheck`, the instance can be either running or stopped.

The `SourceDestCheck` attribute controls whether source/destination checking is enabled. The default value is `true`, which means checking is enabled. This value must be `false` for a NAT instance to perform NAT. For more information, see [NAT Instances](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = instance.reset_source_dest_check(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns `None`

start (***kwargs*)

Starts an Amazon EBS-backed AMI that you’ve previously stopped.

Instances that use Amazon EBS volumes as their root devices can be quickly stopped and started. When an instance is stopped, the compute resources are released and you are not billed for hourly instance usage. However, your root partition Amazon EBS volume remains, continues to persist your data, and you are charged for Amazon EBS volume usage. You can restart your instance at any time. Each time you transition an instance from stopped to started, Amazon EC2 charges a full instance hour, even if transitions happen multiple times within a single hour.

Before stopping an instance, make sure it is in a state from which it can be restarted. Stopping an instance does not preserve data stored in RAM.

Performing this operation on an instance that uses an instance store as its root device returns an error.

For more information, see [Stopping Instances](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = instance.start(
    AdditionalInfo='string',
    DryRun=True|False
)
```

Parameters

- **AdditionalInfo** (*string*) – Reserved.
- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type `dict`

Returns

Response Syntax

```
{
  'StartingInstances': [
    {
      'InstanceId': 'string',
      'CurrentState': {
        'Code': 123,
        'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stopping',
      },
      'PreviousState': {
        'Code': 123,
        'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stopping',
      },
    },
  ],
}
```

Response Structure

- (*dict*) –

– **StartingInstances** (*list*) –

Information about one or more started instances.

* (*dict*) –

Describes an instance state change.

· **InstanceId** (*string*) –

The ID of the instance.

· **CurrentState** (*dict*) –

The current state of the instance.

· **Code** (*integer*) –

The low byte represents the state. The high byte is an opaque internal value and should be ignored.

· 0: pending

· 16: running

· 32: shutting-down

· 48: terminated

· 64: stopping

· 80: stopped

· **Name** (*string*) –

The current state of the instance.

· **PreviousState** (*dict*) –

The previous state of the instance.

· **Code** (*integer*) –

The low byte represents the state. The high byte is an opaque internal value and should be ignored.

· 0: pending

· 16: running

· 32: shutting-down

· 48: terminated

· 64: stopping

· 80: stopped

· **Name** (*string*) –

The current state of the instance.

stop (***kwargs*)

Stops an Amazon EBS-backed instance. Each time you transition an instance from stopped to started, Amazon EC2 charges a full instance hour, even if transitions happen multiple times within a single hour.

You can't start or stop Spot Instances.

Instances that use Amazon EBS volumes as their root devices can be quickly stopped and started. When an instance is stopped, the compute resources are released and you are not billed for hourly instance usage. However, your root partition Amazon EBS volume remains, continues to persist your data, and you are charged for Amazon EBS volume usage. You can restart your instance at any time.

Before stopping an instance, make sure it is in a state from which it can be restarted. Stopping an instance does not preserve data stored in RAM.

Performing this operation on an instance that uses an instance store as its root device returns an error.

You can stop, start, and terminate EBS-backed instances. You can only terminate instance store-backed instances. What happens to an instance differs if you stop it or terminate it. For example, when you stop an instance, the root device and any other devices attached to the instance persist. When you terminate

an instance, the root device and any other devices attached during the instance launch are automatically deleted. For more information about the differences between stopping and terminating instances, see [Instance Lifecycle](#) in the *Amazon Elastic Compute Cloud User Guide*.

For more information about troubleshooting, see [Troubleshooting Stopping Your Instance](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = instance.stop(
    DryRun=True|False,
    Force=True|False
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Force** (*boolean*) – Forces the instances to stop. The instances do not have an opportunity to flush file system caches or file system metadata. If you use this option, you must perform file system check and repair procedures. This option is not recommended for Windows instances.

Default: false

Return type dict

Returns

Response Syntax

```
{
  'StoppingInstances': [
    {
      'InstanceId': 'string',
      'CurrentState': {
        'Code': 123,
        'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stopping',
      },
      'PreviousState': {
        'Code': 123,
        'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stopping',
      },
    },
  ],
}
```

Response Structure

- (*dict*) –
 - **StoppingInstances** (*list*) –

Information about one or more stopped instances.

 - * (*dict*) –

Describes an instance state change.

 - **InstanceId** (*string*) –

The ID of the instance.
 - **CurrentState** (*dict*) –

The current state of the instance.

- **Code** (*integer*) –

The low byte represents the state. The high byte is an opaque internal value and should be ignored.

- 0 : pending
- 16 : running
- 32 : shutting-down
- 48 : terminated
- 64 : stopping
- 80 : stopped
- **Name** (*string*) –

The current state of the instance.

- **PreviousState** (*dict*) –

The previous state of the instance.

- **Code** (*integer*) –

The low byte represents the state. The high byte is an opaque internal value and should be ignored.

- 0 : pending
- 16 : running
- 32 : shutting-down
- 48 : terminated
- 64 : stopping
- 80 : stopped
- **Name** (*string*) –

The current state of the instance.

terminate (***kwargs*)

Shuts down one or more instances. This operation is idempotent; if you terminate an instance more than once, each call succeeds.

Terminated instances remain visible after termination (for approximately one hour).

By default, Amazon EC2 deletes all EBS volumes that were attached when the instance launched. Volumes attached after instance launch continue running.

You can stop, start, and terminate EBS-backed instances. You can only terminate instance store-backed instances. What happens to an instance differs if you stop it or terminate it. For example, when you stop an instance, the root device and any other devices attached to the instance persist. When you terminate an instance, any attached EBS volumes with the `DeleteOnTermination` block device mapping parameter set to `true` are automatically deleted. For more information about the differences between stopping and terminating instances, see [Instance Lifecycle](#) in the *Amazon Elastic Compute Cloud User Guide*.

For more information about troubleshooting, see [Troubleshooting Terminating Your Instance](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = instance.terminate(  
    DryRun=True|False,  
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns

Response Syntax

```

{
    'TerminatingInstances': [
        {
            'InstanceId': 'string',
            'CurrentState': {
                'Code': 123,
                'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stopping',
            },
            'PreviousState': {
                'Code': 123,
                'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stopping',
            },
        },
    ],
}

```

Response Structure

- (*dict*) –
 - **TerminatingInstances** (*list*) –

Information about one or more terminated instances.

* (*dict*) –

Describes an instance state change.

· **InstanceId** (*string*) –

The ID of the instance.

· **CurrentState** (*dict*) –

The current state of the instance.

· **Code** (*integer*) –

The low byte represents the state. The high byte is an opaque internal value and should be ignored.

· 0: pending

· 16: running

· 32: shutting-down

· 48: terminated

· 64: stopping

· 80: stopped

· **Name** (*string*) –

The current state of the instance.

· **PreviousState** (*dict*) –

The previous state of the instance.

· **Code** (*integer*) –

The low byte represents the state. The high byte is an opaque internal value and should be ignored.

· 0: pending

· 16: running

· 32: shutting-down

· 48: terminated

· 64: stopping

- 80 : stopped
- **Name** (*string*) –
The current state of the instance.

unmonitor (**kwargs)

Disables monitoring for a running instance. For more information about monitoring instances, see [Monitoring Your Instances and Volumes](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = instance.unmonitor(  
    DryRun=True|False,  
  
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns

Response Syntax

```
{  
    'InstanceMonitorings': [  
        {  
            'InstanceId': 'string',  
            'Monitoring': {  
                'State': 'disabled'|'disabling'|'enabled'|'pending'  
            }  
        },  
    ],  
}
```

Response Structure

- (*dict*) –
 - **InstanceMonitorings** (*list*) –

Monitoring information for one or more instances.

- * (*dict*) –

Describes the monitoring information of the instance.

- **InstanceId** (*string*) –
The ID of the instance.
- **Monitoring** (*dict*) –
The monitoring information.
- **State** (*string*) –

Indicates whether monitoring is enabled for the instance.

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

volumes

all()

Creates an iterable of all Volume resources in the collection.

Request Syntax

```
volume_iterator = instance.volumes.all()
```

Return type list(ec2.Volume)

Returns A list of Volume resources

filter(kwargs)**

Creates an iterable of all Volume resources in the collection filtered by kwargs passed to method.

Request Syntax

```
volume_iterator = instance.volumes.filter(
    DryRun=True|False,
    VolumeIds=[
        'string',
    ],
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VolumeIds** (*list*) – One or more volume IDs.
 - (*string*) –
- **NextToken** (*string*) – The `NextToken` value returned from a previous paginated `DescribeVolumes` request where `MaxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `NextToken` value. This value is `null` when there are no more results to return.
- **MaxResults** (*integer*) – The maximum number of volume results returned by `DescribeVolumes` in paginated output. When this parameter is used, `DescribeVolumes` only returns `MaxResults` results in a single page along with a `NextToken` response element. The remaining results of the initial request can be seen by sending another `DescribeVolumes` request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. If this parameter is not used, then `DescribeVolumes` returns all results. You cannot specify this parameter and the volume IDs parameter in the same request.

Return type list(ec2.Volume)

Returns A list of Volume resources

limit(kwargs)**

Creates an iterable up to a specified amount of Volume resources in the collection.

Request Syntax

```
volume_iterator = instance.volumes.limit(
    count=123
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(ec2.Volume)`

Returns A list of Volume resources

page_size (***kwargs*)

Creates an iterable of all Volume resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
volume_iterator = instance.volumes.page_size(  
    count=123  
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(ec2.Volume)`

Returns A list of Volume resources

Waiters

Waiters provide an interface to wait for a resource to reach a specific state. For more information about waiters refer to the [Resources Introduction Guide](#).

wait_until_exists (***kwargs*)

Waits until this Instance is exists. This method calls `EC2.Waiter.instance_exists.wait()` which polls `EC2.Client.describe_instances()` every 5 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
instance.wait_until_exists(  
    DryRun=True|False,  
    Filters=[  
        {  
            'Name': 'string',  
            'Values': [  
                'string',  
            ]  
        },  
    ],  
    NextToken='string',  
    MaxResults=123  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Filters** (*list*) – One or more filters.
 - `architecture` - The instance architecture (`i386|x86_64`).
 - `availability-zone` - The Availability Zone of the instance.
 - `block-device-mapping.attach-time` - The attach time for an EBS volume mapped to the instance, for example, `2010-09-15T17:15:20.000Z`.
 - `block-device-mapping.delete-on-termination` - A Boolean that indicates whether the EBS volume is deleted on instance termination.

- `block-device-mapping.device-name` - The device name for the EBS volume (for example, `/dev/sdh` or `xvdh`).
- `block-device-mapping.status` - The status for the EBS volume (`attaching` | `attached` | `detaching` | `detached`).
- `block-device-mapping.volume-id` - The volume ID of the EBS volume.
- `client-token` - The idempotency token you provided when you launched the instance.
- `dns-name` - The public DNS name of the instance.
- `group-id` - The ID of the security group for the instance. EC2-Classic only.
- `group-name` - The name of the security group for the instance. EC2-Classic only.
- `hypervisor` - The hypervisor type of the instance (`ovm` | `xen`).
- `iam-instance-profile.arn` - The instance profile associated with the instance. Specified as an ARN.
- `image-id` - The ID of the image used to launch the instance.
- `instance-id` - The ID of the instance.
- `instance-lifecycle` - Indicates whether this is a Spot Instance (`spot`).
- `instance-state-code` - The state of the instance, as a 16-bit unsigned integer. The high byte is an opaque internal value and should be ignored. The low byte is set based on the state represented. The valid values are: 0 (pending), 16 (running), 32 (shutting-down), 48 (terminated), 64 (stopping), and 80 (stopped).
- `instance-state-name` - The state of the instance (`pending` | `running` | `shutting-down` | `terminated` | `stopping` | `stopped`).
- `instance-type` - The type of instance (for example, `t2.micro`).
- `instance.group-id` - The ID of the security group for the instance.
- `instance.group-name` - The name of the security group for the instance.
- `ip-address` - The public IP address of the instance.
- `kernel-id` - The kernel ID.
- `key-name` - The name of the key pair used when the instance was launched.
- `launch-index` - When launching multiple instances, this is the index for the instance in the launch group (for example, 0, 1, 2, and so on).
- `launch-time` - The time when the instance was launched.
- `monitoring-state` - Indicates whether monitoring is enabled for the instance (`disabled` | `enabled`).
- `owner-id` - The AWS account ID of the instance owner.
- `placement-group-name` - The name of the placement group for the instance.
- `platform` - The platform. Use windows if you have Windows instances; otherwise, leave blank.
- `private-dns-name` - The private DNS name of the instance.
- `private-ip-address` - The private IP address of the instance.
- `product-code` - The product code associated with the AMI used to launch the instance.
- `product-code.type` - The type of product code (`devpay` | `marketplace`).
- `ramdisk-id` - The RAM disk ID.
- `reason` - The reason for the current state of the instance (for example,

shows “User Initiated [date]” when you stop or terminate the instance). Similar to the `state-reason-code` filter.

- `requester-id` - The ID of the entity that launched the instance on your behalf (for example, AWS Management Console, Auto Scaling, and so on).
- `reservation-id` - The ID of the instance’s reservation. A reservation ID is created any time you launch an instance. A reservation ID has a one-to-one relationship with an instance launch request, but can be associated with more than one instance if you launch multiple instances using the same launch request. For example, if you launch one instance, you’ll get one reservation ID. If you launch ten instances using the same launch request, you’ll also get one reservation ID.
- `root-device-name` - The name of the root device for the instance (for example, `/dev/sda1` or `/dev/xvda`).
- `root-device-type` - The type of root device that the instance uses (`ebs` | `instance-store`).
- `source-dest-check` - Indicates whether the instance performs source/destination checking. A value of `true` means that checking is enabled, and `false` means checking is disabled. The value must be `false` for the instance to perform network address translation (NAT) in your VPC.
- `spot-instance-request-id` - The ID of the Spot Instance request.
- `state-reason-code` - The reason code for the state change.
- `state-reason-message` - A message that describes the state change.
- `subnet-id` - The ID of the subnet for the instance.
- `tag :key =*value*` - The key/value combination of a tag assigned to the resource, where `tag :key` is the tag’s key.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag’s value is), and the tag value `X` (regardless of what the tag’s key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `tenancy` - The tenancy of an instance (`dedicated` | `default`).
- `virtualization-type` - The virtualization type of the instance (`paravirtual` | `hvm`).
- `vpc-id` - The ID of the VPC that the instance is running in.
- `network-interface.description` - The description of the network interface.
- `network-interface.subnet-id` - The ID of the subnet for the network interface.
- `network-interface.vpc-id` - The ID of the VPC for the network interface.
- `network-interface.network-interface.id` - The ID of the network interface.
- `network-interface.owner-id` - The ID of the owner of the network interface.
- `network-interface.availability-zone` - The Availability Zone for the network interface.
- `network-interface.requester-id` - The requester ID for the network interface.
- `network-interface.requester-managed` - Indicates whether

the network interface is being managed by AWS.

- `network-interface.status` - The status of the network interface (available | in-use).
- `network-interface.mac-address` - The MAC address of the network interface.
- `network-interface.private-dns-name` - The private DNS name of the network interface.
- `network-interface.source-dest-check` - Whether the network interface performs source/destination checking. A value of `true` means checking is enabled, and `false` means checking is disabled. The value must be `false` for the network interface to perform network address translation (NAT) in your VPC.
- `network-interface.group-id` - The ID of a security group associated with the network interface.
- `network-interface.group-name` - The name of a security group associated with the network interface.
- `network-interface.attachment.attachment-id` - The ID of the interface attachment.
- `network-interface.attachment.instance-id` - The ID of the instance to which the network interface is attached.
- `network-interface.attachment.instance-owner-id` - The owner ID of the instance to which the network interface is attached.
- `network-interface.addresses.private-ip-address` - The private IP address associated with the network interface.
- `network-interface.attachment.device-index` - The device index to which the network interface is attached.
- `network-interface.attachment.status` - The status of the attachment (attaching | attached | detaching | detached).
- `network-interface.attachment.attach-time` - The time that the network interface was attached to an instance.
- `network-interface.attachment.delete-on-termination` - Specifies whether the attachment is deleted when an instance is terminated.
- `network-interface.addresses.primary` - Specifies whether the IP address of the network interface is the primary private IP address.
- `network-interface.addresses.association.public-ip` - The ID of the association of an Elastic IP address with a network interface.
- `network-interface.addresses.association.ip-owner-id` - The owner ID of the private IP address associated with the network interface.
- `association.public-ip` - The address of the Elastic IP address bound to the network interface.
- `association.ip-owner-id` - The owner of the Elastic IP address associated with the network interface.
- `association.allocation-id` - The allocation ID returned when you allocated the Elastic IP address for your network interface.
- `association.association-id` - The association ID returned when the network interface was associated with an IP address.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

- **NextToken** (*string*) – The token to request the next page of results.
- **MaxResults** (*integer*) – The maximum number of results to return for the request in a single page. The remaining results of the initial request can be seen by sending another request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. You cannot specify this parameter and the `instance IDs` parameter in the same request.

Returns None

wait_until_running (***kwargs*)

Waits until this Instance is running. This method calls `EC2.Waiter.instance_running.wait()` which polls `EC2.Client.describe_instances()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
instance.wait_until_running(
    DryRun=True|False,
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Filters** (*list*) – One or more filters.
 - `architecture` - The instance architecture (`i386|x86_64`).
 - `availability-zone` - The Availability Zone of the instance.
 - `block-device-mapping.attach-time` - The attach time for an EBS volume mapped to the instance, for example, `2010-09-15T17:15:20.000Z`.
 - `block-device-mapping.delete-on-termination` - A Boolean that indicates whether the EBS volume is deleted on instance termination.
 - `block-device-mapping.device-name` - The device name for the EBS volume (for example, `/dev/sdh` or `xvdh`).
 - `block-device-mapping.status` - The status for the EBS volume (`attaching|attached|detaching|detached`).
 - `block-device-mapping.volume-id` - The volume ID of the EBS volume.

- `client-token` - The idempotency token you provided when you launched the instance.
- `dns-name` - The public DNS name of the instance.
- `group-id` - The ID of the security group for the instance. EC2-Classic only.
- `group-name` - The name of the security group for the instance. EC2-Classic only.
- `hypervisor` - The hypervisor type of the instance (`ovm` | `xen`).
- `iam-instance-profile.arn` - The instance profile associated with the instance. Specified as an ARN.
- `image-id` - The ID of the image used to launch the instance.
- `instance-id` - The ID of the instance.
- `instance-lifecycle` - Indicates whether this is a Spot Instance (`spot`).
- `instance-state-code` - The state of the instance, as a 16-bit unsigned integer. The high byte is an opaque internal value and should be ignored. The low byte is set based on the state represented. The valid values are: 0 (pending), 16 (running), 32 (shutting-down), 48 (terminated), 64 (stopping), and 80 (stopped).
- `instance-state-name` - The state of the instance (`pending` | `running` | `shutting-down` | `terminated` | `stopping` | `stopped`).
- `instance-type` - The type of instance (for example, `t2.micro`).
- `instance.group-id` - The ID of the security group for the instance.
- `instance.group-name` - The name of the security group for the instance.
- `ip-address` - The public IP address of the instance.
- `kernel-id` - The kernel ID.
- `key-name` - The name of the key pair used when the instance was launched.
- `launch-index` - When launching multiple instances, this is the index for the instance in the launch group (for example, 0, 1, 2, and so on).
- `launch-time` - The time when the instance was launched.
- `monitoring-state` - Indicates whether monitoring is enabled for the instance (`disabled` | `enabled`).
- `owner-id` - The AWS account ID of the instance owner.
- `placement-group-name` - The name of the placement group for the instance.
- `platform` - The platform. Use `windows` if you have Windows instances; otherwise, leave blank.
- `private-dns-name` - The private DNS name of the instance.
- `private-ip-address` - The private IP address of the instance.
- `product-code` - The product code associated with the AMI used to launch the instance.
- `product-code.type` - The type of product code (`devpay` | `marketplace`).
- `ramdisk-id` - The RAM disk ID.
- `reason` - The reason for the current state of the instance (for example, shows “User Initiated [date]” when you stop or terminate the instance). Similar to the state-reason-code filter.
- `requester-id` - The ID of the entity that launched the instance on your behalf (for example, AWS Management Console, Auto Scaling, and so on).
- `reservation-id` - The ID of the instance’s reservation. A reservation

ID is created any time you launch an instance. A reservation ID has a one-to-one relationship with an instance launch request, but can be associated with more than one instance if you launch multiple instances using the same launch request. For example, if you launch one instance, you'll get one reservation ID. If you launch ten instances using the same launch request, you'll also get one reservation ID.

- `root-device-name` - The name of the root device for the instance (for example, `/dev/sda1` or `/dev/xvda`).
- `root-device-type` - The type of root device that the instance uses (`ebs` | `instance-store`).
- `source-dest-check` - Indicates whether the instance performs source/destination checking. A value of `true` means that checking is enabled, and `false` means checking is disabled. The value must be `false` for the instance to perform network address translation (NAT) in your VPC.
- `spot-instance-request-id` - The ID of the Spot Instance request.
- `state-reason-code` - The reason code for the state change.
- `state-reason-message` - A message that describes the state change.
- `subnet-id` - The ID of the subnet for the instance.
- `tag :key =*value*` - The key/value combination of a tag assigned to the resource, where `tag :key` is the tag's key.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter "`tag-key=Purpose`" and the filter "`tag-value=X`", you get any resources assigned both the tag key `Purpose` (regardless of what the tag's value is), and the tag value `X` (regardless of what the tag's key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `tenancy` - The tenancy of an instance (`dedicated` | `default`).
- `virtualization-type` - The virtualization type of the instance (`paravirtual` | `hvm`).
- `vpc-id` - The ID of the VPC that the instance is running in.
- `network-interface.description` - The description of the network interface.
- `network-interface.subnet-id` - The ID of the subnet for the network interface.
- `network-interface.vpc-id` - The ID of the VPC for the network interface.
- `network-interface.network-interface.id` - The ID of the network interface.
- `network-interface.owner-id` - The ID of the owner of the network interface.
- `network-interface.availability-zone` - The Availability Zone for the network interface.
- `network-interface.requester-id` - The requester ID for the network interface.
- `network-interface.requester-managed` - Indicates whether the network interface is being managed by AWS.
- `network-interface.status` - The status of the network interface (`available` | `in-use`).
- `network-interface.mac-address` - The MAC address of the network interface.
- `network-interface-private-dns-name` - The private DNS

- name of the network interface.
- `network-interface.source-dest-check` - Whether the network interface performs source/destination checking. A value of `true` means checking is enabled, and `false` means checking is disabled. The value must be `false` for the network interface to perform network address translation (NAT) in your VPC.
- `network-interface.group-id` - The ID of a security group associated with the network interface.
- `network-interface.group-name` - The name of a security group associated with the network interface.
- `network-interface.attachment.attachment-id` - The ID of the interface attachment.
- `network-interface.attachment.instance-id` - The ID of the instance to which the network interface is attached.
- `network-interface.attachment.instance-owner-id` - The owner ID of the instance to which the network interface is attached.
- `network-interface.addresses.private-ip-address` - The private IP address associated with the network interface.
- `network-interface.attachment.device-index` - The device index to which the network interface is attached.
- `network-interface.attachment.status` - The status of the attachment (`attaching` | `attached` | `detaching` | `detached`).
- `network-interface.attachment.attach-time` - The time that the network interface was attached to an instance.
- `network-interface.attachment.delete-on-termination` - Specifies whether the attachment is deleted when an instance is terminated.
- `network-interface.addresses.primary` - Specifies whether the IP address of the network interface is the primary private IP address.
- `network-interface.addresses.association.public-ip` - The ID of the association of an Elastic IP address with a network interface.
- `network-interface.addresses.association.ip-owner-id` - The owner ID of the private IP address associated with the network interface.
- `association.public-ip` - The address of the Elastic IP address bound to the network interface.
- `association.ip-owner-id` - The owner of the Elastic IP address associated with the network interface.
- `association.allocation-id` - The allocation ID returned when you allocated the Elastic IP address for your network interface.
- `association.association-id` - The association ID returned when the network interface was associated with an IP address.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

- * **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

- * **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

- (*string*) -

- **NextToken** (*string*) – The token to request the next page of results.
- **MaxResults** (*integer*) – The maximum number of results to return for the request in a single page. The remaining results of the initial request can be seen by sending another request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. You cannot specify this parameter and the `instance IDs` parameter in the same request.

Returns None

wait_until_stopped (***kwargs*)

Waits until this Instance is stopped. This method calls `EC2.Waiter.instance_stopped.wait()` which polls `EC2.Client.describe_instances()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
instance.wait_until_stopped(
    DryRun=True|False,
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Filters** (*list*) – One or more filters.
 - `architecture` - The instance architecture (i386|x86_64).
 - `availability-zone` - The Availability Zone of the instance.
 - `block-device-mapping.attach-time` - The attach time for an EBS volume mapped to the instance, for example, 2010-09-15T17:15:20.000Z.
 - `block-device-mapping.delete-on-termination` - A Boolean that indicates whether the EBS volume is deleted on instance termination.
 - `block-device-mapping.device-name` - The device name for the EBS volume (for example, `/dev/sdh` or `xvdh`).
 - `block-device-mapping.status` - The status for the EBS volume (attaching|attached|detaching|detached).
 - `block-device-mapping.volume-id` - The volume ID of the EBS volume.
 - `client-token` - The idempotency token you provided when you launched the instance.
 - `dns-name` - The public DNS name of the instance.
 - `group-id` - The ID of the security group for the instance. EC2-Classic only.

- `group-name` - The name of the security group for the instance. EC2-Classic only.
- `hypervisor` - The hypervisor type of the instance (`ovm` | `xen`).
- `iam-instance-profile.arn` - The instance profile associated with the instance. Specified as an ARN.
- `image-id` - The ID of the image used to launch the instance.
- `instance-id` - The ID of the instance.
- `instance-lifecycle` - Indicates whether this is a Spot Instance (`spot`).
- `instance-state-code` - The state of the instance, as a 16-bit unsigned integer. The high byte is an opaque internal value and should be ignored. The low byte is set based on the state represented. The valid values are: 0 (pending), 16 (running), 32 (shutting-down), 48 (terminated), 64 (stopping), and 80 (stopped).
- `instance-state-name` - The state of the instance (`pending` | `running` | `shutting-down` | `terminated` | `stopping` | `stopped`).
- `instance-type` - The type of instance (for example, `t2.micro`).
- `instance.group-id` - The ID of the security group for the instance.
- `instance.group-name` - The name of the security group for the instance.
- `ip-address` - The public IP address of the instance.
- `kernel-id` - The kernel ID.
- `key-name` - The name of the key pair used when the instance was launched.
- `launch-index` - When launching multiple instances, this is the index for the instance in the launch group (for example, 0, 1, 2, and so on).
- `launch-time` - The time when the instance was launched.
- `monitoring-state` - Indicates whether monitoring is enabled for the instance (`disabled` | `enabled`).
- `owner-id` - The AWS account ID of the instance owner.
- `placement-group-name` - The name of the placement group for the instance.
- `platform` - The platform. Use `windows` if you have Windows instances; otherwise, leave blank.
- `private-dns-name` - The private DNS name of the instance.
- `private-ip-address` - The private IP address of the instance.
- `product-code` - The product code associated with the AMI used to launch the instance.
- `product-code.type` - The type of product code (`devpay` | `marketplace`).
- `ramdisk-id` - The RAM disk ID.
- `reason` - The reason for the current state of the instance (for example, shows “User Initiated [date]” when you stop or terminate the instance). Similar to the `state-reason-code` filter.
- `requester-id` - The ID of the entity that launched the instance on your behalf (for example, AWS Management Console, Auto Scaling, and so on).
- `reservation-id` - The ID of the instance’s reservation. A reservation ID is created any time you launch an instance. A reservation ID has a one-to-one relationship with an instance launch request, but can be associated with more than one instance if you launch multiple instances using the same launch request. For example, if you launch one instance, you’ll get one reservation ID. If you launch ten instances using the same launch

request, you'll also get one reservation ID.

- `root-device-name` - The name of the root device for the instance (for example, `/dev/sda1` or `/dev/xvda`).
- `root-device-type` - The type of root device that the instance uses (`ebs` | `instance-store`).
- `source-dest-check` - Indicates whether the instance performs source/destination checking. A value of `true` means that checking is enabled, and `false` means checking is disabled. The value must be `false` for the instance to perform network address translation (NAT) in your VPC.
- `spot-instance-request-id` - The ID of the Spot Instance request.
- `state-reason-code` - The reason code for the state change.
- `state-reason-message` - A message that describes the state change.
- `subnet-id` - The ID of the subnet for the instance.
- `tag :key =*value*` - The key/value combination of a tag assigned to the resource, where `tag :key` is the tag's key.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter "`tag-key=Purpose`" and the filter "`tag-value=X`", you get any resources assigned both the tag key `Purpose` (regardless of what the tag's value is), and the tag value `X` (regardless of what the tag's key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `tenancy` - The tenancy of an instance (`dedicated` | `default`).
- `virtualization-type` - The virtualization type of the instance (`paravirtual` | `hvm`).
- `vpc-id` - The ID of the VPC that the instance is running in.
- `network-interface.description` - The description of the network interface.
- `network-interface.subnet-id` - The ID of the subnet for the network interface.
- `network-interface.vpc-id` - The ID of the VPC for the network interface.
- `network-interface.network-interface.id` - The ID of the network interface.
- `network-interface.owner-id` - The ID of the owner of the network interface.
- `network-interface.availability-zone` - The Availability Zone for the network interface.
- `network-interface.requester-id` - The requester ID for the network interface.
- `network-interface.requester-managed` - Indicates whether the network interface is being managed by AWS.
- `network-interface.status` - The status of the network interface (`available` | `in-use`).
- `network-interface.mac-address` - The MAC address of the network interface.
- `network-interface.private-dns-name` - The private DNS name of the network interface.
- `network-interface.source-dest-check` - Whether the network interface performs source/destination checking. A value of `true` means checking is enabled, and `false` means checking is disabled. The value must be `false` for the network interface to perform network ad-

dress translation (NAT) in your VPC.

- `network-interface.group-id` - The ID of a security group associated with the network interface.
- `network-interface.group-name` - The name of a security group associated with the network interface.
- `network-interface.attachment.attachment-id` - The ID of the interface attachment.
- `network-interface.attachment.instance-id` - The ID of the instance to which the network interface is attached.
- `network-interface.attachment.instance-owner-id` - The owner ID of the instance to which the network interface is attached.
- `network-interface.addresses.private-ip-address` - The private IP address associated with the network interface.
- `network-interface.attachment.device-index` - The device index to which the network interface is attached.
- `network-interface.attachment.status` - The status of the attachment (`attaching` | `attached` | `detaching` | `detached`).
- `network-interface.attachment.attach-time` - The time that the network interface was attached to an instance.
- `network-interface.attachment.delete-on-termination` - Specifies whether the attachment is deleted when an instance is terminated.
- `network-interface.addresses.primary` - Specifies whether the IP address of the network interface is the primary private IP address.
- `network-interface.addresses.association.public-ip` - The ID of the association of an Elastic IP address with a network interface.
- `network-interface.addresses.association.ip-owner-id` - The owner ID of the private IP address associated with the network interface.
- `association.public-ip` - The address of the Elastic IP address bound to the network interface.
- `association.ip-owner-id` - The owner of the Elastic IP address associated with the network interface.
- `association.allocation-id` - The allocation ID returned when you allocated the Elastic IP address for your network interface.
- `association.association-id` - The association ID returned when the network interface was associated with an IP address.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

- **NextToken** (*string*) - The token to request the next page of results.
- **MaxResults** (*integer*) - The maximum number of results to return for the request in a single page. The remaining results of the initial request can be seen by sending another request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only

1000 results are returned. You cannot specify this parameter and the instance IDs parameter in the same request.

Returns None

wait_until_terminated (**kwargs)

Waits until this Instance is terminated. This method calls `EC2.Waiter.instance_terminated.wait()` which polls `EC2.Client.describe_instances()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
instance.wait_until_terminated(
    DryRun=True|False,
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Filters** (*list*) – One or more filters.
 - `architecture` - The instance architecture (`i386|x86_64`).
 - `availability-zone` - The Availability Zone of the instance.
 - `block-device-mapping.attach-time` - The attach time for an EBS volume mapped to the instance, for example, `2010-09-15T17:15:20.000Z`.
 - `block-device-mapping.delete-on-termination` - A Boolean that indicates whether the EBS volume is deleted on instance termination.
 - `block-device-mapping.device-name` - The device name for the EBS volume (for example, `/dev/sdh` or `xvdh`).
 - `block-device-mapping.status` - The status for the EBS volume (`attaching|attached|detaching|detached`).
 - `block-device-mapping.volume-id` - The volume ID of the EBS volume.
 - `client-token` - The idempotency token you provided when you launched the instance.
 - `dns-name` - The public DNS name of the instance.
 - `group-id` - The ID of the security group for the instance. EC2-Classic only.
 - `group-name` - The name of the security group for the instance. EC2-Classic only.
 - `hypervisor` - The hypervisor type of the instance (`ovm|xen`).
 - `iam-instance-profile.arn` - The instance profile associated with the instance. Specified as an ARN.
 - `image-id` - The ID of the image used to launch the instance.

- `instance-id` - The ID of the instance.
- `instance-lifecycle` - Indicates whether this is a Spot Instance (`spot`).
- `instance-state-code` - The state of the instance, as a 16-bit unsigned integer. The high byte is an opaque internal value and should be ignored. The low byte is set based on the state represented. The valid values are: 0 (pending), 16 (running), 32 (shutting-down), 48 (terminated), 64 (stopping), and 80 (stopped).
- `instance-state-name` - The state of the instance (`pending` | `running` | `shutting-down` | `terminated` | `stopping` | `stopped`).
- `instance-type` - The type of instance (for example, `t2.micro`).
- `instance.group-id` - The ID of the security group for the instance.
- `instance.group-name` - The name of the security group for the instance.
- `ip-address` - The public IP address of the instance.
- `kernel-id` - The kernel ID.
- `key-name` - The name of the key pair used when the instance was launched.
- `launch-index` - When launching multiple instances, this is the index for the instance in the launch group (for example, 0, 1, 2, and so on).
- `launch-time` - The time when the instance was launched.
- `monitoring-state` - Indicates whether monitoring is enabled for the instance (`disabled` | `enabled`).
- `owner-id` - The AWS account ID of the instance owner.
- `placement-group-name` - The name of the placement group for the instance.
- `platform` - The platform. Use `windows` if you have Windows instances; otherwise, leave blank.
- `private-dns-name` - The private DNS name of the instance.
- `private-ip-address` - The private IP address of the instance.
- `product-code` - The product code associated with the AMI used to launch the instance.
- `product-code.type` - The type of product code (`devpay` | `marketplace`).
- `ramdisk-id` - The RAM disk ID.
- `reason` - The reason for the current state of the instance (for example, shows “User Initiated [date]” when you stop or terminate the instance). Similar to the `state-reason-code` filter.
- `requester-id` - The ID of the entity that launched the instance on your behalf (for example, AWS Management Console, Auto Scaling, and so on).
- `reservation-id` - The ID of the instance’s reservation. A reservation ID is created any time you launch an instance. A reservation ID has a one-to-one relationship with an instance launch request, but can be associated with more than one instance if you launch multiple instances using the same launch request. For example, if you launch one instance, you’ll get one reservation ID. If you launch ten instances using the same launch request, you’ll also get one reservation ID.
- `root-device-name` - The name of the root device for the instance (for example, `/dev/sda1` or `/dev/xvda`).
- `root-device-type` - The type of root device that the instance uses (`ebs` | `instance-store`).
- `source-dest-check` - Indicates whether the instance performs

source/destination checking. A value of `true` means that checking is enabled, and `false` means checking is disabled. The value must be `false` for the instance to perform network address translation (NAT) in your VPC.

- `spot-instance-request-id` - The ID of the Spot Instance request.
- `state-reason-code` - The reason code for the state change.
- `state-reason-message` - A message that describes the state change.
- `subnet-id` - The ID of the subnet for the instance.
- `tag :key =*value*` - The key/value combination of a tag assigned to the resource, where `tag :key` is the tag's key.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter "`tag-key=Purpose`" and the filter "`tag-value=X`", you get any resources assigned both the tag key `Purpose` (regardless of what the tag's value is), and the tag value `X` (regardless of what the tag's key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `tenancy` - The tenancy of an instance (`dedicated|default`).
- `virtualization-type` - The virtualization type of the instance (`paravirtual|hvm`).
- `vpc-id` - The ID of the VPC that the instance is running in.
- `network-interface.description` - The description of the network interface.
- `network-interface.subnet-id` - The ID of the subnet for the network interface.
- `network-interface.vpc-id` - The ID of the VPC for the network interface.
- `network-interface.network-interface.id` - The ID of the network interface.
- `network-interface.owner-id` - The ID of the owner of the network interface.
- `network-interface.availability-zone` - The Availability Zone for the network interface.
- `network-interface.requester-id` - The requester ID for the network interface.
- `network-interface.requester-managed` - Indicates whether the network interface is being managed by AWS.
- `network-interface.status` - The status of the network interface (`available|in-use`).
- `network-interface.mac-address` - The MAC address of the network interface.
- `network-interface.private-dns-name` - The private DNS name of the network interface.
- `network-interface.source-dest-check` - Whether the network interface performs source/destination checking. A value of `true` means checking is enabled, and `false` means checking is disabled. The value must be `false` for the network interface to perform network address translation (NAT) in your VPC.
- `network-interface.group-id` - The ID of a security group associated with the network interface.
- `network-interface.group-name` - The name of a security group associated with the network interface.
- `network-interface.attachment.attachment-id` - The ID

of the interface attachment.

- `network-interface.attachment.instance-id` - The ID of the instance to which the network interface is attached.
- `network-interface.attachment.instance-owner-id` - The owner ID of the instance to which the network interface is attached.
- `network-interface.addresses.private-ip-address` - The private IP address associated with the network interface.
- `network-interface.attachment.device-index` - The device index to which the network interface is attached.
- `network-interface.attachment.status` - The status of the attachment (`attaching` | `attached` | `detaching` | `detached`).
- `network-interface.attachment.attach-time` - The time that the network interface was attached to an instance.
- `network-interface.attachment.delete-on-termination` - Specifies whether the attachment is deleted when an instance is terminated.
- `network-interface.addresses.primary` - Specifies whether the IP address of the network interface is the primary private IP address.
- `network-interface.addresses.association.public-ip` - The ID of the association of an Elastic IP address with a network interface.
- `network-interface.addresses.association.ip-owner-id` - The owner ID of the private IP address associated with the network interface.
- `association.public-ip` - The address of the Elastic IP address bound to the network interface.
- `association.ip-owner-id` - The owner of the Elastic IP address associated with the network interface.
- `association.allocation-id` - The allocation ID returned when you allocated the Elastic IP address for your network interface.
- `association.association-id` - The association ID returned when the network interface was associated with an IP address.
- *(dict)* -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

- **NextToken** (*string*) - The token to request the next page of results.
- **MaxResults** (*integer*) - The maximum number of results to return for the request in a single page. The remaining results of the initial request can be seen by sending another request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. You cannot specify this parameter and the `instance IDs` parameter in the same request.

Returns None

InternetGateway

class `EC2.InternetGateway` (*id*)

A resource representing an Amazon Elastic Compute Cloud (EC2) InternetGateway:

```
import boto3

ec2 = boto3.resource('ec2')
internet_gateway = ec2.InternetGateway('id')
```

Parameters *id* (*string*) – The InternetGateway’s id identifier. This **must** be set.

These are the resource’s available identifiers:

- *id*

These are the resource’s available attributes:

- *attachments*
- *internet_gateway_id*
- *tags*

These are the resource’s available actions:

- *attach_to_vpc()*
- *create_tags()*
- *delete()*
- *detach_from_vpc()*
- *load()*
- *reload()*

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

id

(*string*) The InternetGateway’s id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

attachments

(*list*)

Any VPCs attached to the Internet gateway.

internet_gateway_id

(*string*)

The ID of the Internet gateway.

tags

(*list*)

Any tags assigned to the Internet gateway.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

attach_to_vpc (***kwargs*)

Attaches an Internet gateway to a VPC, enabling connectivity between the Internet and the VPC. For more information about your VPC and Internet gateway, see the [Amazon Virtual Private Cloud User Guide](#).

Request Syntax

```
response = internet_gateway.attach_to_vpc(
    DryRun=True|False,
    VpcId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcId** (*string*) – **[REQUIRED]**

The ID of the VPC.

Returns None

`create_tags` (***kwargs*)

Adds or overwrites one or more tags for the specified Amazon EC2 resource or resources. Each resource can have a maximum of 10 tags. Each tag consists of a key and optional value. Tag keys must be unique per resource.

For more information about tags, see [Tagging Your Resources](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
tag = internet_gateway.create_tags(
    DryRun=True|False,
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Tags** (*list*) – **[REQUIRED]**

One or more tags. The `value` parameter is required, but if you don't want the tag to have a value, specify the parameter with no value, and we set the value to an empty string.

– (*dict*) –

Describes a tag.

* **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

* **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

Return type `list(ec2.Tag)`

Returns A list of Tag resources

delete (***kwargs*)

Deletes the specified Internet gateway. You must detach the Internet gateway from the VPC before you can delete it.

Request Syntax

```
response = internet_gateway.delete(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns `None`

detach_from_vpc (***kwargs*)

Detaches an Internet gateway from a VPC, disabling connectivity between the Internet and the VPC. The VPC must not contain any running instances with Elastic IP addresses.

Request Syntax

```
response = internet_gateway.detach_from_vpc(
    DryRun=True|False,
    VpcId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcId** (*string*) – **[REQUIRED]**

The ID of the VPC.

Returns `None`

load ()

Calls `ec2.Client.describe_internet_gateways()` to update the attributes of the Internet-Gateway resource

Request Syntax

```
internet_gateway.load()
```

Returns `None`

reload ()

Calls `ec2.Client.describe_internet_gateways()` to update the attributes of the Internet-Gateway resource

Request Syntax

```
internet_gateway.reload()
```

Returns None

KeyPair

class EC2.**KeyPair** (*name*)

A resource representing an Amazon Elastic Compute Cloud (EC2) KeyPair:

```
import boto3
```

```
ec2 = boto3.resource('ec2')
key_pair = ec2.KeyPair('name')
```

Parameters *name* (*string*) – The KeyPair’s name identifier. This **must** be set.

These are the resource’s available identifiers:

- *name*

These are the resource’s available attributes:

- *key_fingerprint*
- *key_name*

These are the resource’s available actions:

- *delete()*
- *load()*
- *reload()*

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

name

(*string*) The KeyPair’s name identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

key_fingerprint

(*string*)

If you used `CreateKeyPair` to create the key pair, this is the SHA-1 digest of the DER encoded private key. If you used `ImportKeyPair` to provide AWS the public key, this is the MD5 public key fingerprint as specified in section 4 of RFC4716.

key_name

(*string*)

The name of the key pair.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

delete (***kwargs*)

Deletes the specified key pair, by removing the public key from Amazon EC2.

Request Syntax

```
response = key_pair.delete(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns `None`

load()

Calls `ec2.Client.describe_key_pairs()` to update the attributes of the `KeyPair` resource

Request Syntax

```
key_pair.load()
```

Returns `None`

reload()

Calls `ec2.Client.describe_key_pairs()` to update the attributes of the `KeyPair` resource

Request Syntax

```
key_pair.reload()
```

Returns `None`

NetworkAcl

class `EC2.NetworkAcl` (*id*)

A resource representing an Amazon Elastic Compute Cloud (EC2) NetworkAcl:

```
import boto3
```

```
ec2 = boto3.resource('ec2')
network_acl = ec2.NetworkAcl('id')
```

Parameters **id** (*string*) – The NetworkAcl’s id identifier. This **must** be set.

These are the resource’s available identifiers:

- `id`

These are the resource’s available attributes:

- `associations`
- `entries`
- `is_default`
- `network_acl_id`
- `tags`
- `vpc_id`

These are the resource’s available references:

- `vpc`

These are the resource’s available actions:

- `create_entry()`
- `create_tags()`
- `delete()`

- `delete_entry()`
- `load()`
- `reload()`
- `replace_association()`
- `replace_entry()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

id

(*string*) The NetworkAcl's id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

associations

(*list*)

Any associations between the network ACL and one or more subnets

entries

(*list*)

One or more entries (rules) in the network ACL.

is_default

(*boolean*)

Indicates whether this is the default network ACL for the VPC.

network_acl_id

(*string*)

The ID of the network ACL.

tags

(*list*)

Any tags assigned to the network ACL.

vpc_id

(*string*)

The ID of the VPC for the network ACL.

References

References are related resource instances that have a belongs-to relationship. For more information about references refer to the [Resources Introduction Guide](#).

vpc

(Vpc) The related vpc if set, otherwise None.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

create_entry (***kwargs*)

Creates an entry (a rule) in a network ACL with the specified rule number. Each network ACL has a set of numbered ingress rules and a separate set of numbered egress rules. When determining whether a packet should be allowed in or out of a subnet associated with the ACL, we process the entries in the

ACL according to the rule numbers, in ascending order. Each network ACL has a set of ingress rules and a separate set of egress rules.

We recommend that you leave room between the rule numbers (for example, 100, 110, 120, ...), and not number them one right after the other (for example, 101, 102, 103, ...). This makes it easier to add a rule between existing ones without having to renumber the rules.

After you add an entry, you can't modify it; you must either replace it, or create an entry and delete the old one.

For more information about network ACLs, see [Network ACLs](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = network_acl.create_entry(
    DryRun=True|False,
    RuleNumber=123,
    Protocol='string',
    RuleAction='allow'|'deny',
    Egress=True|False,
    CidrBlock='string',
    IcmpTypeCode={
        'Type': 123,
        'Code': 123
    },
    PortRange={
        'From': 123,
        'To': 123
    }
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **RuleNumber** (*integer*) – **[REQUIRED]**

The rule number for the entry (for example, 100). ACL entries are processed in ascending order by rule number.

Constraints: Positive integer from 1 to 32766
- **Protocol** (*string*) – **[REQUIRED]**

The protocol. A value of -1 means all protocols.
- **RuleAction** (*string*) – **[REQUIRED]**

Indicates whether to allow or deny the traffic that matches the rule.
- **Egress** (*boolean*) – **[REQUIRED]**

Indicates whether this is an egress rule (rule is applied to traffic leaving the subnet).
- **CidrBlock** (*string*) – **[REQUIRED]**

The network range to allow or deny, in CIDR notation (for example 172.16.0.0/24).
- **IcmpTypeCode** (*dict*) – ICMP protocol: The ICMP type and code. Required if specifying ICMP for the protocol.

- **Type** (*integer*) –
The ICMP code. A value of -1 means all codes for the specified ICMP type.
- **Code** (*integer*) –
The ICMP type. A value of -1 means all types.
- **PortRange** (*dict*) – TCP or UDP protocols: The range of ports the rule applies to.
 - **From** (*integer*) –
The first port in the range.
 - **To** (*integer*) –
The last port in the range.

Returns None

create_tags (***kwargs*)

Adds or overwrites one or more tags for the specified Amazon EC2 resource or resources. Each resource can have a maximum of 10 tags. Each tag consists of a key and optional value. Tag keys must be unique per resource.

For more information about tags, see [Tagging Your Resources](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
tag = network_acl.create_tags(
    DryRun=True|False,
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Tags** (*list*) – **[REQUIRED]**

One or more tags. The `value` parameter is required, but if you don't want the tag to have a value, specify the parameter with no value, and we set the value to an empty string.

- (*dict*) –

Describes a tag.

- * **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

- * **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

Return type `list(ec2.Tag)`

Returns A list of Tag resources

delete (***kwargs*)

Deletes the specified network ACL. You can't delete the ACL if it's associated with any subnets. You can't delete the default network ACL.

Request Syntax

```
response = network_acl.delete(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns None

delete_entry (***kwargs*)

Deletes the specified ingress or egress entry (rule) from the specified network ACL.

Request Syntax

```
response = network_acl.delete_entry(
    DryRun=True|False,
    RuleNumber=123,
    Egress=True|False
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **RuleNumber** (*integer*) – **[REQUIRED]**
The rule number of the entry to delete.
- **Egress** (*boolean*) – **[REQUIRED]**
Indicates whether the rule is an egress rule.

Returns None

load ()

Calls `ec2.Client.describe_network_acls()` to update the attributes of the `NetworkAcl` resource

Request Syntax

```
network_acl.load()
```

Returns None

reload ()

Calls `ec2.Client.describe_network_acls()` to update the attributes of the `NetworkAcl` resource

Request Syntax

```
network_acl.reload()
```

Returns None

replace_association (**kwargs)

Changes which network ACL a subnet is associated with. By default when you create a subnet, it's automatically associated with the default network ACL. For more information about network ACLs, see [Network ACLs](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = network_acl.replace_association(
    DryRun=True|False,
    AssociationId='string',
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **AssociationId** (*string*) – **[REQUIRED]**

The ID of the current association between the original network ACL and the subnet.

Return type dict

Returns

Response Syntax

```
{
    'NewAssociationId': 'string'
}
```

Response Structure

- (*dict*) –
 - **NewAssociationId** (*string*) –

The ID of the new association.

replace_entry (**kwargs)

Replaces an entry (rule) in a network ACL. For more information about network ACLs, see [Network ACLs](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = network_acl.replace_entry(
    DryRun=True|False,
    RuleNumber=123,
    Protocol='string',
    RuleAction='allow'|'deny',
    Egress=True|False,
    CidrBlock='string',
    IcmpTypeCode={
        'Type': 123,
        'Code': 123
    }
)
```

```
},
PortRange={
    'From': 123,
    'To': 123
}
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **RuleNumber** (*integer*) – **[REQUIRED]**
The rule number of the entry to replace.
- **Protocol** (*string*) – **[REQUIRED]**
The IP protocol. You can specify `all` or `-1` to mean all protocols.
- **RuleAction** (*string*) – **[REQUIRED]**
Indicates whether to allow or deny the traffic that matches the rule.
- **Egress** (*boolean*) – **[REQUIRED]**
Indicates whether to replace the egress rule.
Default: If no value is specified, we replace the ingress rule.
- **CidrBlock** (*string*) – **[REQUIRED]**
The network range to allow or deny, in CIDR notation.
- **IcmpTypeCode** (*dict*) – ICMP protocol: The ICMP type and code. Required if specifying 1 (ICMP) for the protocol.
 - **Type** (*integer*) –
The ICMP code. A value of `-1` means all codes for the specified ICMP type.
 - **Code** (*integer*) –
The ICMP type. A value of `-1` means all types.
- **PortRange** (*dict*) – TCP or UDP protocols: The range of ports the rule applies to. Required if specifying 6 (TCP) or 17 (UDP) for the protocol.
 - **From** (*integer*) –
The first port in the range.
 - **To** (*integer*) –
The last port in the range.

Returns None

NetworkInterface

class `EC2.NetworkInterface` (*id*)

A resource representing an Amazon Elastic Compute Cloud (EC2) NetworkInterface:

```
import boto3

ec2 = boto3.resource('ec2')
network_interface = ec2.NetworkInterface('id')
```

Parameters **id** (*string*) – The NetworkInterface’s id identifier. This **must** be set.

These are the resource's available identifiers:

- `id`

These are the resource's available attributes:

- `association`
- `attachment`
- `availability_zone`
- `description`
- `groups`
- `mac_address`
- `network_interface_id`
- `owner_id`
- `private_dns_name`
- `private_ip_address`
- `private_ip_addresses`
- `requester_id`
- `requester_managed`
- `source_dest_check`
- `status`
- `subnet_id`
- `tag_set`
- `vpc_id`

These are the resource's available references:

- `subnet`
- `vpc`

These are the resource's available actions:

- `assign_private_ip_addresses()`
- `attach()`
- `create_tags()`
- `delete()`
- `describe_attribute()`
- `detach()`
- `load()`
- `modify_attribute()`
- `reload()`
- `reset_attribute()`
- `unassign_private_ip_addresses()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

id

(*string*) The NetworkInterface's id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

association

(*dict*)

The association information for an Elastic IP associated with the network interface.

attachment

(*dict*)

The network interface attachment.

availability_zone

(string)

The Availability Zone.

description

(string)

A description.

groups

(list)

Any security groups for the network interface.

mac_address

(string)

The MAC address.

network_interface_id

(string)

The ID of the network interface.

owner_id

(string)

The AWS account ID of the owner of the network interface.

private_dns_name

(string)

The private DNS name.

private_ip_address

(string)

The IP address of the network interface within the subnet.

private_ip_addresses

(list)

The private IP addresses associated with the network interface.

requester_id

(string)

The ID of the entity that launched the instance on your behalf (for example, AWS Management Console or Auto Scaling).

requester_managed

(boolean)

Indicates whether the network interface is being managed by AWS.

source_dest_check

(boolean)

Indicates whether traffic to or from the instance is validated.

status

(string)

The status of the network interface.

subnet_id
(string)

The ID of the subnet.

tag_set
(list)

Any tags assigned to the network interface.

vpc_id
(string)

The ID of the VPC.

References

References are related resource instances that have a belongs-to relationship. For more information about references refer to the [Resources Introduction Guide](#).

subnet
(Subnet) The related subnet if set, otherwise None.

vpc
(Vpc) The related vpc if set, otherwise None.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

assign_private_ip_addresses (**kwargs)

Assigns one or more secondary private IP addresses to the specified network interface. You can specify one or more specific secondary IP addresses, or you can specify the number of secondary IP addresses to be automatically assigned within the subnet's CIDR block range. The number of secondary IP addresses that you can assign to an instance varies by instance type. For information about instance types, see [Instance Types](#) in the *Amazon Elastic Compute Cloud User Guide*. For more information about Elastic IP addresses, see [Elastic IP Addresses](#) in the *Amazon Elastic Compute Cloud User Guide*.

AssignPrivateIpAddresses is available only in EC2-VPC.

Request Syntax

```
response = network_interface.assign_private_ip_addresses(
    PrivateIpAddresses=[
        'string',
    ],
    SecondaryPrivateIpAddressCount=123,
    AllowReassignment=True|False
)
```

Parameters

- **PrivateIpAddresses** (list) – One or more IP addresses to be assigned as a secondary private IP address to the network interface. You can't specify this parameter when also specifying a number of secondary IP addresses.

If you don't specify an IP address, Amazon EC2 automatically selects an IP address within the subnet range.

– (string) –

- **SecondaryPrivateIpAddressCount** (integer) – The number of secondary IP addresses to assign to the network interface. You can't specify this parameter when also specifying private IP addresses.

- **AllowReassignment** (*boolean*) – Indicates whether to allow an IP address that is already assigned to another network interface or instance to be reassigned to the specified network interface.

Returns None

attach (***kwargs*)

Attaches a network interface to an instance.

Request Syntax

```
response = network_interface.attach(  
    DryRun=True|False,  
    InstanceId='string',  
    DeviceIndex=123  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceId** (*string*) – **[REQUIRED]**
The ID of the instance.
- **DeviceIndex** (*integer*) – **[REQUIRED]**
The index of the device for the network interface attachment.

Return type dict

Returns

Response Syntax

```
{  
    'AttachmentId': 'string'  
}
```

Response Structure

- (*dict*) –
 - **AttachmentId** (*string*) –
The ID of the network interface attachment.

create_tags (***kwargs*)

Adds or overwrites one or more tags for the specified Amazon EC2 resource or resources. Each resource can have a maximum of 10 tags. Each tag consists of a key and optional value. Tag keys must be unique per resource.

For more information about tags, see [Tagging Your Resources](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
tag = network_interface.create_tags(  
    DryRun=True|False,  
    Tags=[  
        {  
            'Key': 'string',  
            'Value': 'string'  
        },  
    ],
```



```
]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Tags** (*list*) – **[REQUIRED]**

One or more tags. The `value` parameter is required, but if you don't want the tag to have a value, specify the parameter with no value, and we set the value to an empty string.

– (*dict*) –

Describes a tag.

* **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

* **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

Return type `list(ec2.Tag)`

Returns A list of Tag resources

delete (***kwargs*)

Deletes the specified network interface. You must detach the network interface before you can delete it.

Request Syntax

```
response = network_interface.delete(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns `None`

describe_attribute (***kwargs*)

Describes a network interface attribute. You can specify only one attribute at a time.

Request Syntax

```
response = network_interface.describe_attribute(
    DryRun=True|False,
    Attribute='description'|'groupSet'|'sourceDestCheck'|'attachment'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Attribute** (*string*) – The attribute of the network interface.

Return type dict

Returns

Response Syntax

```
{
  'NetworkInterfaceId': 'string',
  'Description': {
    'Value': 'string'
  },
  'SourceDestCheck': {
    'Value': True|False
  },
  'Groups': [
    {
      'GroupName': 'string',
      'GroupId': 'string'
    },
  ],
  'Attachment': {
    'AttachmentId': 'string',
    'InstanceId': 'string',
    'InstanceOwnerId': 'string',
    'DeviceIndex': 123,
    'Status': 'attaching'|'attached'|'detaching'|'detached',
    'AttachTime': datetime(2015, 1, 1),
    'DeleteOnTermination': True|False
  }
}
```

Response Structure

- (*dict*) –
 - **NetworkInterfaceId** (*string*) –
The ID of the network interface.
 - **Description** (*dict*) –
The description of the network interface.
 - * **Value** (*string*) –
Valid values are case-sensitive and vary by action.
 - **SourceDestCheck** (*dict*) –
Indicates whether source/destination checking is enabled.
 - * **Value** (*boolean*) –
Valid values are `true` or `false`.
 - **Groups** (*list*) –
The security groups associated with the network interface.
 - * (*dict*) –
Describes a security group.
 - **GroupName** (*string*) –

- The name of the security group.
 - **GroupId** (*string*) –
 - The ID of the security group.
- **Attachment** (*dict*) –
 - The attachment (if any) of the network interface.
 - * **AttachmentId** (*string*) –
 - The ID of the network interface attachment.
 - * **InstanceId** (*string*) –
 - The ID of the instance.
 - * **InstanceOwnerId** (*string*) –
 - The AWS account ID of the owner of the instance.
 - * **DeviceIndex** (*integer*) –
 - The device index of the network interface attachment on the instance.
 - * **Status** (*string*) –
 - The attachment state.
 - * **AttachTime** (*datetime*) –
 - The timestamp indicating when the attachment initiated.
 - * **DeleteOnTermination** (*boolean*) –
 - Indicates whether the network interface is deleted when the instance is terminated.

detach (***kwargs*)

Detaches a network interface from an instance.

Request Syntax

```
response = network_interface.detach(
    DryRun=True|False,
    Force=True|False
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Force** (*boolean*) – Specifies whether to force a detachment.

Returns None

load ()

Calls `ec2.Client.describe_network_interfaces()` to update the attributes of the Network-Interface resource

Request Syntax

```
network_interface.load()
```

Returns None

modify_attribute (***kwargs*)

Modifies the specified network interface attribute. You can specify only one attribute at a time.

Request Syntax

```
response = network_interface.modify_attribute(  
    DryRun=True|False,  
    Description={  
        'Value': 'string'  
    },  
    SourceDestCheck={  
        'Value': True|False  
    },  
    Groups=[  
        'string',  
    ],  
    Attachment={  
        'AttachmentId': 'string',  
        'DeleteOnTermination': True|False  
    }  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Description** (*dict*) – A description for the network interface.
 - **Value** (*string*) –

Valid values are case-sensitive and vary by action.

- **SourceDestCheck** (*dict*) – Indicates whether source/destination checking is enabled. A value of `true` means checking is enabled, and `false` means checking is disabled. This value must be `false` for a NAT instance to perform NAT. For more information, see [NAT Instances](#) in the *Amazon Virtual Private Cloud User Guide*.
 - **Value** (*boolean*) –

Valid values are `true` or `false`.

- **Groups** (*list*) – Changes the security groups for the network interface. The new set of groups you specify replaces the current set. You must specify at least one group, even if it's just the default security group in the VPC. You must specify the ID of the security group, not the name.
 - (*string*) –
- **Attachment** (*dict*) – Information about the interface attachment. If modifying the 'delete on termination' attribute, you must specify the ID of the interface attachment.
 - **AttachmentId** (*string*) –

The ID of the network interface attachment.

- **DeleteOnTermination** (*boolean*) –

Indicates whether the network interface is deleted when the instance is terminated.

Returns None

reload()

Calls `ec2.Client.describe_network_interfaces()` to update the attributes of the Network-Interface resource

Request Syntax

```
network_interface.reload()
```

Returns None

reset_attribute (***kwargs*)

Resets a network interface attribute. You can specify only one attribute at a time.

Request Syntax

```
response = network_interface.reset_attribute(
    DryRun=True|False,
    SourceDestCheck='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SourceDestCheck** (*string*) – The source/destination checking attribute. Resets the value to `true`.

Returns None

unassign_private_ip_addresses (***kwargs*)

Unassigns one or more secondary private IP addresses from a network interface.

Request Syntax

```
response = network_interface.unassign_private_ip_addresses(
    PrivateIpAddresses=[
        'string',
    ]
)
```

Parameters **PrivateIpAddresses** (*list*) – [REQUIRED]

The secondary private IP addresses to unassign from the network interface. You can specify this option multiple times to unassign more than one IP address.

- (*string*) –

Returns None

PlacementGroup

class `EC2.PlacementGroup` (*name*)

A resource representing an Amazon Elastic Compute Cloud (EC2) PlacementGroup:

```
import boto3
```

```
ec2 = boto3.resource('ec2')
placement_group = ec2.PlacementGroup('name')
```

Parameters **name** (*string*) – The PlacementGroup’s name identifier. This **must** be set.

These are the resource’s available identifiers:

- `name`

These are the resource’s available attributes:

- `group_name`

- `state`
- `strategy`

These are the resource's available actions:

- `delete()`
- `load()`
- `reload()`

These are the resource's available collections:

- `instances`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

name

(*string*) The PlacementGroup's name identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

group_name

(*string*)

The name of the placement group.

state

(*string*)

The state of the placement group.

strategy

(*string*)

The placement strategy.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

delete (***kwargs*)

Deletes the specified placement group. You must terminate all instances in the placement group before you can delete the placement group. For more information about placement groups and cluster instances, see [Cluster Instances](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = placement_group.delete(  
    DryRun=True|False,  
  
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns `None`

load()

Calls `ec2.Client.describe_placement_groups()` to update the attributes of the Placement-Group resource

Request Syntax

```
placement_group.load()
```

Returns None**reload()**

Calls `ec2.Client.describe_placement_groups()` to update the attributes of the Placement-Group resource

Request Syntax

```
placement_group.reload()
```

Returns None**Collections**

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

instances**all()**

Creates an iterable of all Instance resources in the collection.

Request Syntax

```
instance_iterator = placement_group.instances.all()
```

Return type `list(ec2.Instance)`**Returns** A list of Instance resources**create_tags(**kwargs)**

Adds or overwrites one or more tags for the specified Amazon EC2 resource or resources. Each resource can have a maximum of 10 tags. Each tag consists of a key and optional value. Tag keys must be unique per resource.

For more information about tags, see [Tagging Your Resources](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = placement_group.instances.create_tags(
    DryRun=True|False,
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **Tags** (*list*) – [REQUIRED]

One or more tags. The `value` parameter is required, but if you don't want the tag to have a value, specify the parameter with no value, and we set the value to an empty string.

– (*dict*) –

Describes a tag.

* **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

* **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

Returns None

filter (***kwargs*)

Creates an iterable of all Instance resources in the collection filtered by kwargs passed to method.

Request Syntax

```
instance_iterator = placement_group.instances.filter(  
    DryRun=True|False,  
    InstanceIds=[  
        'string',  
    ],  
    NextToken='string',  
    MaxResults=123  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceIds** (*list*) – One or more instance IDs.

Default: Describes all your instances.

– (*string*) –

- **NextToken** (*string*) – The token to request the next page of results.
- **MaxResults** (*integer*) – The maximum number of results to return for the request in a single page. The remaining results of the initial request can be seen by sending another request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. You cannot specify this parameter and the instance IDs parameter in the same request.

Return type `list(ec2.Instance)`

Returns A list of Instance resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Instance resources in the collection.

Request Syntax


```
instance_iterator = placement_group.instances.limit(
    count=123
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(ec2.Instance)`

Returns A list of Instance resources

monitor (***kwargs*)

Enables monitoring for a running instance. For more information about monitoring instances, see [Monitoring Your Instances and Volumes](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = placement_group.instances.monitor(
    DryRun=True|False,
)
```

Parameters `DryRun` (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type `dict`

Returns

Response Syntax

```
{
  'InstanceMonitorings': [
    {
      'InstanceId': 'string',
      'Monitoring': {
        'State': 'disabled'|'disabling'|'enabled'|'pending'
      }
    },
  ],
}
```

Response Structure

- (*dict*) –

- `InstanceMonitorings` (*list*) –

Monitoring information for one or more instances.

- * (*dict*) –

Describes the monitoring information of the instance.

- `InstanceId` (*string*) –

The ID of the instance.

- `Monitoring` (*dict*) –

The monitoring information.

- `State` (*string*) –

Indicates whether monitoring is enabled for the instance.

page_size (**kwargs)

Creates an iterable of all Instance resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
instance_iterator = placement_group.instances.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type list(ec2.Instance)

Returns A list of Instance resources

reboot (**kwargs)

Requests a reboot of one or more instances. This operation is asynchronous; it only queues a request to reboot the specified instances. The operation succeeds if the instances are valid and belong to you. Requests to reboot terminated instances are ignored.

If a Linux/Unix instance does not cleanly shut down within four minutes, Amazon EC2 performs a hard reboot.

For more information about troubleshooting, see [Getting Console Output and Rebooting Instances](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = placement_group.instances.reboot(  
    DryRun=True|False,  
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns None

start (**kwargs)

Starts an Amazon EBS-backed AMI that you've previously stopped.

Instances that use Amazon EBS volumes as their root devices can be quickly stopped and started. When an instance is stopped, the compute resources are released and you are not billed for hourly instance usage. However, your root partition Amazon EBS volume remains, continues to persist your data, and you are charged for Amazon EBS volume usage. You can restart your instance at any time. Each time you transition an instance from stopped to started, Amazon EC2 charges a full instance hour, even if transitions happen multiple times within a single hour.

Before stopping an instance, make sure it is in a state from which it can be restarted. Stopping an instance does not preserve data stored in RAM.

Performing this operation on an instance that uses an instance store as its root device returns an error.

For more information, see [Stopping Instances](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = placement_group.instances.start(
    AdditionalInfo='string',
    DryRun=True|False
)
```

Parameters

- **AdditionalInfo** (*string*) – Reserved.
- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns

Response Syntax

```
{
  'StartingInstances': [
    {
      'InstanceId': 'string',
      'CurrentState': {
        'Code': 123,
        'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stop
      },
      'PreviousState': {
        'Code': 123,
        'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stop
      }
    },
  ],
}
```

Response Structure

- (*dict*) –
 - **StartingInstances** (*list*) –

Information about one or more started instances.

* (*dict*) –

Describes an instance state change.

- **InstanceId** (*string*) –

The ID of the instance.

- **CurrentState** (*dict*) –

The current state of the instance.

- **Code** (*integer*) –

The low byte represents the state. The high byte is an opaque internal value and should be ignored.

- 0: pending
- 16: running
- 32: shutting-down
- 48: terminated
- 64: stopping
- 80: stopped
- **Name** (*string*) –

The current state of the instance.

- **PreviousState** (*dict*) –
The previous state of the instance.
- **Code** (*integer*) –
The low byte represents the state. The high byte is an opaque internal value and should be ignored.
 - 0: pending
 - 16: running
 - 32: shutting-down
 - 48: terminated
 - 64: stopping
 - 80: stopped
- **Name** (*string*) –
The current state of the instance.

stop (***kwargs*)

Stops an Amazon EBS-backed instance. Each time you transition an instance from stopped to started, Amazon EC2 charges a full instance hour, even if transitions happen multiple times within a single hour.

You can't start or stop Spot Instances.

Instances that use Amazon EBS volumes as their root devices can be quickly stopped and started. When an instance is stopped, the compute resources are released and you are not billed for hourly instance usage. However, your root partition Amazon EBS volume remains, continues to persist your data, and you are charged for Amazon EBS volume usage. You can restart your instance at any time.

Before stopping an instance, make sure it is in a state from which it can be restarted. Stopping an instance does not preserve data stored in RAM.

Performing this operation on an instance that uses an instance store as its root device returns an error.

You can stop, start, and terminate EBS-backed instances. You can only terminate instance store-backed instances. What happens to an instance differs if you stop it or terminate it. For example, when you stop an instance, the root device and any other devices attached to the instance persist. When you terminate an instance, the root device and any other devices attached during the instance launch are automatically deleted. For more information about the differences between stopping and terminating instances, see [Instance Lifecycle](#) in the *Amazon Elastic Compute Cloud User Guide*.

For more information about troubleshooting, see [Troubleshooting Stopping Your Instance](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = placement_group.instances.stop(  
    DryRun=True|False,  
    Force=True|False  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Force** (*boolean*) – Forces the instances to stop. The instances do not have an opportunity to flush file system caches or file system metadata. If you

use this option, you must perform file system check and repair procedures.
This option is not recommended for Windows instances.

Default: `false`

Return type `dict`

Returns

Response Syntax

```
{
  'StoppingInstances': [
    {
      'InstanceId': 'string',
      'CurrentState': {
        'Code': 123,
        'Name': 'pending' | 'running' | 'shutting-down' | 'terminated' | 'stopping' | 'stopped',
      },
      'PreviousState': {
        'Code': 123,
        'Name': 'pending' | 'running' | 'shutting-down' | 'terminated' | 'stopping' | 'stopped',
      },
    },
  ],
}
```

Response Structure

- (*dict*) –
 - **StoppingInstances** (*list*) –

Information about one or more stopped instances.

* (*dict*) –

Describes an instance state change.

- **InstanceId** (*string*) –
The ID of the instance.
- **CurrentState** (*dict*) –
The current state of the instance.
- **Code** (*integer*) –
The low byte represents the state. The high byte is an opaque internal value and should be ignored.
 - 0: pending
 - 16: running
 - 32: shutting-down
 - 48: terminated
 - 64: stopping
 - 80: stopped
- **Name** (*string*) –
The current state of the instance.
- **PreviousState** (*dict*) –
The previous state of the instance.
- **Code** (*integer*) –
The low byte represents the state. The high byte is an opaque internal value and should be ignored.
 - 0: pending

- 16: running
- 32: shutting-down
- 48: terminated
- 64: stopping
- 80: stopped
- **Name** (*string*) –

The current state of the instance.

terminate (***kwargs*)

Shuts down one or more instances. This operation is idempotent; if you terminate an instance more than once, each call succeeds.

Terminated instances remain visible after termination (for approximately one hour).

By default, Amazon EC2 deletes all EBS volumes that were attached when the instance launched. Volumes attached after instance launch continue running.

You can stop, start, and terminate EBS-backed instances. You can only terminate instance store-backed instances. What happens to an instance differs if you stop it or terminate it. For example, when you stop an instance, the root device and any other devices attached to the instance persist. When you terminate an instance, any attached EBS volumes with the `DeleteOnTermination` block device mapping parameter set to `true` are automatically deleted. For more information about the differences between stopping and terminating instances, see [Instance Lifecycle](#) in the *Amazon Elastic Compute Cloud User Guide*.

For more information about troubleshooting, see [Troubleshooting Terminating Your Instance](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = placement_group.instances.terminate(  
    DryRun=True|False,  
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns

Response Syntax

```
{  
    'TerminatingInstances': [  
        {  
            'InstanceId': 'string',  
            'CurrentState': {  
                'Code': 123,  
                'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stop  
            },  
            'PreviousState': {  
                'Code': 123,  
                'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stop  
            }  
        },  
    ],  
}
```

```
]
}
```

Response Structure

- *(dict)* –
 - **TerminatingInstances** (*list*) –

Information about one or more terminated instances.

* *(dict)* –

Describes an instance state change.

- **InstanceId** (*string*) –

The ID of the instance.

- **CurrentState** (*dict*) –

The current state of the instance.

- **Code** (*integer*) –

The low byte represents the state. The high byte is an opaque internal value and should be ignored.

- 0: pending
- 16: running
- 32: shutting-down
- 48: terminated
- 64: stopping
- 80: stopped
- **Name** (*string*) –

The current state of the instance.

- **PreviousState** (*dict*) –

The previous state of the instance.

- **Code** (*integer*) –

The low byte represents the state. The high byte is an opaque internal value and should be ignored.

- 0: pending
- 16: running
- 32: shutting-down
- 48: terminated
- 64: stopping
- 80: stopped
- **Name** (*string*) –

The current state of the instance.

unmonitor (***kwargs*)

Disables monitoring for a running instance. For more information about monitoring instances, see [Monitoring Your Instances and Volumes](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = placement_group.instances.unmonitor(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an

error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns

Response Syntax

```
{
  'InstanceMonitorings': [
    {
      'InstanceId': 'string',
      'Monitoring': {
        'State': 'disabled'|'disabling'|'enabled'|'pending'
      }
    },
  ],
}
```

Response Structure

- (*dict*) –

– **InstanceMonitorings** (*list*) –

Monitoring information for one or more instances.

- * (*dict*) –

Describes the monitoring information of the instance.

- **InstanceId** (*string*) –

The ID of the instance.

- **Monitoring** (*dict*) –

The monitoring information.

- **State** (*string*) –

Indicates whether monitoring is enabled for the instance.

RouteTable

class `EC2.RouteTable` (*id*)

A resource representing an Amazon Elastic Compute Cloud (EC2) RouteTable:

```
import boto3
```

```
ec2 = boto3.resource('ec2')
route_table = ec2.RouteTable('id')
```

Parameters *id* (*string*) – The RouteTable’s id identifier. This **must** be set.

These are the resource’s available identifiers:

- *id*

These are the resource’s available attributes:

- *associations_attribute*
- *propagating_vgws*
- *route_table_id*
- *routes*
- *tags*
- *vpc_id*

These are the resource's available references:

- `vpc`

These are the resource's available actions:

- `associate_with_subnet()`
- `create_route()`
- `create_tags()`
- `delete()`
- `load()`
- `reload()`

These are the resource's available collections:

- `associations`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

id

(*string*) The RouteTable's id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

associations_attribute

(*list*)

The associations between the route table and one or more subnets.

propagating_vgws

(*list*)

Any virtual private gateway (VGW) propagating routes.

route_table_id

(*string*)

The ID of the route table.

routes

(*list*)

The routes in the route table.

tags

(*list*)

Any tags assigned to the route table.

vpc_id

(*string*)

The ID of the VPC.

References

References are related resource instances that have a belongs-to relationship. For more information about references refer to the [Resources Introduction Guide](#).

vpc

(Vpc) The related vpc if set, otherwise None.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

associate_with_subnet (**kwargs)

Associates a subnet with a route table. The subnet and route table must be in the same VPC. This association causes traffic originating from the subnet to be routed according to the routes in the route table. The action returns an association ID, which you need in order to disassociate the route table from the subnet later. A route table can be associated with multiple subnets.

For more information about route tables, see [Route Tables](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
route_table_association = route_table.associate_with_subnet(
    DryRun=True|False,
    SubnetId='string',
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SubnetId** (*string*) – **[REQUIRED]**

The ID of the subnet.

Return type `ec2.RouteTableAssociation`

Returns `RouteTableAssociation` resource

create_route (**kwargs)

Creates a route in a route table within a VPC.

You must specify one of the following targets: Internet gateway or virtual private gateway, NAT instance, VPC peering connection, or network interface.

When determining how to route traffic, we use the route with the most specific match. For example, let's say the traffic is destined for `192.0.2.3`, and the route table includes the following two routes:

- `192.0.2.0/24` (goes to some target A)
- `192.0.2.0/28` (goes to some target B)

Both routes apply to the traffic destined for `192.0.2.3`. However, the second route in the list covers a smaller number of IP addresses and is therefore more specific, so we use that route to determine where to target the traffic.

For more information about route tables, see [Route Tables](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = route_table.create_route(
    DryRun=True|False,
    DestinationCidrBlock='string',
    GatewayId='string',
    InstanceId='string',
    NetworkInterfaceId='string',
    VpcPeeringConnectionId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **DestinationCidrBlock** (*string*) – **[REQUIRED]**
The CIDR address block used for the destination match. Routing decisions are based on the most specific match.
- **GatewayId** (*string*) – The ID of an Internet gateway or virtual private gateway attached to your VPC.
- **InstanceId** (*string*) – The ID of a NAT instance in your VPC. The operation fails if you specify an instance ID unless exactly one network interface is attached.
- **NetworkInterfaceId** (*string*) – The ID of a network interface.
- **VpcPeeringConnectionId** (*string*) – The ID of a VPC peering connection.

Return type dict

Returns

Response Syntax

```
{
    'Return': True|False
}
```

Response Structure

- (*dict*) –
 - **Return** (*boolean*) –

Returns `true` if the request succeeds; otherwise, it returns an error.

`create_tags` (***kwargs*)

Adds or overwrites one or more tags for the specified Amazon EC2 resource or resources. Each resource can have a maximum of 10 tags. Each tag consists of a key and optional value. Tag keys must be unique per resource.

For more information about tags, see [Tagging Your Resources](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
tag = route_table.create_tags(
    DryRun=True|False,
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Tags** (*list*) – **[REQUIRED]**

One or more tags. The `value` parameter is required, but if you don't want the tag to have a value, specify the parameter with no value, and we set the value to an empty string.

– (dict) –

Describes a tag.

* **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

* **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

Return type `list(ec2.Tag)`

Returns A list of Tag resources

delete (***kwargs*)

Deletes the specified route table. You must disassociate the route table from any subnets before you can delete it. You can't delete the main route table.

Request Syntax

```
response = route_table.delete(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns None

load ()

Calls `ec2.Client.describe_route_tables()` to update the attributes of the `RouteTable` resource

Request Syntax

```
route_table.load()
```

Returns None

reload ()

Calls `ec2.Client.describe_route_tables()` to update the attributes of the `RouteTable` resource

Request Syntax

```
route_table.reload()
```

Returns None

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

associations

all()

Creates an iterable of all RouteTableAssociation resources in the collection.

Request Syntax

```
route_table_association_iterator = route_table.associations.all()
```

Return type list(ec2.RouteTableAssociation)

Returns A list of RouteTableAssociation resources

filter(kwargs)**

Creates an iterable of all RouteTableAssociation resources in the collection filtered by kwargs passed to method.

Request Syntax

```
route_table_association_iterator = route_table.associations.filter(
    DryRun=True|False,
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Filters** (*list*) – One or more filters.
 - `association.route-table-association-id` - The ID of an association ID for the route table.
 - `association.route-table-id` - The ID of the route table involved in the association.
 - `association.subnet-id` - The ID of the subnet involved in the association.
 - `association.main` - Indicates whether the route table is the main route table for the VPC.
 - `route-table-id` - The ID of the route table.
 - `route.destination-cidr-block` - The CIDR range specified in a route in the table.
 - `route.destination-prefix-list-id` - The ID (prefix) of the AWS service specified in a route in the table.
 - `route.gateway-id` - The ID of a gateway specified in a route in the table.
 - `route.instance-id` - The ID of an instance specified in a route in the table.
 - `route.origin` - Describes how the route was created. `CreateRouteTable` indicates that the route was automatically created when the route table was created; `CreateRoute` indicates that the route was manually added to the route table; `EnableVgwRoutePropagation` indicates that the route was propagated by route propagation.

- `route.state` - The state of a route in the route table (`active` | `blackhole`). The blackhole state indicates that the route's target isn't available (for example, the specified gateway isn't attached to the VPC, the specified NAT instance has been terminated, and so on).
- `route.vpc-peering-connection-id` - The ID of a VPC peering connection specified in a route in the table.
- `tag:key=*value*` - The key/value combination of a tag assigned to the resource.
- `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter "tag-key=Purpose" and the filter "tag-value=X", you get any resources assigned both the tag key Purpose (regardless of what the tag's value is), and the tag value X (regardless of what the tag's key is). If you want to list only resources where Purpose is X, see the `tag:key=*value*` filter.
- `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
- `vpc-id` - The ID of the VPC for the route table.

- (*dict*) -

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

Return type `list(ec2.RouteTableAssociation)`

Returns A list of `RouteTableAssociation` resources

limit (***kwargs*)

Creates an iterable up to a specified amount of `RouteTableAssociation` resources in the collection.

Request Syntax

```
route_table_association_iterator = route_table.associations.limit(  
    count=123  
)
```

Parameters `count` (*integer*) - The limit to the number of resources in the iterable.

Return type `list(ec2.RouteTableAssociation)`

Returns A list of `RouteTableAssociation` resources

page_size (***kwargs*)

Creates an iterable of all `RouteTableAssociation` resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
route_table_association_iterator = route_table.associations.page_size(  
    count=123  
)
```

Parameters `count` (*integer*) - The number of items returned by each service call

Return type `list(ec2.RouteTableAssociation)`

Returns A list of RouteTableAssociation resources

RouteTableAssociation

class `EC2.RouteTableAssociation(id)`

A resource representing an Amazon Elastic Compute Cloud (EC2) RouteTableAssociation:

```
import boto3
```

```
ec2 = boto3.resource('ec2')
```

```
route_table_association = ec2.RouteTableAssociation('id')
```

Parameters `id (string)` – The RouteTableAssociation’s id identifier. This **must** be set.

These are the resource’s available identifiers:

- `id`

These are the resource’s available attributes:

- `main`
- `route_table_association_id`
- `route_table_id`
- `subnet_id`

These are the resource’s available references:

- `route_table`
- `subnet`

These are the resource’s available actions:

- `delete()`
- `replace_subnet()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

id

(*string*) The RouteTableAssociation’s id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

main

(*boolean*)

Indicates whether this is the main route table.

route_table_association_id

(*string*)

The ID of the association between a route table and a subnet.

route_table_id

(*string*)

The ID of the route table.

subnet_id

(*string*)

The ID of the subnet. A subnet ID is not returned for an implicit association.

References

References are related resource instances that have a belongs-to relationship. For more information about references refer to the [Resources Introduction Guide](#).

route_table

(RouteTable) The related route_table if set, otherwise None.

subnet

(Subnet) The related subnet if set, otherwise None.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

delete (**kwargs)

Disassociates a subnet from a route table.

After you perform this action, the subnet no longer uses the routes in the route table. Instead, it uses the routes in the VPC's main route table. For more information about route tables, see [Route Tables](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = route_table_association.delete(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns None

replace_subnet (**kwargs)

Changes the route table associated with a given subnet in a VPC. After the operation completes, the subnet uses the routes in the new route table it's associated with. For more information about route tables, see [Route Tables](#) in the *Amazon Virtual Private Cloud User Guide*.

You can also use `ReplaceRouteTableAssociation` to change which table is the main route table in the VPC. You just specify the main route table's association ID and the route table to be the new main route table.

Request Syntax

```
route_table_association = route_table_association.replace_subnet(
    DryRun=True|False,
    RouteTableId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **RouteTableId** (*string*) – **[REQUIRED]**

The ID of the new route table to associate with the subnet.

Return type `ec2.RouteTableAssociation`

Returns RouteTableAssociation resource

SecurityGroup

class `EC2.SecurityGroup` (*id*)

A resource representing an Amazon Elastic Compute Cloud (EC2) SecurityGroup:

```
import boto3

ec2 = boto3.resource('ec2')
security_group = ec2.SecurityGroup('id')
```

Parameters *id* (*string*) – The SecurityGroup’s id identifier. This **must** be set.

These are the resource’s available identifiers:

- *id*

These are the resource’s available attributes:

- *description*
- *group_id*
- *group_name*
- *ip_permissions*
- *ip_permissions_egress*
- *owner_id*
- *tags*
- *vpc_id*

These are the resource’s available actions:

- *authorize_egress()*
- *authorize_ingress()*
- *create_tags()*
- *delete()*
- *load()*
- *reload()*
- *revoke_egress()*
- *revoke_ingress()*

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

id

(*string*) The SecurityGroup’s id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

description

(*string*)

A description of the security group.

group_id

(*string*)

The ID of the security group.

group_name

(*string*)

The name of the security group.

ip_permissions

(list)

One or more inbound rules associated with the security group.

ip_permissions_egress

(list)

[EC2-VPC] One or more outbound rules associated with the security group.

owner_id

(string)

The AWS account ID of the owner of the security group.

tags

(list)

Any tags assigned to the security group.

vpc_id

(string)

[EC2-VPC] The ID of the VPC for the security group.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

authorize_egress (**kwargs)

Adds one or more egress rules to a security group for use with a VPC. Specifically, this action permits instances to send traffic to one or more destination CIDR IP address ranges, or to one or more destination security groups for the same VPC.

Warning: You can have up to 50 rules per security group (covering both ingress and egress rules).

A security group is for use with instances either in the EC2-Classic platform or in a specific VPC. This action doesn't apply to security groups for use in EC2-Classic. For more information, see [Security Groups for Your VPC](#) in the *Amazon Virtual Private Cloud User Guide*.

Each rule consists of the protocol (for example, TCP), plus either a CIDR range or a source group. For the TCP and UDP protocols, you must also specify the destination port or port range. For the ICMP protocol, you must also specify the ICMP type and code. You can use -1 for the type or code to mean all types or all codes.

Rule changes are propagated to affected instances as quickly as possible. However, a small delay might occur.

Request Syntax

```
response = security_group.authorize_egress(
    DryRun=True|False,
    SourceSecurityGroupName='string',
    SourceSecurityGroupOwnerId='string',
    IpProtocol='string',
    FromPort=123,
    ToPort=123,
    CidrIp='string',
    IpPermissions=[
        {
```

```

'IpProtocol': 'string',
'FromPort': 123,
'ToPort': 123,
'UserIdGroupPairs': [
    {
        'UserId': 'string',
        'GroupName': 'string',
        'GroupId': 'string'
    },
],
'IpRanges': [
    {
        'CidrIp': 'string'
    },
],
'PrefixListIds': [
    {
        'PrefixListId': 'string'
    },
],
],
},
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SourceSecurityGroupName** (*string*) – The name of a destination security group. To authorize outbound access to a destination security group, we recommend that you use a set of IP permissions instead.
- **SourceSecurityGroupOwnerId** (*string*) – The AWS account number for a destination security group. To authorize outbound access to a destination security group, we recommend that you use a set of IP permissions instead.
- **IpProtocol** (*string*) – The IP protocol name (`tcp`, `udp`, `icmp`) or number (see [Protocol Numbers](#)). Use `-1` to specify all.
- **FromPort** (*integer*) – The start of port range for the TCP and UDP protocols, or an ICMP type number. For the ICMP type number, use `-1` to specify all ICMP types.
- **ToPort** (*integer*) – The end of port range for the TCP and UDP protocols, or an ICMP code number. For the ICMP code number, use `-1` to specify all ICMP codes for the ICMP type.
- **CidrIp** (*string*) – The CIDR IP address range. You can't specify this parameter when specifying a source security group.
- **IpPermissions** (*list*) – A set of IP permissions. You can't specify a destination security group and a CIDR IP address range.

– (*dict*) –

Describes a security group rule.

* **IpProtocol** (*string*) –

The protocol.

When you call `DescribeSecurityGroups`, the protocol value returned is the number. Exception: For TCP, UDP, and ICMP, the value returned is the name (for example, `tcp`, `udp`, or `icmp`). For a list

of protocol numbers, see [Protocol Numbers](#) . (VPC only) When you call `AuthorizeSecurityGroupIngress` , you can use `-1` to specify all.

* **FromPort** (*integer*) –

The start of port range for the TCP and UDP protocols, or an ICMP type number. A value of `-1` indicates all ICMP types.

* **ToPort** (*integer*) –

The end of port range for the TCP and UDP protocols, or an ICMP code. A value of `-1` indicates all ICMP codes for the specified ICMP type.

* **UserIdGroupPairs** (*list*) –

One or more security group and AWS account ID pairs.

· (*dict*) –

Describes a security group and AWS account ID pair.

· **UserId** (*string*) –

The ID of an AWS account. EC2-Classic only.

· **GroupName** (*string*) –

The name of the security group. In a request, use this parameter for a security group in EC2-Classic or a default VPC only. For a security group in a nondefault VPC, use `GroupId` .

· **GroupId** (*string*) –

The ID of the security group.

* **IpRanges** (*list*) –

One or more IP ranges.

· (*dict*) –

Describes an IP range.

· **CidrIp** (*string*) –

The CIDR range. You can either specify a CIDR range or a source security group, not both.

* **PrefixListIds** (*list*) –

(Valid for `AuthorizeSecurityGroupEgress` , `RevokeSecurityGroupEgress` and `DescribeSecurityGroups` only) One or more prefix list IDs for an AWS service. In an `AuthorizeSecurityGroupEgress` request, this is the AWS service that you want to access through a VPC endpoint from instances associated with the security group.

· (*dict*) –

The ID of the prefix.

· **PrefixListId** (*string*) –

The ID of the prefix.

Returns None

authorize_ingress (***kwargs*)

Adds one or more ingress rules to a security group.

Warning: EC2-Classic: You can have up to 100 rules per group.

EC2-VPC: You can have up to 50 rules per group (covering both ingress and egress rules).

Rule changes are propagated to instances within the security group as quickly as possible. However, a small delay might occur.

[EC2-Classic] This action gives one or more CIDR IP address ranges permission to access a security group in your account, or gives one or more security groups (called the *source groups*) permission to access a security group for your account. A source group can be for your own AWS account, or another.

[EC2-VPC] This action gives one or more CIDR IP address ranges permission to access a security group in your VPC, or gives one or more other security groups (called the *source groups*) permission to access a security group for your VPC. The security groups must all be for the same VPC.

Request Syntax

```
response = security_group.authorize_ingress(
    DryRun=True|False,
    GroupName='string',
    SourceSecurityGroupName='string',
    SourceSecurityGroupOwnerId='string',
    IpProtocol='string',
    FromPort=123,
    ToPort=123,
    CidrIp='string',
    IpPermissions=[
        {
            'IpProtocol': 'string',
            'FromPort': 123,
            'ToPort': 123,
            'UserIdGroupPairs': [
                {
                    'UserId': 'string',
                    'GroupName': 'string',
                    'GroupId': 'string'
                },
            ],
            'IpRanges': [
                {
                    'CidrIp': 'string'
                },
            ],
            'PrefixListIds': [
                {
                    'PrefixListId': 'string'
                },
            ],
        },
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **GroupName** (*string*) – [EC2-Classic, default VPC] The name of the security group.
- **SourceSecurityGroupName** (*string*) – [EC2-Classic, default VPC] The name of the source security group. You can't specify this parameter in combination with the following parameters: the CIDR IP address range, the start of the port

range, the IP protocol, and the end of the port range. For EC2-VPC, the source security group must be in the same VPC.

- **SourceSecurityGroupOwnerId** (*string*) – [EC2-Classic, default VPC] The AWS account number for the source security group. For EC2-VPC, the source security group must be in the same VPC. You can't specify this parameter in combination with the following parameters: the CIDR IP address range, the IP protocol, the start of the port range, and the end of the port range. Creates rules that grant full ICMP, UDP, and TCP access. To create a rule with a specific IP protocol and port range, use a set of IP permissions instead.
- **IpProtocol** (*string*) – The IP protocol name (`tcp`, `udp`, `icmp`) or number (see [Protocol Numbers](#)). (VPC only) Use `-1` to specify all.
- **FromPort** (*integer*) – The start of port range for the TCP and UDP protocols, or an ICMP type number. For the ICMP type number, use `-1` to specify all ICMP types.
- **ToPort** (*integer*) – The end of port range for the TCP and UDP protocols, or an ICMP code number. For the ICMP code number, use `-1` to specify all ICMP codes for the ICMP type.
- **CidrIp** (*string*) – The CIDR IP address range. You can't specify this parameter when specifying a source security group.
- **IpPermissions** (*list*) – A set of IP permissions. Can be used to specify multiple rules in a single command.
 - (*dict*) –

Describes a security group rule.

- * **IpProtocol** (*string*) –

The protocol.

When you call `DescribeSecurityGroups`, the protocol value returned is the number. Exception: For TCP, UDP, and ICMP, the value returned is the name (for example, `tcp`, `udp`, or `icmp`). For a list of protocol numbers, see [Protocol Numbers](#). (VPC only) When you call `AuthorizeSecurityGroupIngress`, you can use `-1` to specify all.

- * **FromPort** (*integer*) –

The start of port range for the TCP and UDP protocols, or an ICMP type number. A value of `-1` indicates all ICMP types.

- * **ToPort** (*integer*) –

The end of port range for the TCP and UDP protocols, or an ICMP code. A value of `-1` indicates all ICMP codes for the specified ICMP type.

- * **UserIdGroupPairs** (*list*) –

One or more security group and AWS account ID pairs.

- (*dict*) –

Describes a security group and AWS account ID pair.

- **UserId** (*string*) –

The ID of an AWS account. EC2-Classic only.

- **GroupName** (*string*) –

The name of the security group. In a request, use this parameter for a security group in EC2-Classic or a default VPC only. For a security group in a nondefault VPC, use `GroupId`.

- **GroupId** (*string*) –

The ID of the security group.

- * **IpRanges** (*list*) –

One or more IP ranges.

- (*dict*) –

Describes an IP range.

- **CidrIp** (*string*) –

The CIDR range. You can either specify a CIDR range or a source security group, not both.

- * **PrefixListIds** (*list*) –

(Valid for `AuthorizeSecurityGroupEgress`, `RevokeSecurityGroupEgress` and `DescribeSecurityGroups` only) One or more prefix list IDs for an AWS service. In an `AuthorizeSecurityGroupEgress` request, this is the AWS service that you want to access through a VPC endpoint from instances associated with the security group.

- (*dict*) –

The ID of the prefix.

- **PrefixListId** (*string*) –

The ID of the prefix.

Returns None

`create_tags` (**kwargs)

Adds or overwrites one or more tags for the specified Amazon EC2 resource or resources. Each resource can have a maximum of 10 tags. Each tag consists of a key and optional value. Tag keys must be unique per resource.

For more information about tags, see [Tagging Your Resources](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
tag = security_group.create_tags(
    DryRun=True|False,
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Tags** (*list*) – **[REQUIRED]**

One or more tags. The `value` parameter is required, but if you don't want the tag to have a value, specify the parameter with no value, and we set the value to an empty string.

- (*dict*) –

Describes a tag.

* **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

* **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

Return type `list(ec2.Tag)`

Returns A list of Tag resources

delete (***kwargs*)

Deletes a security group.

If you attempt to delete a security group that is associated with an instance, or is referenced by another security group, the operation fails with `InvalidGroup.InUse` in EC2-Classic or `DependencyViolation` in EC2-VPC.

Request Syntax

```
response = security_group.delete(
    DryRun=True|False,
    GroupName='string',
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **GroupName** (*string*) – [EC2-Classic, default VPC] The name of the security group. You can specify either the security group name or the security group ID.

Returns None

load ()

Calls `ec2.Client.describe_security_groups()` to update the attributes of the Security-Group resource

Request Syntax

```
security_group.load()
```

Returns None

reload ()

Calls `ec2.Client.describe_security_groups()` to update the attributes of the Security-Group resource

Request Syntax

```
security_group.reload()
```

Returns None

revoke_egress (**kwargs)

Removes one or more egress rules from a security group for EC2-VPC. The values that you specify in the revoke request (for example, ports) must match the existing rule's values for the rule to be revoked.

Each rule consists of the protocol and the CIDR range or source security group. For the TCP and UDP protocols, you must also specify the destination port or range of ports. For the ICMP protocol, you must also specify the ICMP type and code.

Rule changes are propagated to instances within the security group as quickly as possible. However, a small delay might occur.

Request Syntax

```
response = security_group.revoke_egress(
    DryRun=True|False,
    SourceSecurityGroupName='string',
    SourceSecurityGroupOwnerId='string',
    IpProtocol='string',
    FromPort=123,
    ToPort=123,
    CidrIp='string',
    IpPermissions=[
        {
            'IpProtocol': 'string',
            'FromPort': 123,
            'ToPort': 123,
            'UserIdGroupPairs': [
                {
                    'UserId': 'string',
                    'GroupName': 'string',
                    'GroupId': 'string'
                },
            ],
            'IpRanges': [
                {
                    'CidrIp': 'string'
                },
            ],
            'PrefixListIds': [
                {
                    'PrefixListId': 'string'
                },
            ],
        },
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SourceSecurityGroupName** (*string*) – The name of a destination security group. To revoke outbound access to a destination security group, we recommend that you use a set of IP permissions instead.
- **SourceSecurityGroupOwnerId** (*string*) – The AWS account number for a destination security group. To revoke outbound access to a destination security group, we recommend that you use a set of IP permissions instead.

- **IpProtocol** (*string*) – The IP protocol name (`tcp`, `udp`, `icmp`) or number (see [Protocol Numbers](#)). Use `-1` to specify all.
- **FromPort** (*integer*) – The start of port range for the TCP and UDP protocols, or an ICMP type number. For the ICMP type number, use `-1` to specify all ICMP types.
- **ToPort** (*integer*) – The end of port range for the TCP and UDP protocols, or an ICMP code number. For the ICMP code number, use `-1` to specify all ICMP codes for the ICMP type.
- **CidrIp** (*string*) – The CIDR IP address range. You can't specify this parameter when specifying a source security group.
- **IpPermissions** (*list*) – A set of IP permissions. You can't specify a destination security group and a CIDR IP address range.
 - (*dict*) –

Describes a security group rule.

- * **IpProtocol** (*string*) –

The protocol.

When you call `DescribeSecurityGroups`, the protocol value returned is the number. Exception: For TCP, UDP, and ICMP, the value returned is the name (for example, `tcp`, `udp`, or `icmp`). For a list of protocol numbers, see [Protocol Numbers](#). (VPC only) When you call `AuthorizeSecurityGroupIngress`, you can use `-1` to specify all.

- * **FromPort** (*integer*) –

The start of port range for the TCP and UDP protocols, or an ICMP type number. A value of `-1` indicates all ICMP types.

- * **ToPort** (*integer*) –

The end of port range for the TCP and UDP protocols, or an ICMP code. A value of `-1` indicates all ICMP codes for the specified ICMP type.

- * **UserIdGroupPairs** (*list*) –

One or more security group and AWS account ID pairs.

- (*dict*) –

Describes a security group and AWS account ID pair.

- **UserId** (*string*) –

The ID of an AWS account. EC2-Classic only.

- **GroupName** (*string*) –

The name of the security group. In a request, use this parameter for a security group in EC2-Classic or a default VPC only. For a security group in a nondefault VPC, use `GroupId`.

- **GroupId** (*string*) –

The ID of the security group.

- * **IpRanges** (*list*) –

One or more IP ranges.

- (*dict*) –

Describes an IP range.

- **CidrIp** (*string*) –

The CIDR range. You can either specify a CIDR range or a source security group, not both.

* **PrefixListIds** (*list*) –

(Valid for AuthorizeSecurityGroupEgress , RevokeSecurityGroupEgress and DescribeSecurityGroups only) One or more prefix list IDs for an AWS service. In an AuthorizeSecurityGroupEgress request, this is the AWS service that you want to access through a VPC endpoint from instances associated with the security group.

· (*dict*) –

The ID of the prefix.

· **PrefixListId** (*string*) –

The ID of the prefix.

Returns None

revoke_ingress (***kwargs*)

Removes one or more ingress rules from a security group. The values that you specify in the revoke request (for example, ports) must match the existing rule's values for the rule to be removed.

Each rule consists of the protocol and the CIDR range or source security group. For the TCP and UDP protocols, you must also specify the destination port or range of ports. For the ICMP protocol, you must also specify the ICMP type and code.

Rule changes are propagated to instances within the security group as quickly as possible. However, a small delay might occur.

Request Syntax

```
response = security_group.revoke_ingress(
    DryRun=True|False,
    GroupName='string',
    SourceSecurityGroupName='string',
    SourceSecurityGroupOwnerId='string',
    IpProtocol='string',
    FromPort=123,
    ToPort=123,
    CidrIp='string',
    IpPermissions=[
        {
            'IpProtocol': 'string',
            'FromPort': 123,
            'ToPort': 123,
            'UserIdGroupPairs': [
                {
                    'UserId': 'string',
                    'GroupName': 'string',
                    'GroupId': 'string'
                },
            ],
            'IpRanges': [
                {
                    'CidrIp': 'string'
                },
            ],
            'PrefixListIds': [
                {
                    'PrefixListId': 'string'
                },
            ],
        },
    ]
)
```

```
    },  
]  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **GroupName** (*string*) – [EC2-Classic, default VPC] The name of the security group.
- **SourceSecurityGroupName** (*string*) – [EC2-Classic, default VPC] The name of the source security group. You can't specify this parameter in combination with the following parameters: the CIDR IP address range, the start of the port range, the IP protocol, and the end of the port range. For EC2-VPC, the source security group must be in the same VPC.
- **SourceSecurityGroupOwnerId** (*string*) – [EC2-Classic, default VPC] The AWS account ID of the source security group. For EC2-VPC, the source security group must be in the same VPC. You can't specify this parameter in combination with the following parameters: the CIDR IP address range, the IP protocol, the start of the port range, and the end of the port range. To revoke a specific rule for an IP protocol and port range, use a set of IP permissions instead.
- **IpProtocol** (*string*) – The IP protocol name (`tcp`, `udp`, `icmp`) or number (see [Protocol Numbers](#)). Use `-1` to specify all.
- **FromPort** (*integer*) – The start of port range for the TCP and UDP protocols, or an ICMP type number. For the ICMP type number, use `-1` to specify all ICMP types.
- **ToPort** (*integer*) – The end of port range for the TCP and UDP protocols, or an ICMP code number. For the ICMP code number, use `-1` to specify all ICMP codes for the ICMP type.
- **CidrIp** (*string*) – The CIDR IP address range. You can't specify this parameter when specifying a source security group.
- **IpPermissions** (*list*) – A set of IP permissions. You can't specify a source security group and a CIDR IP address range.
 - (*dict*) –

Describes a security group rule.

- * **IpProtocol** (*string*) –

The protocol.

When you call `DescribeSecurityGroups`, the protocol value returned is the number. Exception: For TCP, UDP, and ICMP, the value returned is the name (for example, `tcp`, `udp`, or `icmp`). For a list of protocol numbers, see [Protocol Numbers](#). (VPC only) When you call `AuthorizeSecurityGroupIngress`, you can use `-1` to specify all.

- * **FromPort** (*integer*) –

The start of port range for the TCP and UDP protocols, or an ICMP type number. A value of `-1` indicates all ICMP types.

- * **ToPort** (*integer*) –

The end of port range for the TCP and UDP protocols, or an ICMP code. A value of `-1` indicates all ICMP codes for the specified ICMP type.

- * **UserIdGroupPairs** (*list*) –

One or more security group and AWS account ID pairs.

- *(dict)* –

Describes a security group and AWS account ID pair.

- **UserId** (*string*) –

The ID of an AWS account. EC2-Classic only.

- **GroupName** (*string*) –

The name of the security group. In a request, use this parameter for a security group in EC2-Classic or a default VPC only. For a security group in a nondefault VPC, use `GroupId`.

- **GroupId** (*string*) –

The ID of the security group.

- * **IpRanges** (*list*) –

One or more IP ranges.

- *(dict)* –

Describes an IP range.

- **CidrIp** (*string*) –

The CIDR range. You can either specify a CIDR range or a source security group, not both.

- * **PrefixListIds** (*list*) –

(Valid for `AuthorizeSecurityGroupEgress`, `RevokeSecurityGroupEgress` and `DescribeSecurityGroups` only) One or more prefix list IDs for an AWS service. In an `AuthorizeSecurityGroupEgress` request, this is the AWS service that you want to access through a VPC endpoint from instances associated with the security group.

- *(dict)* –

The ID of the prefix.

- **PrefixListId** (*string*) –

The ID of the prefix.

Returns None

Snapshot

class `EC2.Snapshot` (*id*)

A resource representing an Amazon Elastic Compute Cloud (EC2) Snapshot:

```
import boto3
```

```
ec2 = boto3.resource('ec2')
snapshot = ec2.Snapshot('id')
```

Parameters **id** (*string*) – The Snapshot's id identifier. This **must** be set.

These are the resource's available identifiers:

- `id`

These are the resource's available attributes:

- `data_encryption_key_id`
- `description`
- `encrypted`

- kms_key_id
- owner_alias
- owner_id
- progress
- snapshot_id
- start_time
- state
- state_message
- tags
- volume_id
- volume_size

These are the resource's available references:

- volume

These are the resource's available actions:

- copy()
- create_tags()
- delete()
- describe_attribute()
- load()
- modify_attribute()
- reload()
- reset_attribute()

These are the resource's available waiters:

- wait_until_completed()

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

id

(string) The Snapshot's id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

data_encryption_key_id

(string)

The data encryption key identifier for the snapshot. This value is a unique identifier that corresponds to the data encryption key that was used to encrypt the original volume or snapshot copy. Because data encryption keys are inherited by volumes created from snapshots, and vice versa, if snapshots share the same data encryption key identifier, then they belong to the same volume/snapshot lineage. This parameter is only returned by the DescribeSnapshots API operation.

description

(string)

The description for the snapshot.

encrypted

(boolean)

Indicates whether the snapshot is encrypted.

kms_key_id

(string)

The full ARN of the AWS Key Management Service (AWS KMS) customer master key (CMK) that was used to protect the volume encryption key for the parent volume.

owner_alias

(string)

The AWS account alias (for example, `amazon` , `self`) or AWS account ID that owns the snapshot.

owner_id

(string)

The AWS account ID of the EBS snapshot owner.

progress

(string)

The progress of the snapshot, as a percentage.

snapshot_id

(string)

The ID of the snapshot. Each snapshot receives a unique identifier when it is created.

start_time

(datetime)

The time stamp when the snapshot was initiated.

state

(string)

The snapshot state.

state_message

(string)

Encrypted Amazon EBS snapshots are copied asynchronously. If a snapshot copy operation fails (for example, if the proper AWS Key Management Service (AWS KMS) permissions are not obtained) this field displays error state details to help you diagnose why the error occurred. This parameter is only returned by the `DescribeSnapshots` API operation.

tags

(list)

Any tags assigned to the snapshot.

volume_id

(string)

The ID of the volume that was used to create the snapshot.

volume_size

(integer)

The size of the volume, in GiB.

References

References are related resource instances that have a belongs-to relationship. For more information about references refer to the [Resources Introduction Guide](#).

volume

(Volume) The related volume if set, otherwise `None`.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

`copy` (***kwargs*)

Copies a point-in-time snapshot of an EBS volume and stores it in Amazon S3. You can copy the snapshot within the same region or from one region to another. You can use the snapshot to create EBS volumes or Amazon Machine Images (AMIs). The snapshot is copied to the regional endpoint that you send the HTTP request to.

Copies of encrypted EBS snapshots remain encrypted. Copies of unencrypted snapshots remain unencrypted, unless the `Encrypted` flag is specified during the snapshot copy operation. By default, encrypted snapshot copies use the default AWS Key Management Service (AWS KMS) customer master key (CMK); however, you can specify a non-default CMK with the `KmsKeyId` parameter.

For more information, see [Copying an Amazon EBS Snapshot](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = snapshot.copy(  
    DryRun=True|False,  
    SourceRegion='string',  
    Description='string',  
    Encrypted=True|False,  
    KmsKeyId='string'  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SourceRegion** (*string*) – **[REQUIRED]**
The ID of the region that contains the snapshot to be copied.
- **Description** (*string*) – A description for the EBS snapshot.
- **DestinationRegion** (*string*) – The destination region to use in the `PresignedUrl` parameter of a snapshot copy operation. This parameter is only valid for specifying the destination region in a `PresignedUrl` parameter, where it is required.

Note: `CopySnapshot` sends the snapshot copy to the regional endpoint that you send the HTTP request to, such as `ec2.us-east-1.amazonaws.com` (in the AWS CLI, this is specified with the `--region` parameter or the default region in your AWS configuration file).

Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required

- **PresignedUrl** (*string*) – The pre-signed URL that facilitates copying an encrypted snapshot. This parameter is only required when copying an encrypted snapshot with the Amazon EC2 Query API; it is available as an optional parameter in all other cases. The `PresignedUrl` should use the snapshot source endpoint, the `CopySnapshot` action, and include the `SourceRegion`, `SourceSnapshotId`, and `DestinationRegion` parameters. The `PresignedUrl` must be signed using AWS Signature Version 4. Because EBS snapshots are stored in Amazon S3, the signing algorithm for this parameter uses

the same logic that is described in [Authenticating Requests by Using Query Parameters \(AWS Signature Version 4\)](#) in the *Amazon Simple Storage Service API Reference* . An invalid or improperly signed `PresignedUrl` will cause the copy operation to fail asynchronously, and the snapshot will move to an `error` state.

Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required

- **Encrypted** (*boolean*) – Specifies whether the destination snapshot should be encrypted. There is no way to create an unencrypted snapshot copy from an encrypted snapshot; however, you can encrypt a copy of an unencrypted snapshot with this flag. The default CMK for EBS is used unless a non-default AWS Key Management Service (AWS KMS) CMK is specified with `KmsKeyId` . For more information, see [Amazon EBS Encryption](#) in the *Amazon Elastic Compute Cloud User Guide* .
- **KmsKeyId** (*string*) – The full ARN of the AWS Key Management Service (AWS KMS) CMK to use when creating the snapshot copy. This parameter is only required if you want to use a non-default CMK; if this parameter is not specified, the default CMK for EBS is used. The ARN contains the `arn:aws:kms` namespace, followed by the region of the CMK, the AWS account ID of the CMK owner, the `key` namespace, and then the CMK ID. For example, `arn:aws:kms:us-east-1:012345678910:key/abcd1234-a123-456a-a12b-a123b4cd56ef` . The specified CMK must exist in the region that the snapshot is being copied to. If a `KmsKeyId` is specified, the `Encrypted` flag must also be set.

Return type dict

Returns

Response Syntax

```
{
    'SnapshotId': 'string'
}
```

Response Structure

- (*dict*) –
 - **SnapshotId** (*string*) –
The ID of the new snapshot.

create_tags (***kwargs*)

Adds or overwrites one or more tags for the specified Amazon EC2 resource or resources. Each resource can have a maximum of 10 tags. Each tag consists of a key and optional value. Tag keys must be unique per resource.

For more information about tags, see [Tagging Your Resources](#) in the *Amazon Elastic Compute Cloud User Guide* .

Request Syntax

```
tag = snapshot.create_tags(
    DryRun=True|False,
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
```

```
]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Tags** (*list*) – **[REQUIRED]**

One or more tags. The `value` parameter is required, but if you don't want the tag to have a value, specify the parameter with no value, and we set the value to an empty string.

– (*dict*) –

Describes a tag.

* **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

* **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

Return type `list(ec2.Tag)`

Returns A list of Tag resources

delete (***kwargs*)

Deletes the specified snapshot.

When you make periodic snapshots of a volume, the snapshots are incremental, and only the blocks on the device that have changed since your last snapshot are saved in the new snapshot. When you delete a snapshot, only the data not needed for any other snapshot is removed. So regardless of which prior snapshots have been deleted, all active snapshots will have access to all the information needed to restore the volume.

You cannot delete a snapshot of the root device of an EBS volume used by a registered AMI. You must first de-register the AMI before you can delete the snapshot.

For more information, see [Deleting an Amazon EBS Snapshot](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = snapshot.delete(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns `None`

describe_attribute (**kwargs)

Describes the specified attribute of the specified snapshot. You can specify only one attribute at a time.

For more information about EBS snapshots, see [Amazon EBS Snapshots](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = snapshot.describe_attribute(
    DryRun=True|False,
    Attribute='productCodes'|'createVolumePermission'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Attribute** (*string*) – **[REQUIRED]**

The snapshot attribute you would like to view.

Return type dict

Returns**Response Syntax**

```
{
    'SnapshotId': 'string',
    'CreateVolumePermissions': [
        {
            'UserId': 'string',
            'Group': 'all'
        },
    ],
    'ProductCodes': [
        {
            'ProductCodeId': 'string',
            'ProductCodeType': 'devpay'|'marketplace'
        },
    ]
}
```

Response Structure

- (*dict*) –
 - **SnapshotId** (*string*) –

The ID of the EBS snapshot.
 - **CreateVolumePermissions** (*list*) –

A list of permissions for creating volumes from the snapshot.

 - * (*dict*) –

Describes the user or group to be added or removed from the permissions for a volume.

 - **UserId** (*string*) –

The specific AWS account ID that is to be added or removed from a volume's list of create volume permissions.

- **Group** (*string*) –
The specific group that is to be added or removed from a volume's list of create volume permissions.
- **ProductCodes** (*list*) –
A list of product codes.
 - * (*dict*) –
Describes a product code.
 - **ProductCodeId** (*string*) –
The product code.
 - **ProductCodeType** (*string*) –
The type of product code.

load()

Calls `ec2.Client.describe_snapshots()` to update the attributes of the Snapshot resource

Request Syntax

```
snapshot.load()
```

Returns None

modify_attribute (***kwargs*)

Adds or removes permission settings for the specified snapshot. You may add or remove specified AWS account IDs from a snapshot's list of create volume permissions, but you cannot do both in a single API call. If you need to both add and remove account IDs for a snapshot, you must use multiple API calls.

For more information on modifying snapshot permissions, see [Sharing Snapshots](#) in the *Amazon Elastic Compute Cloud User Guide*.

Note: Snapshots with AWS Marketplace product codes cannot be made public.

Request Syntax

```
response = snapshot.modify_attribute(  
    DryRun=True|False,  
    Attribute='productCodes'|'createVolumePermission',  
    OperationType='add'|'remove',  
    UserIds=[  
        'string',  
    ],  
    GroupNames=[  
        'string',  
    ],  
    CreateVolumePermission={  
        'Add': [  
            {  
                'UserId': 'string',  
                'Group': 'all'  
            },  
        ],  
        'Remove': [  
            {  
                'UserId': 'string',  
                'Group': 'all'  
            },  
        ],  
    },  
)
```

```
]
}
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Attribute** (*string*) – The snapshot attribute to modify.

Note: Only volume creation permissions may be modified at the customer level.

- **OperationType** (*string*) – The type of operation to perform to the attribute.
- **UserIds** (*list*) – The account ID to modify for the snapshot.
 - (*string*) –
- **GroupNames** (*list*) – The group to modify for the snapshot.
 - (*string*) –
- **CreateVolumePermission** (*dict*) – A JSON representation of the snapshot attribute modification.
 - **Add** (*list*) –

Adds a specific AWS account ID or group to a volume's list of create volume permissions.

* (*dict*) –

Describes the user or group to be added or removed from the permissions for a volume.

· **UserId** (*string*) –

The specific AWS account ID that is to be added or removed from a volume's list of create volume permissions.

· **Group** (*string*) –

The specific group that is to be added or removed from a volume's list of create volume permissions.

– **Remove** (*list*) –

Removes a specific AWS account ID or group from a volume's list of create volume permissions.

* (*dict*) –

Describes the user or group to be added or removed from the permissions for a volume.

· **UserId** (*string*) –

The specific AWS account ID that is to be added or removed from a volume's list of create volume permissions.

· **Group** (*string*) –

The specific group that is to be added or removed from a volume's list of create volume permissions.

Returns None

reload()

Calls `ec2.Client.describe_snapshots()` to update the attributes of the Snapshot resource

Request Syntax

```
snapshot.reload()
```

Returns None

reset_attribute (**kwargs)

Resets permission settings for the specified snapshot.

For more information on modifying snapshot permissions, see [Sharing Snapshots](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = snapshot.reset_attribute(  
    DryRun=True|False,  
    Attribute='productCodes' | 'createVolumePermission'  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Attribute** (*string*) – **[REQUIRED]**
The attribute to reset. Currently, only the attribute for permission to create volumes can be reset.

Returns None

Waiters

Waiters provide an interface to wait for a resource to reach a specific state. For more information about waiters refer to the [Resources Introduction Guide](#).

wait_until_completed (**kwargs)

Waits until this Snapshot is completed. This method calls `EC2.Waiter.snapshot_completed.wait()` which polls `EC2.Client.describe_snapshots()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
snapshot.wait_until_completed(  
    DryRun=True|False,  
    OwnerIds=[  
        'string',  
    ],  
    RestorableByUserIds=[  
        'string',  
    ],  
    Filters=[  
        {  
            'Name': 'string',  
            'Values': [  
                'string',  
            ]  
        },  
    ],  
    NextToken='string',  
    MaxResults=123  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **OwnerIds** (*list*) – Returns the snapshots owned by the specified owner. Multiple owners can be specified.
 - (*string*) –
- **RestorableByUserIds** (*list*) – One or more AWS accounts IDs that can create volumes from the snapshot.
 - (*string*) –
- **Filters** (*list*) – One or more filters.
 - `description` - A description of the snapshot.
 - `owner-alias` - The AWS account alias (for example, `amazon`) that owns the snapshot.
 - `owner-id` - The ID of the AWS account that owns the snapshot.
 - `progress` - The progress of the snapshot, as a percentage (for example, 80%).
 - `snapshot-id` - The snapshot ID.
 - `start-time` - The time stamp when the snapshot was initiated.
 - `status` - The status of the snapshot (`pending` | `completed` | `error`).
 - `tag :key =*value*` - The key/value combination of a tag assigned to the resource.
 - `tag-key` - The key of a tag assigned to the resource. This filter is independent of the `tag-value` filter. For example, if you use both the filter “`tag-key=Purpose`” and the filter “`tag-value=X`”, you get any resources assigned both the tag key `Purpose` (regardless of what the tag’s value is), and the tag value `X` (regardless of what the tag’s key is). If you want to list only resources where `Purpose` is `X`, see the `tag :key =*value*` filter.
 - `tag-value` - The value of a tag assigned to the resource. This filter is independent of the `tag-key` filter.
 - `volume-id` - The ID of the volume the snapshot is for.
 - `volume-size` - The size of the volume, in GiB.
 - (*dict*) –

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) –

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) –

One or more filter values. Filter values are case-sensitive.

· (*string*) –

- **NextToken** (*string*) – The `NextToken` value returned from a previous paginated `DescribeSnapshots` request where `MaxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `NextToken` value. This value is `null` when there are no more results to return.
- **MaxResults** (*integer*) – The maximum number of snapshot results returned by `DescribeSnapshots` in paginated output. When this parameter is used, `DescribeSnapshots` only returns `MaxResults` results in a single page along with a `NextToken` response element. The remaining results of the ini-

tial request can be seen by sending another `DescribeSnapshots` request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. If this parameter is not used, then `DescribeSnapshots` returns all results. You cannot specify this parameter and the snapshot IDs parameter in the same request.

Returns None

Subnet

class `EC2.Subnet` (*id*)

A resource representing an Amazon Elastic Compute Cloud (EC2) Subnet:

```
import boto3
```

```
ec2 = boto3.resource('ec2')
subnet = ec2.Subnet('id')
```

Parameters *id* (*string*) – The Subnet’s id identifier. This **must** be set.

These are the resource’s available identifiers:

- `id`

These are the resource’s available attributes:

- `availability_zone`
- `available_ip_address_count`
- `cidr_block`
- `default_for_az`
- `map_public_ip_on_launch`
- `state`
- `subnet_id`
- `tags`
- `vpc_id`

These are the resource’s available references:

- `vpc`

These are the resource’s available actions:

- `create_instances()`
- `create_network_interface()`
- `create_tags()`
- `delete()`
- `load()`
- `reload()`

These are the resource’s available collections:

- `instances`
- `network_interfaces`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

id

(*string*) The Subnet’s id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

availability_zone*(string)*

The Availability Zone of the subnet.

available_ip_address_count*(integer)*

The number of unused IP addresses in the subnet. Note that the IP addresses for any stopped instances are considered unavailable.

cidr_block*(string)*

The CIDR block assigned to the subnet.

default_for_az*(boolean)*

Indicates whether this is the default subnet for the Availability Zone.

map_public_ip_on_launch*(boolean)*

Indicates whether instances launched in this subnet receive a public IP address.

state*(string)*

The current state of the subnet.

subnet_id*(string)*

The ID of the subnet.

tags*(list)*

Any tags assigned to the subnet.

vpc_id*(string)*

The ID of the VPC the subnet is in.

References

References are related resource instances that have a belongs-to relationship. For more information about references refer to the [Resources Introduction Guide](#).

vpc

(Vpc) The related vpc if set, otherwise None.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

create_instances (***kwargs*)

Launches the specified number of instances using an AMI for which you have permissions.

When you launch an instance, it enters the `pending` state. After the instance is ready for you, it enters the `running` state. To check the state of your instance, call `DescribeInstances`.

If you don't specify a security group when launching an instance, Amazon EC2 uses the default security group. For more information, see [Security Groups](#) in the *Amazon Elastic Compute Cloud User Guide* .

[EC2-VPC only accounts] If you don't specify a subnet in the request, we choose a default subnet from your default VPC for you.

[EC2-Classic accounts] If you're launching into EC2-Classic and you don't specify an Availability Zone, we choose one for you.

Linux instances have access to the public key of the key pair at boot. You can use this key to provide secure access to the instance. Amazon EC2 public images use this feature to provide secure access without passwords. For more information, see [Key Pairs](#) in the *Amazon Elastic Compute Cloud User Guide* .

You can provide optional user data when launching an instance. For more information, see [Instance Metadata](#) in the *Amazon Elastic Compute Cloud User Guide* .

If any of the AMIs have a product code attached for which the user has not subscribed, `RunInstances` fails.

T2 instance types can only be launched into a VPC. If you do not have a default VPC, or if you do not specify a subnet ID in the request, `RunInstances` fails.

For more information about troubleshooting, see [What To Do If An Instance Immediately Terminates](#) , and [Troubleshooting Connecting to Your Instance](#) in the *Amazon Elastic Compute Cloud User Guide* .

Request Syntax

```
instance = subnet.create_instances(  
    DryRun=True|False,  
    ImageId='string',  
    MinCount=123,  
    MaxCount=123,  
    KeyName='string',  
    SecurityGroups=[  
        'string',  
    ],  
    SecurityGroupIds=[  
        'string',  
    ],  
    UserData='string',  
    InstanceType='t1.micro'|'m1.small'|'m1.medium'|'m1.large'|'m1.xlarge'|'m3.medium'|'m3.1',  
    Placement={  
        'AvailabilityZone': 'string',  
        'GroupName': 'string',  
        'Tenancy': 'default'|'dedicated'  
    },  
    KernelId='string',  
    RamdiskId='string',  
    BlockDeviceMappings=[  
        {  
            'VirtualName': 'string',  
            'DeviceName': 'string',  
            'Ebs': {  
                'SnapshotId': 'string',  
                'VolumeSize': 123,  
                'DeleteOnTermination': True|False,  
                'VolumeType': 'standard'|'io1'|'gp2',  
                'Iops': 123,  
                'Encrypted': True|False
```

```

        },
        'NoDevice': 'string'
    },
],
Monitoring={
    'Enabled': True|False
},
DisableApiTermination=True|False,
InstanceInitiatedShutdownBehavior='stop'|'terminate',
PrivateIpAddress='string',
ClientToken='string',
AdditionalInfo='string',
NetworkInterfaces=[
    {
        'NetworkInterfaceId': 'string',
        'DeviceIndex': 123,
        'SubnetId': 'string',
        'Description': 'string',
        'PrivateIpAddress': 'string',
        'Groups': [
            'string',
        ],
        'DeleteOnTermination': True|False,
        'PrivateIpAddresses': [
            {
                'PrivateIpAddress': 'string',
                'Primary': True|False
            },
        ],
        'SecondaryPrivateIpAddressCount': 123,
        'AssociatePublicIpAddress': True|False
    },
],
IamInstanceProfile={
    'Arn': 'string',
    'Name': 'string'
},
EbsOptimized=True|False
)

```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **ImageId** (*string*) – **[REQUIRED]**

The ID of the AMI, which you can get by calling `DescribeImages`.

- **MinCount** (*integer*) – **[REQUIRED]**

The minimum number of instances to launch. If you specify a minimum that is more instances than Amazon EC2 can launch in the target Availability Zone, Amazon EC2 launches no instances.

Constraints: Between 1 and the maximum number you're allowed for the specified instance type. For more information about the default limits, and how to request an increase, see [How many instances can I run in Amazon EC2](#) in the Amazon EC2 General FAQ.

- **MaxCount** (*integer*) – [REQUIRED]

The maximum number of instances to launch. If you specify more instances than Amazon EC2 can launch in the target Availability Zone, Amazon EC2 launches the largest possible number of instances above `MinCount` .

Constraints: Between 1 and the maximum number you’re allowed for the specified instance type. For more information about the default limits, and how to request an increase, see [How many instances can I run in Amazon EC2](#) in the Amazon EC2 General FAQ.

- **KeyName** (*string*) – The name of the key pair. You can create a key pair using `CreateKeyPair` or `ImportKeyPair` .

Warning: If you do not specify a key pair, you can’t connect to the instance unless you choose an AMI that is configured to allow users another way to log in.

- **SecurityGroups** (*list*) – [EC2-Classic, default VPC] One or more security group names. For a nondefault VPC, you must use security group IDs instead.

Default: Amazon EC2 uses the default security group.

– (*string*) –

- **SecurityGroupIds** (*list*) – One or more security group IDs. You can create a security group using `CreateSecurityGroup` .

Default: Amazon EC2 uses the default security group.

– (*string*) –

- **UserData** (*string*) – The Base64-encoded MIME user data for the instances.

This value will be base64 encoded automatically. Do not base64 encode this value prior to performing the operation.

- **InstanceType** (*string*) – The instance type. For more information, see [Instance Types](#) in the *Amazon Elastic Compute Cloud User Guide* .

Default: `m1.small`

- **Placement** (*dict*) – The placement for the instance.

– **AvailabilityZone** (*string*) –

The Availability Zone of the instance.

– **GroupName** (*string*) –

The name of the placement group the instance is in (for cluster compute instances).

– **Tenancy** (*string*) –

The tenancy of the instance (if the instance is running in a VPC). An instance with a tenancy of `dedicated` runs on single-tenant hardware.

- **KernelId** (*string*) – The ID of the kernel.

Warning: We recommend that you use PV-GRUB instead of kernels and RAM disks. For more information, see [PV-GRUB](#) in the *Amazon Elastic Compute Cloud User Guide* .

- **RamdiskId** (*string*) – The ID of the RAM disk.

Warning: We recommend that you use PV-GRUB instead of kernels and RAM disks. For more information, see [PV-GRUB](#) in the *Amazon Elastic Compute Cloud User Guide* .

- **BlockDeviceMappings** (*list*) – The block device mapping.

– (*dict*) –

Describes a block device mapping.

* **VirtualName** (*string*) –

The virtual device name (ephemeral N). Instance store volumes are numbered starting from 0. An instance type with 2 available instance store volumes can specify mappings for `ephemeral0` and `ephemeral1`. The number of available instance store volumes depends on the instance type. After you connect to the instance, you must mount the volume.

Constraints: For M3 instances, you must specify instance store volumes in the block device mapping for the instance. When you launch an M3 instance, we ignore any instance store volumes specified in the block device mapping for the AMI.

* **DeviceName** (*string*) –

The device name exposed to the instance (for example, `/dev/sdh` or `xvdh`).

* **Ebs** (*dict*) –

Parameters used to automatically set up EBS volumes when the instance is launched.

· **SnapshotId** (*string*) –

The ID of the snapshot.

· **VolumeSize** (*integer*) –

The size of the volume, in GiB.

Constraints: 1–1024 for standard volumes, 1–16384 for `gp2` volumes, and 4–16384 for `io1` volumes. If you specify a snapshot, the volume size must be equal to or larger than the snapshot size.

Default: If you’re creating the volume from a snapshot and don’t specify a volume size, the default is the snapshot size.

· **DeleteOnTermination** (*boolean*) –

Indicates whether the EBS volume is deleted on instance termination.

· **VolumeType** (*string*) –

The volume type. `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, and `standard` for Magnetic volumes.

Default: `standard`

· **Iops** (*integer*) –

The number of I/O operations per second (IOPS) that the volume supports. For Provisioned IOPS (SSD) volumes, this represents the number of IOPS that are provisioned for the volume. For General Purpose (SSD) volumes, this represents the baseline performance of the volume and the rate at which the volume accumulates I/O credits for bursting. For more information on General Purpose (SSD) baseline performance, I/O credits, and bursting, see [Amazon EBS Volume Types](#) in the *Amazon Elastic Compute Cloud User Guide*.

Constraint: Range is 100 to 20000 for Provisioned IOPS (SSD) volumes and 3 to 10000 for General Purpose (SSD) volumes.

Condition: This parameter is required for requests to create `io1` volumes; it is not used in requests to create standard or `gp2` volumes.

- **Encrypted** (*boolean*) –

Indicates whether the EBS volume is encrypted. Encrypted Amazon EBS volumes may only be attached to instances that support Amazon EBS encryption.

- * **NoDevice** (*string*) –

Suppresses the specified device included in the block device mapping of the AMI.

- **Monitoring** (*dict*) – The monitoring for the instance.

- **Enabled** (*boolean*) – [REQUIRED]

Indicates whether monitoring is enabled for the instance.

- **DisableApiTermination** (*boolean*) – If you set this parameter to `true`, you can't terminate the instance using the Amazon EC2 console, CLI, or API; otherwise, you can. If you set this parameter to `true` and then later want to be able to terminate the instance, you must first change the value of the `disableApiTermination` attribute to `false` using `ModifyInstanceAttribute`. Alternatively, if you set `InstanceInitiatedShutdownBehavior` to `terminate`, you can terminate the instance by running the shutdown command from the instance.

Default: `false`

- **InstanceInitiatedShutdownBehavior** (*string*) – Indicates whether an instance stops or terminates when you initiate shutdown from the instance (using the operating system command for system shutdown).

Default: `stop`

- **PrivateIpAddress** (*string*) – [EC2-VPC] The primary IP address. You must specify a value from the IP address range of the subnet.

Only one private IP address can be designated as primary. Therefore, you can't specify this parameter if `PrivateIpAddresses.n.Primary` is set to `true` and `PrivateIpAddresses.n.PrivateIpAddress` is set to an IP address.

Default: We select an IP address from the IP address range of the subnet.

- **ClientToken** (*string*) – Unique, case-sensitive identifier you provide to ensure the idempotency of the request. For more information, see [Ensuring Idempotency](#).

Constraints: Maximum 64 ASCII characters

- **AdditionalInfo** (*string*) – Reserved.

- **NetworkInterfaces** (*list*) – One or more network interfaces.
 - (*dict*) –

Describes a network interface.

- * **NetworkInterfaceId** (*string*) –

The ID of the network interface.

- * **DeviceIndex** (*integer*) –

The index of the device on the instance for the network interface attachment. If you are specifying a network interface in a `RunInstances` request, you must provide the device index.

* **SubnetId** (*string*) –

The ID of the subnet associated with the network string. Applies only if creating a network interface when launching an instance.

* **Description** (*string*) –

The description of the network interface. Applies only if creating a network interface when launching an instance.

* **PrivateIpAddress** (*string*) –

The private IP address of the network interface. Applies only if creating a network interface when launching an instance.

* **Groups** (*list*) –

The IDs of the security groups for the network interface. Applies only if creating a network interface when launching an instance.

· (*string*) –

* **DeleteOnTermination** (*boolean*) –

If set to `true`, the interface is deleted when the instance is terminated. You can specify `true` only if creating a new network interface when launching an instance.

* **PrivateIpAddresses** (*list*) –

One or more private IP addresses to assign to the network interface. Only one private IP address can be designated as primary.

· (*dict*) –

Describes a secondary private IP address for a network interface.

· **PrivateIpAddress** (*string*) – [REQUIRED]

The private IP addresses.

· **Primary** (*boolean*) –

Indicates whether the private IP address is the primary private IP address. Only one IP address can be designated as primary.

* **SecondaryPrivateIpAddressCount** (*integer*) –

The number of secondary private IP addresses. You can't specify this option and specify more than one private IP address using the private IP addresses option.

* **AssociatePublicIpAddress** (*boolean*) –

Indicates whether to assign a public IP address to an instance you launch in a VPC. The public IP address can only be assigned to a network interface for `eth0`, and can only be assigned to a new network interface, not an existing one. You cannot specify more than one network interface in the request. If launching into a default subnet, the default value is `true`.

• **IamInstanceProfile** (*dict*) – The IAM instance profile.

– **Arn** (*string*) –

The Amazon Resource Name (ARN) of the instance profile.

– **Name** (*string*) –

The name of the instance profile.

- **EbsOptimized** (*boolean*) – Indicates whether the instance is optimized for EBS I/O. This optimization provides dedicated throughput to Amazon EBS and an optimized configuration stack to provide optimal EBS I/O performance. This optimization isn't available with all instance types. Additional usage charges apply when using an EBS-optimized instance.

Default: `false`

Return type `list(ec2.Instance)`

Returns A list of Instance resources

create_network_interface (***kwargs*)

Creates a network interface in the specified subnet.

For more information about network interfaces, see [Elastic Network Interfaces](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
network_interface = subnet.create_network_interface(
    Description='string',
    PrivateIpAddress='string',
    Groups=[
        'string',
    ],
    PrivateIpAddresses=[
        {
            'PrivateIpAddress': 'string',
            'Primary': True|False
        },
    ],
    SecondaryPrivateIpAddressCount=123,
    DryRun=True|False
)
```

Parameters

- **Description** (*string*) – A description for the network interface.
- **PrivateIpAddress** (*string*) – The primary private IP address of the network interface. If you don't specify an IP address, Amazon EC2 selects one for you from the subnet range. If you specify an IP address, you cannot indicate any IP addresses specified in `privateIpAddresses` as primary (only one IP address can be designated as primary).
- **Groups** (*list*) – The IDs of one or more security groups.
 - (*string*) –
- **PrivateIpAddresses** (*list*) – One or more private IP addresses.
 - (*dict*) –

Describes a secondary private IP address for a network interface.

* **PrivateIpAddress** (*string*) – **[REQUIRED]**

The private IP addresses.

* **Primary** (*boolean*) –

Indicates whether the private IP address is the primary private IP address. Only one IP address can be designated as primary.

- **SecondaryPrivateIpAddressCount** (*integer*) – The number of secondary private IP addresses to assign to a network interface. When you specify a number of secondary IP addresses, Amazon EC2 selects these IP addresses within the subnet range. You can't specify this option and specify more than one private IP address using `privateIpAddresses`.

The number of IP addresses you can assign to a network interface varies by instance type. For more information, see [Private IP Addresses Per ENI Per Instance Type](#) in the *Amazon Elastic Compute Cloud User Guide*.

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type `ec2.NetworkInterface`

Returns `NetworkInterface` resource

create_tags (***kwargs*)

Adds or overwrites one or more tags for the specified Amazon EC2 resource or resources. Each resource can have a maximum of 10 tags. Each tag consists of a key and optional value. Tag keys must be unique per resource.

For more information about tags, see [Tagging Your Resources](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
tag = subnet.create_tags(
    DryRun=True|False,
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **Tags** (*list*) – **[REQUIRED]**

One or more tags. The `value` parameter is required, but if you don't want the tag to have a value, specify the parameter with no value, and we set the value to an empty string.

– (*dict*) –

Describes a tag.

* **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

* **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

Return type `list(ec2.Tag)`

Returns A list of Tag resources

delete (**kwargs)

Deletes the specified subnet. You must terminate all running instances in the subnet before you can delete the subnet.

Request Syntax

```
response = subnet.delete(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns None

load()

Calls `ec2.Client.describe_subnets()` to update the attributes of the Subnet resource

Request Syntax

```
subnet.load()
```

Returns None

reload()

Calls `ec2.Client.describe_subnets()` to update the attributes of the Subnet resource

Request Syntax

```
subnet.reload()
```

Returns None

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

instances

all()

Creates an iterable of all Instance resources in the collection.

Request Syntax

```
instance_iterator = subnet.instances.all()
```

Return type `list(ec2.Instance)`

Returns A list of Instance resources

create_tags (**kwargs)

Adds or overwrites one or more tags for the specified Amazon EC2 resource or resources. Each resource can have a maximum of 10 tags. Each tag consists of a key and optional value. Tag keys must be unique per resource.

For more information about tags, see [Tagging Your Resources](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = subnet.instances.create_tags(
    DryRun=True|False,
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Tags** (*list*) – **[REQUIRED]**

One or more tags. The `value` parameter is required, but if you don't want the tag to have a value, specify the parameter with no value, and we set the value to an empty string.

– (*dict*) –

Describes a tag.

* **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

* **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

Returns None

filter (***kwargs*)

Creates an iterable of all Instance resources in the collection filtered by kwargs passed to method.

Request Syntax

```
instance_iterator = subnet.instances.filter(
    DryRun=True|False,
    InstanceIds=[
        'string',
    ],
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **InstanceIds** (*list*) – One or more instance IDs.

Default: Describes all your instances.

– (*string*) –

- **NextToken** (*string*) – The token to request the next page of results.
- **MaxResults** (*integer*) – The maximum number of results to return for the request in a single page. The remaining results of the initial request can be seen by sending another request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. You cannot specify this parameter and the instance IDs parameter in the same request.

Return type `list(ec2.Instance)`

Returns A list of Instance resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Instance resources in the collection.

Request Syntax

```
instance_iterator = subnet.instances.limit(  
    count=123  
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(ec2.Instance)`

Returns A list of Instance resources

monitor (***kwargs*)

Enables monitoring for a running instance. For more information about monitoring instances, see [Monitoring Your Instances and Volumes](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = subnet.instances.monitor(  
    DryRun=True|False,  
)
```

Parameters `DryRun` (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type `dict`

Returns

Response Syntax

```
{  
    'InstanceMonitorings': [  
        {  
            'InstanceId': 'string',  
            'Monitoring': {  
                'State': 'disabled'|'disabling'|'enabled'|'pending'  
            }  
        },  
    ],  
}
```

Response Structure

- *(dict)* –
 - **InstanceMonitorings** (*list*) –

Monitoring information for one or more instances.

 - * *(dict)* –

Describes the monitoring information of the instance.

 - **InstanceId** (*string*) –

The ID of the instance.
 - **Monitoring** (*dict*) –

The monitoring information.
 - **State** (*string*) –

Indicates whether monitoring is enabled for the instance.

page_size (***kwargs*)

Creates an iterable of all Instance resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
instance_iterator = subnet.instances.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(ec2.Instance)`

Returns A list of Instance resources

reboot (***kwargs*)

Requests a reboot of one or more instances. This operation is asynchronous; it only queues a request to reboot the specified instances. The operation succeeds if the instances are valid and belong to you. Requests to reboot terminated instances are ignored.

If a Linux/Unix instance does not cleanly shut down within four minutes, Amazon EC2 performs a hard reboot.

For more information about troubleshooting, see [Getting Console Output and Rebooting Instances](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = subnet.instances.reboot(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns `None`

start (***kwargs*)

Starts an Amazon EBS-backed AMI that you've previously stopped.

Instances that use Amazon EBS volumes as their root devices can be quickly stopped and started. When an instance is stopped, the compute resources are released and you are not billed for hourly

instance usage. However, your root partition Amazon EBS volume remains, continues to persist your data, and you are charged for Amazon EBS volume usage. You can restart your instance at any time. Each time you transition an instance from stopped to started, Amazon EC2 charges a full instance hour, even if transitions happen multiple times within a single hour.

Before stopping an instance, make sure it is in a state from which it can be restarted. Stopping an instance does not preserve data stored in RAM.

Performing this operation on an instance that uses an instance store as its root device returns an error.

For more information, see [Stopping Instances](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = subnet.instances.start(
    AdditionalInfo='string',
    DryRun=True|False
)
```

Parameters

- **AdditionalInfo** (*string*) – Reserved.
- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns

Response Syntax

```
{
  'StartingInstances': [
    {
      'InstanceId': 'string',
      'CurrentState': {
        'Code': 123,
        'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stopped'|'stopping',
      },
      'PreviousState': {
        'Code': 123,
        'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stopped'|'stopping',
      },
    },
  ],
}
```

Response Structure

- (*dict*) –
 - **StartingInstances** (*list*) –
Information about one or more started instances.
 - * (*dict*) –
Describes an instance state change.
 - **InstanceId** (*string*) –
The ID of the instance.

- **CurrentState** (*dict*) –
The current state of the instance.
- **Code** (*integer*) –
The low byte represents the state. The high byte is an opaque internal value and should be ignored.
 - 0: pending
 - 16: running
 - 32: shutting-down
 - 48: terminated
 - 64: stopping
 - 80: stopped
- **Name** (*string*) –
The current state of the instance.
- **PreviousState** (*dict*) –
The previous state of the instance.
- **Code** (*integer*) –
The low byte represents the state. The high byte is an opaque internal value and should be ignored.
 - 0: pending
 - 16: running
 - 32: shutting-down
 - 48: terminated
 - 64: stopping
 - 80: stopped
- **Name** (*string*) –
The current state of the instance.

stop (***kwargs*)

Stops an Amazon EBS-backed instance. Each time you transition an instance from stopped to started, Amazon EC2 charges a full instance hour, even if transitions happen multiple times within a single hour.

You can't start or stop Spot Instances.

Instances that use Amazon EBS volumes as their root devices can be quickly stopped and started. When an instance is stopped, the compute resources are released and you are not billed for hourly instance usage. However, your root partition Amazon EBS volume remains, continues to persist your data, and you are charged for Amazon EBS volume usage. You can restart your instance at any time.

Before stopping an instance, make sure it is in a state from which it can be restarted. Stopping an instance does not preserve data stored in RAM.

Performing this operation on an instance that uses an instance store as its root device returns an error.

You can stop, start, and terminate EBS-backed instances. You can only terminate instance store-backed instances. What happens to an instance differs if you stop it or terminate it. For example, when you stop an instance, the root device and any other devices attached to the instance persist. When you terminate an instance, the root device and any other devices attached during the instance launch are automatically deleted. For more information about the differences between stopping and terminating instances, see [Instance Lifecycle](#) in the *Amazon Elastic Compute Cloud User Guide*.

For more information about troubleshooting, see [Troubleshooting Stopping Your Instance](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = subnet.instances.stop(
    DryRun=True|False,
    Force=True|False
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Force** (*boolean*) – Forces the instances to stop. The instances do not have an opportunity to flush file system caches or file system metadata. If you use this option, you must perform file system check and repair procedures. This option is not recommended for Windows instances.

Default: false

Return type dict

Returns

Response Syntax

```
{
  'StoppingInstances': [
    {
      'InstanceId': 'string',
      'CurrentState': {
        'Code': 123,
        'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stopped'|'unknown',
      },
      'PreviousState': {
        'Code': 123,
        'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stopped'|'unknown',
      },
    },
  ],
}
```

Response Structure

- (*dict*) –
 - **StoppingInstances** (*list*) –
Information about one or more stopped instances.
 - * (*dict*) –

Describes an instance state change.

- **InstanceId** (*string*) –

The ID of the instance.

- **CurrentState** (*dict*) –

The current state of the instance.

- **Code** (*integer*) –

The low byte represents the state. The high byte is an opaque internal value and should be ignored.

- 0: pending
- 16: running
- 32: shutting-down
- 48: terminated
- 64: stopping
- 80: stopped
- **Name** (*string*) –
The current state of the instance.
- **PreviousState** (*dict*) –
The previous state of the instance.
- **Code** (*integer*) –
The low byte represents the state. The high byte is an opaque internal value and should be ignored.
- 0: pending
- 16: running
- 32: shutting-down
- 48: terminated
- 64: stopping
- 80: stopped
- **Name** (*string*) –
The current state of the instance.

terminate (***kwargs*)

Shuts down one or more instances. This operation is idempotent; if you terminate an instance more than once, each call succeeds.

Terminated instances remain visible after termination (for approximately one hour).

By default, Amazon EC2 deletes all EBS volumes that were attached when the instance launched. Volumes attached after instance launch continue running.

You can stop, start, and terminate EBS-backed instances. You can only terminate instance store-backed instances. What happens to an instance differs if you stop it or terminate it. For example, when you stop an instance, the root device and any other devices attached to the instance persist. When you terminate an instance, any attached EBS volumes with the `DeleteOnTermination` block device mapping parameter set to `true` are automatically deleted. For more information about the differences between stopping and terminating instances, see [Instance Lifecycle](#) in the *Amazon Elastic Compute Cloud User Guide*.

For more information about troubleshooting, see [Troubleshooting Terminating Your Instance](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = subnet.instances.terminate(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns

Response Syntax

```
{
  'TerminatingInstances': [
    {
      'InstanceId': 'string',
      'CurrentState': {
        'Code': 123,
        'Name': 'pending' | 'running' | 'shutting-down' | 'terminated' | 'stop
      },
      'PreviousState': {
        'Code': 123,
        'Name': 'pending' | 'running' | 'shutting-down' | 'terminated' | 'stop
    }
  ],
}
```

Response Structure

- (*dict*) –
 - **TerminatingInstances** (*list*) –

Information about one or more terminated instances.

- * (*dict*) –

Describes an instance state change.

- **InstanceId** (*string*) –
The ID of the instance.
- **CurrentState** (*dict*) –
The current state of the instance.
- **Code** (*integer*) –
The low byte represents the state. The high byte is an opaque internal value and should be ignored.
 - 0: pending
 - 16: running
 - 32: shutting-down
 - 48: terminated
 - 64: stopping
 - 80: stopped
- **Name** (*string*) –
The current state of the instance.
- **PreviousState** (*dict*) –
The previous state of the instance.
- **Code** (*integer*) –
The low byte represents the state. The high byte is an opaque internal value and should be ignored.
 - 0: pending
 - 16: running
 - 32: shutting-down
 - 48: terminated
 - 64: stopping
 - 80: stopped

- **Name** (*string*) –
The current state of the instance.

unmonitor (***kwargs*)

Disables monitoring for a running instance. For more information about monitoring instances, see [Monitoring Your Instances and Volumes](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = subnet.instances.unmonitor(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns

Response Syntax

```
{
  'InstanceMonitorings': [
    {
      'InstanceId': 'string',
      'Monitoring': {
        'State': 'disabled'|'disabling'|'enabled'|'pending'
      }
    },
  ],
}
```

Response Structure

- (*dict*) –
 - **InstanceMonitorings** (*list*) –
Monitoring information for one or more instances.
 - * (*dict*) –
Describes the monitoring information of the instance.
 - **InstanceId** (*string*) –
The ID of the instance.
 - **Monitoring** (*dict*) –
The monitoring information.
 - **State** (*string*) –
Indicates whether monitoring is enabled for the instance.

network_interfaces

all ()

Creates an iterable of all NetworkInterface resources in the collection.

Request Syntax

```
network_interface_iterator = subnet.network_interfaces.all()
```

Return type `list(ec2.NetworkInterface)`

Returns A list of NetworkInterface resources

filter (***kwargs*)

Creates an iterable of all NetworkInterface resources in the collection filtered by kwargs passed to method.

Request Syntax

```
network_interface_iterator = subnet.network_interfaces.filter(  
    DryRun=True|False,  
    NetworkInterfaceIds=[  
        'string',  
    ],  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **NetworkInterfaceIds** (*list*) – One or more network interface IDs.

Default: Describes all your network interfaces.

– (*string*) –

Return type `list(ec2.NetworkInterface)`

Returns A list of NetworkInterface resources

limit (***kwargs*)

Creates an iterable up to a specified amount of NetworkInterface resources in the collection.

Request Syntax

```
network_interface_iterator = subnet.network_interfaces.limit(  
    count=123  
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(ec2.NetworkInterface)`

Returns A list of NetworkInterface resources

page_size (***kwargs*)

Creates an iterable of all NetworkInterface resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
network_interface_iterator = subnet.network_interfaces.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(ec2.NetworkInterface)`

Returns A list of NetworkInterface resources

Tag

class `EC2.Tag(resource_id, key, value)`

A resource representing an Amazon Elastic Compute Cloud (EC2) Tag:

```
import boto3

ec2 = boto3.resource('ec2')
tag = ec2.Tag('resource_id', 'key', 'value')
```

Parameters

- **resource_id** (*string*) – The Tag’s resource_id identifier. This **must** be set.
- **key** (*string*) – The Tag’s key identifier. This **must** be set.
- **value** (*string*) – The Tag’s value identifier. This **must** be set.

These are the resource’s available identifiers:

- `resource_id`
- `key`
- `value`

These are the resource’s available attributes:

- `resource_type`

These are the resource’s available actions:

- `delete()`
- `load()`
- `reload()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

resource_id

(*string*) The Tag’s resource_id identifier. This **must** be set.

key

(*string*) The Tag’s key identifier. This **must** be set.

value

(*string*) The Tag’s value identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

resource_type

(*string*)

The resource type.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

delete (***kwargs*)

Deletes the specified set of tags from the specified set of resources. This call is designed to follow a `DescribeTags` request.

For more information about tags, see [Tagging Your Resources](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = tag.delete(  
    DryRun=True|False,  
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns `None`

load()

Calls `ec2.Client.describe_tags()` to update the attributes of the Tag resource

Request Syntax

```
tag.load()
```

Returns `None`

reload()

Calls `ec2.Client.describe_tags()` to update the attributes of the Tag resource

Request Syntax

```
tag.reload()
```

Returns `None`

Volume

class EC2.Volume (*id*)

A resource representing an Amazon Elastic Compute Cloud (EC2) Volume:

```
import boto3
```

```
ec2 = boto3.resource('ec2')  
volume = ec2.Volume('id')
```

Parameters **id** (*string*) – The Volume's id identifier. This **must** be set.

These are the resource's available identifiers:

- `id`

These are the resource's available attributes:

- `attachments`
- `availability_zone`
- `create_time`
- `encrypted`
- `iops`
- `kms_key_id`
- `size`
- `snapshot_id`
- `state`
- `tags`

- `volume_id`
- `volume_type`

These are the resource's available actions:

- `attach_to_instance()`
- `create_snapshot()`
- `create_tags()`
- `delete()`
- `describe_attribute()`
- `describe_status()`
- `detach_from_instance()`
- `enable_io()`
- `load()`
- `modify_attribute()`
- `reload()`

These are the resource's available collections:

- `snapshots`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

id

(*string*) The Volume's id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

attachments

(*list*)

Information about the volume attachments.

availability_zone

(*string*)

The Availability Zone for the volume.

create_time

(*datetime*)

The time stamp when volume creation was initiated.

encrypted

(*boolean*)

Indicates whether the volume will be encrypted.

iops

(*integer*)

The number of I/O operations per second (IOPS) that the volume supports. For Provisioned IOPS (SSD) volumes, this represents the number of IOPS that are provisioned for the volume. For General Purpose (SSD) volumes, this represents the baseline performance of the volume and the rate at which the volume accumulates I/O credits for bursting. For more information on General Purpose (SSD) baseline performance, I/O credits, and bursting, see [Amazon EBS Volume Types](#) in the *Amazon Elastic Compute Cloud User Guide*.

Constraint: Range is 100 to 20000 for Provisioned IOPS (SSD) volumes and 3 to 10000 for General Purpose (SSD) volumes.

Condition: This parameter is required for requests to create `io1` volumes; it is not used in requests to create `standard` or `gp2` volumes.

kms_key_id

(string)

The full ARN of the AWS Key Management Service (AWS KMS) customer master key (CMK) that was used to protect the volume encryption key for the volume.

size

(integer)

The size of the volume, in GiBs.

snapshot_id

(string)

The snapshot from which the volume was created, if applicable.

state

(string)

The volume state.

tags

(list)

Any tags assigned to the volume.

volume_id

(string)

The ID of the volume.

volume_type

(string)

The volume type. This can be `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, or `standard` for Magnetic volumes.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

attach_to_instance (**kwargs)

Attaches an EBS volume to a running or stopped instance and exposes it to the instance with the specified device name.

Encrypted EBS volumes may only be attached to instances that support Amazon EBS encryption. For more information, see [Amazon EBS Encryption](#) in the *Amazon Elastic Compute Cloud User Guide*.

For a list of supported device names, see [Attaching an EBS Volume to an Instance](#). Any device names that aren't reserved for instance store volumes can be used for EBS volumes. For more information, see [Amazon EC2 Instance Store](#) in the *Amazon Elastic Compute Cloud User Guide*.

If a volume has an AWS Marketplace product code:

- The volume can be attached only to a stopped instance.
- AWS Marketplace product codes are copied from the volume to the instance.
- You must be subscribed to the product.
- The instance type and operating system of the instance must support the product. For example, you can't detach a volume from a Windows instance and attach it to a Linux instance.

For an overview of the AWS Marketplace, see [Introducing AWS Marketplace](#) .

For more information about EBS volumes, see [Attaching Amazon EBS Volumes](#) in the *Amazon Elastic Compute Cloud User Guide* .

Request Syntax

```
response = volume.attach_to_instance(
    DryRun=True|False,
    InstanceId='string',
    Device='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation` . Otherwise, it is `UnauthorizedOperation` .
- **InstanceId** (*string*) – **[REQUIRED]**
The ID of the instance.
- **Device** (*string*) – **[REQUIRED]**
The device name to expose to the instance (for example, `/dev/sdh` or `xvdh`).

Return type dict

Returns

Response Syntax

```
{
    'VolumeId': 'string',
    'InstanceId': 'string',
    'Device': 'string',
    'State': 'attaching'|'attached'|'detaching'|'detached',
    'AttachTime': datetime(2015, 1, 1),
    'DeleteOnTermination': True|False
}
```

Response Structure

- (*dict*) –
Information about the volume attachment.
 - **VolumeId** (*string*) –
The ID of the volume.
 - **InstanceId** (*string*) –
The ID of the instance.
 - **Device** (*string*) –
The device name.
 - **State** (*string*) –
The attachment state of the volume.
 - **AttachTime** (*datetime*) –
The time stamp when the attachment initiated.
 - **DeleteOnTermination** (*boolean*) –
Indicates whether the EBS volume is deleted on instance termination.

create_snapshot (***kwargs*)

Creates a snapshot of an EBS volume and stores it in Amazon S3. You can use snapshots for backups, to make copies of EBS volumes, and to save data before shutting down an instance.

When a snapshot is created, any AWS Marketplace product codes that are associated with the source volume are propagated to the snapshot.

You can take a snapshot of an attached volume that is in use. However, snapshots only capture data that has been written to your EBS volume at the time the snapshot command is issued; this may exclude any data that has been cached by any applications or the operating system. If you can pause any file systems on the volume long enough to take a snapshot, your snapshot should be complete. However, if you cannot pause all file writes to the volume, you should unmount the volume from within the instance, issue the snapshot command, and then remount the volume to ensure a consistent and complete snapshot. You may remount and use your volume while the snapshot status is `pending`.

To create a snapshot for EBS volumes that serve as root devices, you should stop the instance before taking the snapshot.

Snapshots that are taken from encrypted volumes are automatically encrypted. Volumes that are created from encrypted snapshots are also automatically encrypted. Your encrypted volumes and any associated snapshots always remain protected.

For more information, see [Amazon Elastic Block Store](#) and [Amazon EBS Encryption](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
snapshot = volume.create_snapshot(  
    DryRun=True|False,  
    Description='string'  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Description** (*string*) – A description for the snapshot.

Return type `ec2.Snapshot`

Returns Snapshot resource

create_tags (***kwargs*)

Adds or overwrites one or more tags for the specified Amazon EC2 resource or resources. Each resource can have a maximum of 10 tags. Each tag consists of a key and optional value. Tag keys must be unique per resource.

For more information about tags, see [Tagging Your Resources](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
tag = volume.create_tags(  
    DryRun=True|False,  
    Tags=[  
        {  
            'Key': 'string',  
            'Value': 'string'  
        },  
    ],
```

```
]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Tags** (*list*) – **[REQUIRED]**

One or more tags. The `value` parameter is required, but if you don't want the tag to have a value, specify the parameter with no value, and we set the value to an empty string.

– (*dict*) –

Describes a tag.

* **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

* **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

Return type `list(ec2.Tag)`

Returns A list of Tag resources

delete (***kwargs*)

Deletes the specified EBS volume. The volume must be in the `available` state (not attached to an instance).

Note: The volume may remain in the `deleting` state for several minutes.

For more information, see [Deleting an Amazon EBS Volume](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = volume.delete(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns `None`

describe_attribute (***kwargs*)

Describes the specified attribute of the specified volume. You can specify only one attribute at a time.

For more information about EBS volumes, see [Amazon EBS Volumes](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = volume.describe_attribute(
    DryRun=True|False,
    Attribute='autoEnableIO'|'productCodes'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Attribute** (*string*) – The instance attribute.

Return type dict

Returns**Response Syntax**

```
{
  'VolumeId': 'string',
  'AutoEnableIO': {
    'Value': True|False
  },
  'ProductCodes': [
    {
      'ProductCodeId': 'string',
      'ProductCodeType': 'devpay'|'marketplace'
    },
    ...
  ]
}
```

Response Structure

- (*dict*) –
 - **VolumeId** (*string*) –
The ID of the volume.
 - **AutoEnableIO** (*dict*) –
The state of `autoEnableIO` attribute.
 - * **Value** (*boolean*) –
Valid values are `true` or `false`.
 - **ProductCodes** (*list*) –
A list of product codes.
 - * (*dict*) –
Describes a product code.
 - **ProductCodeId** (*string*) –
The product code.
 - **ProductCodeType** (*string*) –
The type of product code.

describe_status (***kwargs*)

Describes the status of the specified volumes. Volume status provides the result of the checks performed on your volumes to determine events that can impair the performance of your volumes. The performance of a volume can be affected if an issue occurs on the volume's underlying host. If the volume's underlying host experiences a power outage or system issue, after the system is restored, there could be data

inconsistencies on the volume. Volume events notify you if this occurs. Volume actions notify you if any action needs to be taken in response to the event.

The `DescribeVolumeStatus` operation provides the following information about the specified volumes:

Status : Reflects the current status of the volume. The possible values are `ok`, `impaired`, `warning`, or `insufficient-data`. If all checks pass, the overall status of the volume is `ok`. If the check fails, the overall status is `impaired`. If the status is `insufficient-data`, then the checks may still be taking place on your volume at the time. We recommend that you retry the request. For more information on volume status, see [Monitoring the Status of Your Volumes](#).

Events : Reflect the cause of a volume status and may require you to take action. For example, if your volume returns an `impaired` status, then the volume event might be `potential-data-inconsistency`. This means that your volume has been affected by an issue with the underlying host, has all I/O operations disabled, and may have inconsistent data.

Actions : Reflect the actions you may have to take in response to an event. For example, if the status of the volume is `impaired` and the volume event shows `potential-data-inconsistency`, then the action shows `enable-volume-io`. This means that you may want to enable the I/O operations for the volume by calling the `EnableVolumeIO` action and then check the volume for data consistency.

Note: Volume status is based on the volume status checks, and does not reflect the volume state. Therefore, volume status does not indicate volumes in the `error` state (for example, when a volume is incapable of accepting I/O.)

Request Syntax

```
response = volume.describe_status(
    DryRun=True|False,
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Filters** (*list*) – One or more filters.
 - `action.code` - The action code for the event (for example, `enable-volume-io`).
 - `action.description` - A description of the action.
 - `action.event-id` - The event ID associated with the action.
 - `availability-zone` - The Availability Zone of the instance.
 - `event.description` - A description of the event.
 - `event.event-id` - The event ID.
 - `event.event-type` - The event type (for `io-enabled`: `passed` | `failed`; for `io-performance`: `io-performance:degraded`

```
|          io-performance:severely-degraded          |
io-performance:stalled).
- event.not-after - The latest end time for the event.
- event.not-before - The earliest start time for the event.
- volume-status.details-name - The cause for
  volume-status.status (io-enabled | io-performance
  ).
- volume-status.details-status - The status of
  volume-status.details-name (for io-enabled : passed
  | failed ; for io-performance : normal | degraded |
  severely-degraded | stalled).
- volume-status.status - The status of the volume (ok | impaired
  | warning | insufficient-data ).
- (dict) -
```

A filter name and value pair that is used to return a more specific list of results. Filters can be used to match a set of resources by various criteria, such as tags, attributes, or IDs.

* **Name** (*string*) -

The name of the filter. Filter names are case-sensitive.

* **Values** (*list*) -

One or more filter values. Filter values are case-sensitive.

· (*string*) -

- **NextToken** (*string*) - The NextToken value to include in a future DescribeVolumeStatus request. When the results of the request exceed MaxResults , this value can be used to retrieve the next page of results. This value is null when there are no more results to return.
- **MaxResults** (*integer*) - The maximum number of volume results returned by DescribeVolumeStatus in paginated output. When this parameter is used, the request only returns MaxResults results in a single page along with a NextToken response element. The remaining results of the initial request can be seen by sending another request with the returned NextToken value. This value can be between 5 and 1000; if MaxResults is given a value larger than 1000, only 1000 results are returned. If this parameter is not used, then DescribeVolumeStatus returns all results. You cannot specify this parameter and the volume IDs parameter in the same request.

Return type dict

Returns

Response Syntax

```
{
  'VolumeStatuses': [
    {
      'VolumeId': 'string',
      'AvailabilityZone': 'string',
      'VolumeStatus': {
        'Status': 'ok'|'impaired'|'insufficient-data',
        'Details': [
          {
            'Name': 'io-enabled'|'io-performance',
            'Status': 'string'
          },
        ],
      },
    ],
  ],
}
```

```

    },
    'Events': [
        {
            'EventType': 'string',
            'Description': 'string',
            'NotBefore': datetime(2015, 1, 1),
            'NotAfter': datetime(2015, 1, 1),
            'EventId': 'string'
        },
    ],
    'Actions': [
        {
            'Code': 'string',
            'Description': 'string',
            'EventType': 'string',
            'EventId': 'string'
        },
    ],
]
},
'NextToken': 'string'
}

```

Response Structure

- *(dict)* –
 - **VolumeStatuses** (*list*) –
 - A list of volumes.
 - * *(dict)* –
 - Describes the volume status.
 - **VolumeId** (*string*) –
 - The volume ID.
 - **AvailabilityZone** (*string*) –
 - The Availability Zone of the volume.
 - **VolumeStatus** (*dict*) –
 - The volume status.
 - **Status** (*string*) –
 - The status of the volume.
 - **Details** (*list*) –
 - The details of the volume status.
 - *(dict)* –
 - Describes a volume status.
 - **Name** (*string*) –
 - The name of the volume status.
 - **Status** (*string*) –
 - The intended status of the volume status.
 - **Events** (*list*) –
 - A list of events associated with the volume.
 - *(dict)* –
 - Describes a volume status event.

- **EventType** (*string*) –
The type of this event.
- **Description** (*string*) –
A description of the event.
- **NotBefore** (*datetime*) –
The earliest start time of the event.
- **NotAfter** (*datetime*) –
The latest end time of the event.
- **EventId** (*string*) –
The ID of this event.
- **Actions** (*list*) –
The details of the operation.
- (*dict*) –
Describes a volume status operation code.
- **Code** (*string*) –
The code identifying the operation, for example, `enable-volume-io`.
- **Description** (*string*) –
A description of the operation.
- **EventType** (*string*) –
The event type associated with this operation.
- **EventId** (*string*) –
The ID of the event associated with this operation.
- **NextToken** (*string*) –
The token to use to retrieve the next page of results. This value is `null` when there are no more results to return.

detach_from_instance (***kwargs*)

Detaches an EBS volume from an instance. Make sure to unmount any file systems on the device within your operating system before detaching the volume. Failure to do so results in the volume being stuck in a busy state while detaching.

If an Amazon EBS volume is the root device of an instance, it can't be detached while the instance is running. To detach the root volume, stop the instance first.

When a volume with an AWS Marketplace product code is detached from an instance, the product code is no longer associated with the instance.

For more information, see [Detaching an Amazon EBS Volume](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = volume.detach_from_instance(  
    DryRun=True|False,  
    InstanceId='string',  
    Device='string',  
    Force=True|False  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceId** (*string*) – The ID of the instance.
- **Device** (*string*) – The device name.
- **Force** (*boolean*) – Forces detachment if the previous detachment attempt did not occur cleanly (for example, logging into an instance, unmounting the volume, and detaching normally). This option can lead to data loss or a corrupted file system. Use this option only as a last resort to detach a volume from a failed instance. The instance won't have an opportunity to flush file system caches or file system metadata. If you use this option, you must perform file system check and repair procedures.

Return type dict

Returns

Response Syntax

```
{
    'VolumeId': 'string',
    'InstanceId': 'string',
    'Device': 'string',
    'State': 'attaching'|'attached'|'detaching'|'detached',
    'AttachTime': datetime(2015, 1, 1),
    'DeleteOnTermination': True|False
}
```

Response Structure

- (*dict*) –

Information about the volume attachment.

- **VolumeId** (*string*) –
The ID of the volume.
- **InstanceId** (*string*) –
The ID of the instance.
- **Device** (*string*) –
The device name.
- **State** (*string*) –
The attachment state of the volume.
- **AttachTime** (*datetime*) –
The time stamp when the attachment initiated.
- **DeleteOnTermination** (*boolean*) –
Indicates whether the EBS volume is deleted on instance termination.

enable_io (***kwargs*)

Enables I/O operations for a volume that had I/O operations disabled because the data on the volume was potentially inconsistent.

Request Syntax

```
response = volume.enable_io(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns `None`

load()

Calls `ec2.Client.describe_volumes()` to update the attributes of the Volume resource

Request Syntax

```
volume.load()
```

Returns `None`

modify_attribute (***kwargs*)

Modifies a volume attribute.

By default, all I/O operations for the volume are suspended when the data on the volume is determined to be potentially inconsistent, to prevent undetectable, latent data corruption. The I/O access to the volume can be resumed by first enabling I/O access and then checking the data consistency on your volume.

You can change the default behavior to resume I/O operations. We recommend that you change this only for boot volumes or for volumes that are stateless or disposable.

Request Syntax

```
response = volume.modify_attribute(
    DryRun=True|False,
    AutoEnableIO={
        'Value': True|False
    }
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **AutoEnableIO** (*dict*) – Indicates whether the volume should be auto-enabled for I/O operations.
 - **Value** (*boolean*) –

Valid values are `true` or `false`.

Returns `None`

reload()

Calls `ec2.Client.describe_volumes()` to update the attributes of the Volume resource

Request Syntax

```
volume.reload()
```

Returns `None`

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

snapshots**all()**

Creates an iterable of all Snapshot resources in the collection.

Request Syntax

```
snapshot_iterator = volume.snapshots.all()
```

Return type list(ec2.Snapshot)

Returns A list of Snapshot resources

filter(kwargs)**

Creates an iterable of all Snapshot resources in the collection filtered by kwargs passed to method.

Request Syntax

```
snapshot_iterator = volume.snapshots.filter(
    DryRun=True|False,
    SnapshotIds=[
        'string',
    ],
    OwnerIds=[
        'string',
    ],
    RestorableByUserIds=[
        'string',
    ],
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SnapshotIds** (*list*) – One or more snapshot IDs.

Default: Describes snapshots for which you have launch permissions.

– (*string*) –

- **OwnerIds** (*list*) – Returns the snapshots owned by the specified owner. Multiple owners can be specified.
- **RestorableByUserIds** (*list*) – One or more AWS accounts IDs that can create volumes from the snapshot.

– (*string*) –

- **NextToken** (*string*) – The `NextToken` value returned from a previous paginated `DescribeSnapshots` request where `MaxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `NextToken` value. This value is `null` when there are no more results to return.
- **MaxResults** (*integer*) – The maximum number of snapshot results returned by `DescribeSnapshots` in paginated output. When this parameter is used, `DescribeSnapshots` only returns `MaxResults` results in a single page along with a `NextToken` response element. The remaining results of the initial request can be seen by sending another

`DescribeSnapshots` request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. If this parameter is not used, then `DescribeSnapshots` returns all results. You cannot specify this parameter and the snapshot IDs parameter in the same request.

Return type `list(ec2.Snapshot)`

Returns A list of Snapshot resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Snapshot resources in the collection.

Request Syntax

```
snapshot_iterator = volume.snapshots.limit(  
    count=123  
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(ec2.Snapshot)`

Returns A list of Snapshot resources

page_size (***kwargs*)

Creates an iterable of all Snapshot resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
snapshot_iterator = volume.snapshots.page_size(  
    count=123  
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(ec2.Snapshot)`

Returns A list of Snapshot resources

Vpc

class `EC2.Vpc` (*id*)

A resource representing an Amazon Elastic Compute Cloud (EC2) Vpc:

```
import boto3
```

```
ec2 = boto3.resource('ec2')  
vpc = ec2.Vpc('id')
```

Parameters `id` (*string*) – The Vpc's id identifier. This **must** be set.

These are the resource's available identifiers:

- `id`

These are the resource's available attributes:

- `cidr_block`
- `dhcp_options_id`
- `instance_tenancy`
- `is_default`
- `state`
- `tags`

- `vpc_id`

These are the resource's available references:

- `dhcp_options`

These are the resource's available actions:

- `associate_dhcp_options()`
- `attach_classic_link_instance()`
- `attach_internet_gateway()`
- `create_network_acl()`
- `create_route_table()`
- `create_security_group()`
- `create_subnet()`
- `create_tags()`
- `delete()`
- `describe_attribute()`
- `detach_classic_link_instance()`
- `detach_internet_gateway()`
- `disable_classic_link()`
- `enable_classic_link()`
- `load()`
- `modify_attribute()`
- `reload()`
- `request_vpc_peering_connection()`

These are the resource's available collections:

- `accepted_vpc_peering_connections`
- `instances`
- `internet_gateways`
- `network_acls`
- `network_interfaces`
- `requested_vpc_peering_connections`
- `route_tables`
- `security_groups`
- `subnets`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the *Resources Introduction Guide*.

id

(string) The Vpc's id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the *Resources Introduction Guide*.

cidr_block

(string)

The CIDR block for the VPC.

dhcp_options_id

(string)

The ID of the set of DHCP options you've associated with the VPC (or `default` if the default options are associated with the VPC).

instance_tenancy

(string)

The allowed tenancy of instances launched into the VPC.

is_default
(*boolean*)

Indicates whether the VPC is the default VPC.

state
(*string*)

The current state of the VPC.

tags
(*list*)

Any tags assigned to the VPC.

vpc_id
(*string*)

The ID of the VPC.

References

References are related resource instances that have a belongs-to relationship. For more information about references refer to the [Resources Introduction Guide](#).

dhcp_options
(DhcpOptions) The related dhcp_options if set, otherwise None.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

associate_dhcp_options (***kwargs*)
Associates a set of DHCP options (that you've previously created) with the specified VPC, or associates no DHCP options with the VPC.

After you associate the options with the VPC, any existing instances and all new instances that you launch in that VPC use the options. You don't need to restart or relaunch the instances. They automatically pick up the changes within a few hours, depending on how frequently the instance renews its DHCP lease. You can explicitly renew the lease using the operating system on the instance.

For more information, see [DHCP Options Sets](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
response = vpc.associate_dhcp_options(  
    DryRun=True|False,  
    DhcpOptionsId='string',  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **DhcpOptionsId** (*string*) – **[REQUIRED]**

The ID of the DHCP options set, or default to associate no DHCP options with the VPC.

Returns None**attach_classic_link_instance** (***kwargs*)

Links an EC2-Classic instance to a ClassicLink-enabled VPC through one or more of the VPC's security groups. You cannot link an EC2-Classic instance to more than one VPC at a time. You can only link an instance that's in the `running` state. An instance is automatically unlinked from a VPC when it's stopped - you can link it to the VPC again when you restart it.

After you've linked an instance, you cannot change the VPC security groups that are associated with it. To change the security groups, you must first unlink the instance, and then link it again.

Linking your instance to a VPC is sometimes referred to as *attaching* your instance.

Request Syntax

```
response = vpc.attach_classic_link_instance(
    DryRun=True|False,
    InstanceId='string',
    Groups=[
        'string',
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceId** (*string*) – **[REQUIRED]**
The ID of an EC2-Classic instance to link to the ClassicLink-enabled VPC.
- **Groups** (*list*) – **[REQUIRED]**
The ID of one or more of the VPC's security groups. You cannot specify security groups from a different VPC.
– (*string*) –

Return type dict

Returns**Response Syntax**

```
{
    'Return': True|False
}
```

Response Structure

- (*dict*) –
– **Return** (*boolean*) –
Returns `true` if the request succeeds; otherwise, it returns an error.

attach_internet_gateway (***kwargs*)

Attaches an Internet gateway to a VPC, enabling connectivity between the Internet and the VPC. For more information about your VPC and Internet gateway, see the [Amazon Virtual Private Cloud User Guide](#).

Request Syntax

```
response = vpc.attach_internet_gateway(
    DryRun=True|False,
    InternetGatewayId='string',
```

```
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InternetGatewayId** (*string*) – **[REQUIRED]**

The ID of the Internet gateway.

Returns None

create_network_acl (***kwargs*)

Creates a network ACL in a VPC. Network ACLs provide an optional layer of security (in addition to security groups) for the instances in your VPC.

For more information about network ACLs, see [Network ACLs](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
network_acl = vpc.create_network_acl(  
    DryRun=True|False,  
  
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type `ec2.NetworkAcl`

Returns NetworkAcl resource

create_route_table (***kwargs*)

Creates a route table for the specified VPC. After you create a route table, you can add routes and associate the table with a subnet.

For more information about route tables, see [Route Tables](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
route_table = vpc.create_route_table(  
    DryRun=True|False,  
  
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type `ec2.RouteTable`

Returns RouteTable resource

create_security_group (***kwargs*)

Creates a security group.

A security group is for use with instances either in the EC2-Classic platform or in a specific VPC. For more information, see [Amazon EC2 Security Groups](#) in the *Amazon Elastic Compute Cloud User Guide* and [Security Groups for Your VPC](#) in the *Amazon Virtual Private Cloud User Guide*.

Warning: EC2-Classic: You can have up to 500 security groups.
EC2-VPC: You can create up to 100 security groups per VPC.

When you create a security group, you specify a friendly name of your choice. You can have a security group for use in EC2-Classic with the same name as a security group for use in a VPC. However, you can't have two security groups for use in EC2-Classic with the same name or two security groups for use in a VPC with the same name.

You have a default security group for use in EC2-Classic and a default security group for use in your VPC. If you don't specify a security group when you launch an instance, the instance is launched into the appropriate default security group. A default security group includes a default rule that grants instances unrestricted network access to each other.

You can add or remove rules from your security groups using `AuthorizeSecurityGroupIngress`, `AuthorizeSecurityGroupEgress`, `RevokeSecurityGroupIngress`, and `RevokeSecurityGroupEgress`.

Request Syntax

```
security_group = vpc.create_security_group(
    DryRun=True|False,
    GroupName='string',
    Description='string',
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **GroupName** (*string*) – **[REQUIRED]**
The name of the security group.
Constraints: Up to 255 characters in length
Constraints for EC2-Classic: ASCII characters
Constraints for EC2-VPC: a-z, A-Z, 0-9, spaces, and `._-:/()#,@[]+=;{}!$*`
- **Description** (*string*) – **[REQUIRED]**
A description for the security group. This is informational only.
Constraints: Up to 255 characters in length
Constraints for EC2-Classic: ASCII characters
Constraints for EC2-VPC: a-z, A-Z, 0-9, spaces, and `._-:/()#,@[]+=;{}!$*`

Return type `ec2.SecurityGroup`

Returns SecurityGroup resource

create_subnet (***kwargs*)

Creates a subnet in an existing VPC.

When you create each subnet, you provide the VPC ID and the CIDR block you want for the subnet. After you create a subnet, you can't change its CIDR block. The subnet's CIDR block can be the same as the VPC's CIDR block (assuming you want only a single subnet in the VPC), or a subset of the VPC's CIDR

block. If you create more than one subnet in a VPC, the subnets' CIDR blocks must not overlap. The smallest subnet (and VPC) you can create uses a /28 netmask (16 IP addresses), and the largest uses a /16 netmask (65,536 IP addresses).

Warning: AWS reserves both the first four and the last IP address in each subnet's CIDR block. They're not available for use.

If you add more than one subnet to a VPC, they're set up in a star topology with a logical router in the middle.

If you launch an instance in a VPC using an Amazon EBS-backed AMI, the IP address doesn't change if you stop and restart the instance (unlike a similar instance launched outside a VPC, which gets a new IP address when restarted). It's therefore possible to have a subnet with no running instances (they're all stopped), but no remaining IP addresses available.

For more information about subnets, see [Your VPC and Subnets](#) in the *Amazon Virtual Private Cloud User Guide*.

Request Syntax

```
subnet = vpc.create_subnet(  
    DryRun=True|False,  
    CidrBlock='string',  
    AvailabilityZone='string'  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **CidrBlock** (*string*) – **[REQUIRED]**

The network range for the subnet, in CIDR notation. For example, `10.0.0.0/24`.

- **AvailabilityZone** (*string*) – The Availability Zone for the subnet.

Default: Amazon EC2 selects one for you (recommended).

Return type `ec2.Subnet`

Returns Subnet resource

`create_tags` (***kwargs*)

Adds or overwrites one or more tags for the specified Amazon EC2 resource or resources. Each resource can have a maximum of 10 tags. Each tag consists of a key and optional value. Tag keys must be unique per resource.

For more information about tags, see [Tagging Your Resources](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
tag = vpc.create_tags(  
    DryRun=True|False,  
    Tags=[  
        {  
            'Key': 'string',  
            'Value': 'string'  
        },  
    ],
```

```
]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Tags** (*list*) – **[REQUIRED]**

One or more tags. The `value` parameter is required, but if you don't want the tag to have a value, specify the parameter with no value, and we set the value to an empty string.

– (*dict*) –

Describes a tag.

* **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

* **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

Return type `list(ec2.Tag)`

Returns A list of Tag resources

delete (***kwargs*)

Deletes the specified VPC. You must detach or delete all gateways and resources that are associated with the VPC before you can delete it. For example, you must terminate all instances running in the VPC, delete all security groups associated with the VPC (except the default one), delete all route tables associated with the VPC (except the default one), and so on.

Request Syntax

```
response = vpc.delete(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns `None`

describe_attribute (***kwargs*)

Describes the specified attribute of the specified VPC. You can specify only one attribute at a time.

Request Syntax

```
response = vpc.describe_attribute(
    DryRun=True|False,
    Attribute='enableDnsSupport'|'enableDnsHostnames'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Attribute** (*string*) – The VPC attribute.

Return type dict**Returns****Response Syntax**

```
{
  'VpcId': 'string',
  'EnableDnsSupport': {
    'Value': True|False
  },
  'EnableDnsHostnames': {
    'Value': True|False
  }
}
```

Response Structure

- (*dict*) –
 - **VpcId** (*string*) –
The ID of the VPC.
 - **EnableDnsSupport** (*dict*) –
Indicates whether DNS resolution is enabled for the VPC. If this attribute is `true`, the Amazon DNS server resolves DNS hostnames for your instances to their corresponding IP addresses; otherwise, it does not.
 - * **Value** (*boolean*) –
Valid values are `true` or `false`.
 - **EnableDnsHostnames** (*dict*) –
Indicates whether the instances launched in the VPC get DNS hostnames. If this attribute is `true`, instances in the VPC get DNS hostnames; otherwise, they do not.
 - * **Value** (*boolean*) –
Valid values are `true` or `false`.

detach_classic_link_instance (***kwargs*)

Unlinks (detaches) a linked EC2-Classic instance from a VPC. After the instance has been unlinked, the VPC security groups are no longer associated with it. An instance is automatically unlinked from a VPC when it's stopped.

Request Syntax

```
response = vpc.detach_classic_link_instance(
    DryRun=True|False,
    InstanceId='string',
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If

you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **InstanceId** (*string*) – [REQUIRED]

The ID of the instance to unlink from the VPC.

Return type dict

Returns

Response Syntax

```
{
    'Return': True|False
}
```

Response Structure

- (*dict*) –
 - **Return** (*boolean*) –

Returns `true` if the request succeeds; otherwise, it returns an error.

detach_internet_gateway (***kwargs*)

Detaches an Internet gateway from a VPC, disabling connectivity between the Internet and the VPC. The VPC must not contain any running instances with Elastic IP addresses.

Request Syntax

```
response = vpc.detach_internet_gateway(
    DryRun=True|False,
    InternetGatewayId='string',
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InternetGatewayId** (*string*) – [REQUIRED]

The ID of the Internet gateway.

Returns None

disable_classic_link (***kwargs*)

Disables ClassicLink for a VPC. You cannot disable ClassicLink for a VPC that has EC2-Classic instances linked to it.

Request Syntax

```
response = vpc.disable_classic_link(
    DryRun=True|False,
)
```

- Parameters**
- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns**Response Syntax**

```
{
    'Return': True|False
}
```

Response Structure

- (*dict*) –
 - **Return** (*boolean*) –

Returns `true` if the request succeeds; otherwise, it returns an error.

enable_classic_link (***kwargs*)

Enables a VPC for ClassicLink. You can then link EC2-Classical instances to your ClassicLink-enabled VPC to allow communication over private IP addresses. You cannot enable your VPC for ClassicLink if any of your VPC's route tables have existing routes for address ranges within the 10.0.0.0/8 IP address range, excluding local routes for VPCs in the 10.0.0.0/16 and 10.1.0.0/16 IP address ranges. For more information, see [ClassicLink](#) in the Amazon Elastic Compute Cloud User Guide.

Request Syntax

```
response = vpc.enable_classic_link(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type `dict`

Returns**Response Syntax**

```
{
    'Return': True|False
}
```

Response Structure

- (*dict*) –
 - **Return** (*boolean*) –

Returns `true` if the request succeeds; otherwise, it returns an error.

load ()

Calls `ec2.Client.describe_vpcs()` to update the attributes of the Vpc resource

Request Syntax

```
vpc.load()
```

Returns `None`

modify_attribute (***kwargs*)

Modifies the specified attribute of the specified VPC.

Request Syntax

```
response = vpc.modify_attribute(
    EnableDnsSupport={
        'Value': True|False
    },
    EnableDnsHostnames={
        'Value': True|False
    }
)
```

Parameters

- **EnableDnsSupport** (*dict*) – Indicates whether the DNS resolution is supported for the VPC. If enabled, queries to the Amazon provided DNS server at the 169.254.169.253 IP address, or the reserved IP address at the base of the VPC network range “plus two” will succeed. If disabled, the Amazon provided DNS service in the VPC that resolves public DNS hostnames to IP addresses is not enabled.

– **Value** (*boolean*) –

Valid values are `true` or `false`.

- **EnableDnsHostnames** (*dict*) – Indicates whether the instances launched in the VPC get DNS hostnames. If enabled, instances in the VPC get DNS hostnames; otherwise, they do not.

You can only enable DNS hostnames if you also enable DNS support.

– **Value** (*boolean*) –

Valid values are `true` or `false`.

Returns None

reload()

Calls `ec2.Client.describe_vpcs()` to update the attributes of the Vpc resource

Request Syntax

```
vpc.reload()
```

Returns None

request_vpc_peering_connection (***kwargs*)

Requests a VPC peering connection between two VPCs: a requester VPC that you own and a peer VPC with which to create the connection. The peer VPC can belong to another AWS account. The requester VPC and peer VPC cannot have overlapping CIDR blocks.

The owner of the peer VPC must accept the peering request to activate the peering connection. The VPC peering connection request expires after 7 days, after which it cannot be accepted or rejected.

A `CreateVpcPeeringConnection` request between VPCs with overlapping CIDR blocks results in the VPC peering connection having a status of `failed`.

Request Syntax

```
vpc_peering_connection = vpc.request_vpc_peering_connection(
    DryRun=True|False,
    PeerVpcId='string',
    PeerOwnerId='string'
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **PeerVpcId** (*string*) – The ID of the VPC with which you are creating the VPC peering connection.
- **PeerOwnerId** (*string*) – The AWS account ID of the owner of the peer VPC.

Default: Your AWS account ID

Return type `ec2.VpcPeeringConnection`

Returns `VpcPeeringConnection` resource

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

accepted_vpc_peering_connections

all ()

Creates an iterable of all `VpcPeeringConnection` resources in the collection.

Request Syntax

```
vpc_peering_connection_iterator = vpc.accepted_vpc_peering_connections.all()
```

Return type `list(ec2.VpcPeeringConnection)`

Returns A list of `VpcPeeringConnection` resources

filter (***kwargs*)

Creates an iterable of all `VpcPeeringConnection` resources in the collection filtered by `kwargs` passed to method.

Request Syntax

```
vpc_peering_connection_iterator = vpc.accepted_vpc_peering_connections.filter(  
    DryRun=True|False,  
    VpcPeeringConnectionIds=[  
        'string',  
    ],  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcPeeringConnectionIds** (*list*) – One or more VPC peering connection IDs.

Default: Describes all your VPC peering connections.

– (*string*) –

Return type `list(ec2.VpcPeeringConnection)`

Returns A list of `VpcPeeringConnection` resources

limit (***kwargs*)

Creates an iterable up to a specified amount of `VpcPeeringConnection` resources in the collection.

Request Syntax

```
vpc_peering_connection_iterator = vpc.accepted_vpc_peering_connections.limit(
    count=123
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(ec2.VpcPeeringConnection)`

Returns A list of `VpcPeeringConnection` resources

page_size (***kwargs*)

Creates an iterable of all `VpcPeeringConnection` resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
vpc_peering_connection_iterator = vpc.accepted_vpc_peering_connections.page_size(
    count=123
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(ec2.VpcPeeringConnection)`

Returns A list of `VpcPeeringConnection` resources

instances

all ()

Creates an iterable of all `Instance` resources in the collection.

Request Syntax

```
instance_iterator = vpc.instances.all()
```

Return type `list(ec2.Instance)`

Returns A list of `Instance` resources

create_tags (***kwargs*)

Adds or overwrites one or more tags for the specified Amazon EC2 resource or resources. Each resource can have a maximum of 10 tags. Each tag consists of a key and optional value. Tag keys must be unique per resource.

For more information about tags, see [Tagging Your Resources](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = vpc.instances.create_tags(
    DryRun=True|False,
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **Tags** (*list*) – **[REQUIRED]**

One or more tags. The `value` parameter is required, but if you don't want the tag to have a value, specify the parameter with no value, and we set the value to an empty string.

– (*dict*) –

Describes a tag.

* **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

* **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

Returns None

filter (***kwargs*)

Creates an iterable of all Instance resources in the collection filtered by kwargs passed to method.

Request Syntax

```
instance_iterator = vpc.instances.filter(
    DryRun=True|False,
    InstanceIds=[
        'string',
    ],
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InstanceIds** (*list*) – One or more instance IDs.

Default: Describes all your instances.

– (*string*) –

- **NextToken** (*string*) – The token to request the next page of results.
- **MaxResults** (*integer*) – The maximum number of results to return for the request in a single page. The remaining results of the initial request can be seen by sending another request with the returned `NextToken` value. This value can be between 5 and 1000; if `MaxResults` is given a value larger than 1000, only 1000 results are returned. You cannot specify this parameter and the instance IDs parameter in the same request.

Return type `list(ec2.Instance)`

Returns A list of Instance resources

limit (**kwargs)

Creates an iterable up to a specified amount of Instance resources in the collection.

Request Syntax

```
instance_iterator = vpc.instances.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type list(ec2.Instance)

Returns A list of Instance resources

monitor (**kwargs)

Enables monitoring for a running instance. For more information about monitoring instances, see [Monitoring Your Instances and Volumes](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = vpc.instances.monitor(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns

Response Syntax

```
{
  'InstanceMonitorings': [
    {
      'InstanceId': 'string',
      'Monitoring': {
        'State': 'disabled'|'disabling'|'enabled'|'pending'
      }
    },
  ],
}
```

Response Structure

- (*dict*) –
 - **InstanceMonitorings** (*list*) –
 - Monitoring information for one or more instances.
 - * (*dict*) –
 - Describes the monitoring information of the instance.
 - **InstanceId** (*string*) –
 - The ID of the instance.
 - **Monitoring** (*dict*) –
 - The monitoring information.

· **State** (*string*) –

Indicates whether monitoring is enabled for the instance.

page_size (***kwargs*)

Creates an iterable of all Instance resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
instance_iterator = vpc.instances.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type list(ec2.Instance)

Returns A list of Instance resources

reboot (***kwargs*)

Requests a reboot of one or more instances. This operation is asynchronous; it only queues a request to reboot the specified instances. The operation succeeds if the instances are valid and belong to you. Requests to reboot terminated instances are ignored.

If a Linux/Unix instance does not cleanly shut down within four minutes, Amazon EC2 performs a hard reboot.

For more information about troubleshooting, see [Getting Console Output and Rebooting Instances](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = vpc.instances.reboot(  
    DryRun=True|False,  
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Returns None

start (***kwargs*)

Starts an Amazon EBS-backed AMI that you've previously stopped.

Instances that use Amazon EBS volumes as their root devices can be quickly stopped and started. When an instance is stopped, the compute resources are released and you are not billed for hourly instance usage. However, your root partition Amazon EBS volume remains, continues to persist your data, and you are charged for Amazon EBS volume usage. You can restart your instance at any time. Each time you transition an instance from stopped to started, Amazon EC2 charges a full instance hour, even if transitions happen multiple times within a single hour.

Before stopping an instance, make sure it is in a state from which it can be restarted. Stopping an instance does not preserve data stored in RAM.

Performing this operation on an instance that uses an instance store as its root device returns an error.

For more information, see [Stopping Instances](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = vpc.instances.start(
    AdditionalInfo='string',
    DryRun=True|False
)
```

Parameters

- **AdditionalInfo** (*string*) – Reserved.
- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns

Response Syntax

```
{
    'StartingInstances': [
        {
            'InstanceId': 'string',
            'CurrentState': {
                'Code': 123,
                'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stop
            },
            'PreviousState': {
                'Code': 123,
                'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stop
            }
        },
    ],
}
```

Response Structure

- (*dict*) –
 - **StartingInstances** (*list*) –

Information about one or more started instances.

* (*dict*) –

Describes an instance state change.

- **InstanceId** (*string*) –

The ID of the instance.

- **CurrentState** (*dict*) –

The current state of the instance.

- **Code** (*integer*) –

The low byte represents the state. The high byte is an opaque internal value and should be ignored.

- 0: pending
- 16: running
- 32: shutting-down
- 48: terminated
- 64: stopping
- 80: stopped

- **Name** (*string*) –
The current state of the instance.
- **PreviousState** (*dict*) –
The previous state of the instance.
- **Code** (*integer*) –
The low byte represents the state. The high byte is an opaque internal value and should be ignored.
 - 0: pending
 - 16: running
 - 32: shutting-down
 - 48: terminated
 - 64: stopping
 - 80: stopped
- **Name** (*string*) –
The current state of the instance.

stop (***kwargs*)

Stops an Amazon EBS-backed instance. Each time you transition an instance from stopped to started, Amazon EC2 charges a full instance hour, even if transitions happen multiple times within a single hour.

You can't start or stop Spot Instances.

Instances that use Amazon EBS volumes as their root devices can be quickly stopped and started. When an instance is stopped, the compute resources are released and you are not billed for hourly instance usage. However, your root partition Amazon EBS volume remains, continues to persist your data, and you are charged for Amazon EBS volume usage. You can restart your instance at any time.

Before stopping an instance, make sure it is in a state from which it can be restarted. Stopping an instance does not preserve data stored in RAM.

Performing this operation on an instance that uses an instance store as its root device returns an error.

You can stop, start, and terminate EBS-backed instances. You can only terminate instance store-backed instances. What happens to an instance differs if you stop it or terminate it. For example, when you stop an instance, the root device and any other devices attached to the instance persist. When you terminate an instance, the root device and any other devices attached during the instance launch are automatically deleted. For more information about the differences between stopping and terminating instances, see [Instance Lifecycle](#) in the *Amazon Elastic Compute Cloud User Guide*.

For more information about troubleshooting, see [Troubleshooting Stopping Your Instance](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = vpc.instances.stop(  
    DryRun=True|False,  
    Force=True|False  
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an er-

ror response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

- **Force** (*boolean*) – Forces the instances to stop. The instances do not have an opportunity to flush file system caches or file system metadata. If you use this option, you must perform file system check and repair procedures. This option is not recommended for Windows instances.

Default: `false`

Return type `dict`

Returns

Response Syntax

```
{
  'StoppingInstances': [
    {
      'InstanceId': 'string',
      'CurrentState': {
        'Code': 123,
        'Name': 'pending' | 'running' | 'shutting-down' | 'terminated' | 'stop
      },
      'PreviousState': {
        'Code': 123,
        'Name': 'pending' | 'running' | 'shutting-down' | 'terminated' | 'stop
      }
    },
  ],
}
```

Response Structure

- (*dict*) –
 - **StoppingInstances** (*list*) –

Information about one or more stopped instances.

* (*dict*) –

Describes an instance state change.

- **InstanceId** (*string*) –

The ID of the instance.

- **CurrentState** (*dict*) –

The current state of the instance.

- **Code** (*integer*) –

The low byte represents the state. The high byte is an opaque internal value and should be ignored.

- 0: pending
- 16: running
- 32: shutting-down
- 48: terminated
- 64: stopping
- 80: stopped
- **Name** (*string*) –

The current state of the instance.

- **PreviousState** (*dict*) –

The previous state of the instance.

- **Code** (*integer*) –

The low byte represents the state. The high byte is an opaque internal value and should be ignored.

- 0: pending
- 16: running
- 32: shutting-down
- 48: terminated
- 64: stopping
- 80: stopped
- **Name** (*string*) –

The current state of the instance.

terminate (**kwargs)

Shuts down one or more instances. This operation is idempotent; if you terminate an instance more than once, each call succeeds.

Terminated instances remain visible after termination (for approximately one hour).

By default, Amazon EC2 deletes all EBS volumes that were attached when the instance launched. Volumes attached after instance launch continue running.

You can stop, start, and terminate EBS-backed instances. You can only terminate instance store-backed instances. What happens to an instance differs if you stop it or terminate it. For example, when you stop an instance, the root device and any other devices attached to the instance persist. When you terminate an instance, any attached EBS volumes with the `DeleteOnTermination` block device mapping parameter set to `true` are automatically deleted. For more information about the differences between stopping and terminating instances, see [Instance Lifecycle](#) in the *Amazon Elastic Compute Cloud User Guide*.

For more information about troubleshooting, see [Troubleshooting Terminating Your Instance](#) in the *Amazon Elastic Compute Cloud User Guide*.

Request Syntax

```
response = vpc.instances.terminate(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns

Response Syntax

```
{
    'TerminatingInstances': [
        {
            'InstanceId': 'string',
            'CurrentState': {
                'Code': 123,
                'Name': 'pending'|'running'|'shutting-down'|'terminated'|'stop
            },
            'PreviousState': {
```



```

        'Code': 123,
        'Name': 'pending' | 'running' | 'shutting-down' | 'terminated' | 'stop
    },
    ],
}

```

Response Structure

- *(dict)* –
 - **TerminatingInstances** (*list*) –

Information about one or more terminated instances.

* *(dict)* –

Describes an instance state change.

- **InstanceId** (*string*) –
The ID of the instance.
- **CurrentState** (*dict*) –
The current state of the instance.
- **Code** (*integer*) –
The low byte represents the state. The high byte is an opaque internal value and should be ignored.
 - 0: pending
 - 16: running
 - 32: shutting-down
 - 48: terminated
 - 64: stopping
 - 80: stopped
- **Name** (*string*) –
The current state of the instance.
- **PreviousState** (*dict*) –
The previous state of the instance.
- **Code** (*integer*) –
The low byte represents the state. The high byte is an opaque internal value and should be ignored.
 - 0: pending
 - 16: running
 - 32: shutting-down
 - 48: terminated
 - 64: stopping
 - 80: stopped
- **Name** (*string*) –
The current state of the instance.

unmonitor (***kwargs*)

Disables monitoring for a running instance. For more information about monitoring instances, see [Monitoring Your Instances and Volumes in the Amazon Elastic Compute Cloud User Guide](#).

Request Syntax

```

response = vpc.instances.unmonitor(
    DryRun=True|False,

```

)

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns

Response Syntax

```
{
  'InstanceMonitorings': [
    {
      'InstanceId': 'string',
      'Monitoring': {
        'State': 'disabled'|'disabling'|'enabled'|'pending'
      }
    },
  ]
}
```

Response Structure

- (*dict*) –

- **InstanceMonitorings** (*list*) –

Monitoring information for one or more instances.

- * (*dict*) –

- Describes the monitoring information of the instance.

- **InstanceId** (*string*) –

- The ID of the instance.

- **Monitoring** (*dict*) –

- The monitoring information.

- **State** (*string*) –

- Indicates whether monitoring is enabled for the instance.

internet_gateways

all()

Creates an iterable of all InternetGateway resources in the collection.

Request Syntax

```
internet_gateway_iterator = vpc.internet_gateways.all()
```

Return type list(ec2.InternetGateway)

Returns A list of InternetGateway resources

filter (***kwargs*)

Creates an iterable of all InternetGateway resources in the collection filtered by kwargs passed to method.

Request Syntax

```
internet_gateway_iterator = vpc.internet_gateways.filter(
    DryRun=True|False,
    InternetGatewayIds=[
        'string',
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **InternetGatewayIds** (*list*) – One or more Internet gateway IDs.

Default: Describes all your Internet gateways.

– (*string*) –

Return type `list(ec2.InternetGateway)`

Returns A list of InternetGateway resources

limit (***kwargs*)

Creates an iterable up to a specified amount of InternetGateway resources in the collection.

Request Syntax

```
internet_gateway_iterator = vpc.internet_gateways.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(ec2.InternetGateway)`

Returns A list of InternetGateway resources

page_size (***kwargs*)

Creates an iterable of all InternetGateway resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
internet_gateway_iterator = vpc.internet_gateways.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(ec2.InternetGateway)`

Returns A list of InternetGateway resources

network_acls

all ()

Creates an iterable of all NetworkAcl resources in the collection.

Request Syntax

```
network_acl_iterator = vpc.network_acls.all()
```

Return type `list(ec2.NetworkAcl)`

Returns A list of NetworkAcl resources

filter (***kwargs*)

Creates an iterable of all NetworkAcl resources in the collection filtered by kwargs passed to method.

Request Syntax

```
network_acl_iterator = vpc.network_acls.filter(
    DryRun=True|False,
    NetworkAclIds=[
        'string',
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **NetworkAclIds** (*list*) – One or more network ACL IDs.

Default: Describes all your network ACLs.

– (*string*) –

Return type `list(ec2.NetworkAcl)`

Returns A list of NetworkAcl resources

limit (***kwargs*)

Creates an iterable up to a specified amount of NetworkAcl resources in the collection.

Request Syntax

```
network_acl_iterator = vpc.network_acls.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(ec2.NetworkAcl)`

Returns A list of NetworkAcl resources

page_size (***kwargs*)

Creates an iterable of all NetworkAcl resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
network_acl_iterator = vpc.network_acls.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(ec2.NetworkAcl)`

Returns A list of NetworkAcl resources

network_interfaces

all()

Creates an iterable of all NetworkInterface resources in the collection.

Request Syntax

```
network_interface_iterator = vpc.network_interfaces.all()
```

Return type list(ec2.NetworkInterface)

Returns A list of NetworkInterface resources

filter(kwargs)**

Creates an iterable of all NetworkInterface resources in the collection filtered by kwargs passed to method.

Request Syntax

```
network_interface_iterator = vpc.network_interfaces.filter(
    DryRun=True|False,
    NetworkInterfaceIds=[
        'string',
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **NetworkInterfaceIds** (*list*) – One or more network interface IDs.

Default: Describes all your network interfaces.

– (*string*) –

Return type list(ec2.NetworkInterface)

Returns A list of NetworkInterface resources

limit(kwargs)**

Creates an iterable up to a specified amount of NetworkInterface resources in the collection.

Request Syntax

```
network_interface_iterator = vpc.network_interfaces.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type list(ec2.NetworkInterface)

Returns A list of NetworkInterface resources

page_size(kwargs)**

Creates an iterable of all NetworkInterface resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
network_interface_iterator = vpc.network_interfaces.page_size(
    count=123
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(ec2.NetworkInterface)`

Returns A list of `NetworkInterface` resources

`requested_vpc_peering_connections`

all ()

Creates an iterable of all `VpcPeeringConnection` resources in the collection.

Request Syntax

```
vpc_peering_connection_iterator = vpc.requested_vpc_peering_connections.all()
```

Return type `list(ec2.VpcPeeringConnection)`

Returns A list of `VpcPeeringConnection` resources

filter (***kwargs*)

Creates an iterable of all `VpcPeeringConnection` resources in the collection filtered by `kwargs` passed to method.

Request Syntax

```
vpc_peering_connection_iterator = vpc.requested_vpc_peering_connections.filter(
    DryRun=True|False,
    VpcPeeringConnectionIds=[
        'string',
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **VpcPeeringConnectionIds** (*list*) – One or more VPC peering connection IDs.

Default: Describes all your VPC peering connections.

– (*string*) –

Return type `list(ec2.VpcPeeringConnection)`

Returns A list of `VpcPeeringConnection` resources

limit (***kwargs*)

Creates an iterable up to a specified amount of `VpcPeeringConnection` resources in the collection.

Request Syntax

```
vpc_peering_connection_iterator = vpc.requested_vpc_peering_connections.limit(
    count=123
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(ec2.VpcPeeringConnection)`

Returns A list of `VpcPeeringConnection` resources

page_size (***kwargs*)

Creates an iterable of all `VpcPeeringConnection` resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
vpc_peering_connection_iterator = vpc.requested_vpc_peering_connections.page_size(
    count=123
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(ec2.VpcPeeringConnection)`

Returns A list of `VpcPeeringConnection` resources

route_tables

all ()

Creates an iterable of all `RouteTable` resources in the collection.

Request Syntax

```
route_table_iterator = vpc.route_tables.all()
```

Return type `list(ec2.RouteTable)`

Returns A list of `RouteTable` resources

filter (***kwargs*)

Creates an iterable of all `RouteTable` resources in the collection filtered by `kwargs` passed to method.

Request Syntax

```
route_table_iterator = vpc.route_tables.filter(
    DryRun=True|False,
    RouteTableIds=[
        'string',
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **RouteTableIds** (*list*) – One or more route table IDs.

Default: Describes all your route tables.

– (*string*) –

Return type `list(ec2.RouteTable)`

Returns A list of `RouteTable` resources

limit (***kwargs*)

Creates an iterable up to a specified amount of `RouteTable` resources in the collection.

Request Syntax

```
route_table_iterator = vpc.route_tables.limit(
    count=123
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(ec2.RouteTable)`

Returns A list of RouteTable resources

page_size (***kwargs*)

Creates an iterable of all RouteTable resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
route_table_iterator = vpc.route_tables.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type list(ec2.RouteTable)

Returns A list of RouteTable resources

security_groups

all ()

Creates an iterable of all SecurityGroup resources in the collection.

Request Syntax

```
security_group_iterator = vpc.security_groups.all()
```

Return type list(ec2.SecurityGroup)

Returns A list of SecurityGroup resources

filter (***kwargs*)

Creates an iterable of all SecurityGroup resources in the collection filtered by kwargs passed to method.

Request Syntax

```
security_group_iterator = vpc.security_groups.filter(
    DryRun=True|False,
    GroupNames=[
        'string',
    ],
    GroupIds=[
        'string',
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **GroupNames** (*list*) – [EC2-Classic and default VPC only] One or more security group names. You can specify either the security group name or the security group ID. For security groups in a nondefault VPC, use the `group-name` filter to describe security groups by name.

Default: Describes all your security groups.

– (*string*) –

- **GroupIds** (*list*) – One or more security group IDs. Required for security groups in a nondefault VPC.

Default: Describes all your security groups.

– (*string*) –

Return type `list(ec2.SecurityGroup)`

Returns A list of SecurityGroup resources

limit (***kwargs*)

Creates an iterable up to a specified amount of SecurityGroup resources in the collection.

Request Syntax

```
security_group_iterator = vpc.security_groups.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(ec2.SecurityGroup)`

Returns A list of SecurityGroup resources

page_size (***kwargs*)

Creates an iterable of all SecurityGroup resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
security_group_iterator = vpc.security_groups.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(ec2.SecurityGroup)`

Returns A list of SecurityGroup resources

subnets

all ()

Creates an iterable of all Subnet resources in the collection.

Request Syntax

```
subnet_iterator = vpc.subnets.all()
```

Return type `list(ec2.Subnet)`

Returns A list of Subnet resources

filter (***kwargs*)

Creates an iterable of all Subnet resources in the collection filtered by kwargs passed to method.

Request Syntax

```
subnet_iterator = vpc.subnets.filter(
    DryRun=True|False,
    SubnetIds=[
        'string',
    ],
)
```

Parameters

- **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.
- **SubnetIds** (*list*) – One or more subnet IDs.

Default: Describes all your subnets.

– (*string*) –

Return type `list(ec2.Subnet)`

Returns A list of Subnet resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Subnet resources in the collection.

Request Syntax

```
subnet_iterator = vpc.subnets.limit(  
    count=123  
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(ec2.Subnet)`

Returns A list of Subnet resources

page_size (***kwargs*)

Creates an iterable of all Subnet resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
subnet_iterator = vpc.subnets.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(ec2.Subnet)`

Returns A list of Subnet resources

VpcPeeringConnection

class `EC2.VpcPeeringConnection` (*id*)

A resource representing an Amazon Elastic Compute Cloud (EC2) VpcPeeringConnection:

```
import boto3
```

```
ec2 = boto3.resource('ec2')  
vpc_peering_connection = ec2.VpcPeeringConnection('id')
```

Parameters **id** (*string*) – The VpcPeeringConnection’s id identifier. This **must** be set.

These are the resource’s available identifiers:

- `id`

These are the resource’s available attributes:

- `accepter_vpc_info`
- `expiration_time`
- `requester_vpc_info`

- `status`
- `tags`
- `vpc_peering_connection_id`

These are the resource's available references:

- `accepter_vpc`
- `requester_vpc`

These are the resource's available actions:

- `accept()`
- `delete()`
- `load()`
- `reject()`
- `reload()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

id

(*string*) The VpcPeeringConnection's id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

accepter_vpc_info

(*dict*)

The information of the peer VPC.

expiration_time

(*datetime*)

The time that an unaccepted VPC peering connection will expire.

requester_vpc_info

(*dict*)

The information of the requester VPC.

status

(*dict*)

The status of the VPC peering connection.

tags

(*list*)

Any tags assigned to the resource.

vpc_peering_connection_id

(*string*)

The ID of the VPC peering connection.

References

References are related resource instances that have a belongs-to relationship. For more information about references refer to the [Resources Introduction Guide](#).

accepter_vpc

(Vpc) The related accepter_vpc if set, otherwise None.

requester_vpc

(Vpc) The related requester_vpc if set, otherwise None.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

accept (***kwargs*)

Accept a VPC peering connection request. To accept a request, the VPC peering connection must be in the `pending-acceptance` state, and you must be the owner of the peer VPC. Use the `DescribeVpcPeeringConnections` request to view your outstanding VPC peering connection requests.

Request Syntax

```
response = vpc_peering_connection.accept(  
    DryRun=True|False,  
  
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns**Response Syntax**

```
{  
    'VpcPeeringConnection': {  
        'AcceptorVpcInfo': {  
            'CidrBlock': 'string',  
            'OwnerId': 'string',  
            'VpcId': 'string'  
        },  
        'ExpirationTime': datetime(2015, 1, 1),  
        'RequesterVpcInfo': {  
            'CidrBlock': 'string',  
            'OwnerId': 'string',  
            'VpcId': 'string'  
        },  
        'Status': {  
            'Code': 'initiating-request'|'pending-acceptance'|'active'|'deleted'|'  
            'Message': 'string'  
        },  
        'Tags': [  
            {  
                'Key': 'string',  
                'Value': 'string'  
            },  
        ],  
        'VpcPeeringConnectionId': 'string'  
    }  
}
```

Response Structure

- (*dict*) –

– **VpcPeeringConnection** (*dict*) –

Information about the VPC peering connection.

* **AccepterVpcInfo** (*dict*) –

The information of the peer VPC.

· **CidrBlock** (*string*) –

The CIDR block for the VPC.

· **OwnerId** (*string*) –

The AWS account ID of the VPC owner.

· **VpcId** (*string*) –

The ID of the VPC.

* **ExpirationTime** (*datetime*) –

The time that an unaccepted VPC peering connection will expire.

* **RequesterVpcInfo** (*dict*) –

The information of the requester VPC.

· **CidrBlock** (*string*) –

The CIDR block for the VPC.

· **OwnerId** (*string*) –

The AWS account ID of the VPC owner.

· **VpcId** (*string*) –

The ID of the VPC.

* **Status** (*dict*) –

The status of the VPC peering connection.

· **Code** (*string*) –

The status of the VPC peering connection.

· **Message** (*string*) –

A message that provides more information about the status, if applicable.

* **Tags** (*list*) –

Any tags assigned to the resource.

· (*dict*) –

Describes a tag.

· **Key** (*string*) –

The key of the tag.

Constraints: Tag keys are case-sensitive and accept a maximum of 127 Unicode characters. May not begin with `aws:`

· **Value** (*string*) –

The value of the tag.

Constraints: Tag values are case-sensitive and accept a maximum of 255 Unicode characters.

* **VpcPeeringConnectionId** (*string*) –

The ID of the VPC peering connection.

delete (***kwargs*)

Deletes a VPC peering connection. Either the owner of the requester VPC or the owner of the peer VPC

can delete the VPC peering connection if it's in the `active` state. The owner of the requester VPC can delete a VPC peering connection in the `pending-acceptance` state.

Request Syntax

```
response = vpc_peering_connection.delete(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns

Response Syntax

```
{
    'Return': True|False
}
```

Response Structure

- (*dict*) –
 - **Return** (*boolean*) –
Returns `true` if the request succeeds; otherwise, it returns an error.

load()

Calls `ec2.Client.describe_vpc_peering_connections()` to update the attributes of the `VpcPeeringConnection` resource

Request Syntax

```
vpc_peering_connection.load()
```

Returns None

reject (***kwargs*)

Rejects a VPC peering connection request. The VPC peering connection must be in the `pending-acceptance` state. Use the `DescribeVpcPeeringConnections` request to view your outstanding VPC peering connection requests. To delete an active VPC peering connection, or to delete a VPC peering connection request that you initiated, use `DeleteVpcPeeringConnection`.

Request Syntax

```
response = vpc_peering_connection.reject(
    DryRun=True|False,
)
```

Parameters **DryRun** (*boolean*) – Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

Return type dict

Returns**Response Syntax**

```
{
    'Return': True|False
}
```

Response Structure

- (dict) –
 - **Return** (boolean) –

Returns `true` if the request succeeds; otherwise, it returns an error.

reload()

Calls `ec2.Client.describe_vpc_peering_connections()` to update the attributes of the `VpcPeeringConnection` resource

Request Syntax

```
vpc_peering_connection.reload()
```

Returns None**ECS****Table of Contents**

- [ECS](#)
 - [Client](#)
 - [Paginators](#)
 - [Waiters](#)

Client**class ECS.Client**

A low-level client representing Amazon EC2 Container Service (ECS):

```
import boto3
```

```
client = boto3.client('ecs')
```

These are the available methods:

- `can_paginate()`
- `create_cluster()`
- `create_service()`
- `delete_cluster()`
- `delete_service()`
- `deregister_container_instance()`
- `deregister_task_definition()`
- `describe_clusters()`
- `describe_container_instances()`
- `describe_services()`
- `describe_task_definition()`

- `describe_tasks()`
- `discover_poll_endpoint()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_waiter()`
- `list_clusters()`
- `list_container_instances()`
- `list_services()`
- `list_task_definition_families()`
- `list_task_definitions()`
- `list_tasks()`
- `register_container_instance()`
- `register_task_definition()`
- `run_task()`
- `start_task()`
- `stop_task()`
- `submit_container_state_change()`
- `submit_task_state_change()`
- `update_container_agent()`
- `update_service()`

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

create_cluster (***kwargs*)

Creates a new Amazon ECS cluster. By default, your account will receive a default cluster when you launch your first container instance. However, you can create your own cluster with a unique name with the `CreateCluster` action.

Request Syntax

```
response = client.create_cluster(  
    clusterName='string'  
)
```

Parameters *clusterName* (*string*) – The name of your cluster. If you do not specify a name for your cluster, you will create a cluster named `default`. Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.

Return type dict

Returns

Response Syntax

```
{  
    'cluster': {  
        'clusterArn': 'string',  
        'clusterName': 'string',  
        'status': 'string',  
        'registeredContainerInstancesCount': 123,  
        'runningTasksCount': 123,  
        'pendingTasksCount': 123,
```



```

        'activeServicesCount': 123
    }
}

```

Response Structure

- (*dict*) –
 - **cluster** (*dict*) –

The full description of your new cluster.

- * **clusterArn** (*string*) –

The Amazon Resource Name (ARN) that identifies the cluster. The ARN contains the `arn:aws:ecs` namespace, followed by the region of the cluster, the AWS account ID of the cluster owner, the `cluster` namespace, and then the cluster name. For example, `arn:aws:ecs:region:012345678910:cluster/test`.

- * **clusterName** (*string*) –

A user-generated string that you can use to identify your cluster.

- * **status** (*string*) –

The status of the cluster. The valid values are `ACTIVE` or `INACTIVE`. `ACTIVE` indicates that you can register container instances with the cluster and the associated instances can accept tasks.

- * **registeredContainerInstancesCount** (*integer*) –

The number of container instances registered into the cluster.

- * **runningTasksCount** (*integer*) –

The number of tasks in the cluster that are in the `RUNNING` state.

- * **pendingTasksCount** (*integer*) –

The number of tasks in the cluster that are in the `PENDING` state.

- * **activeServicesCount** (*integer*) –

The number of services that are running on the cluster in an `ACTIVE` state. You can view these services with `ListServices`.

create_service (***kwargs*)

Runs and maintains a desired number of tasks from a specified task definition. If the number of tasks running in a service drops below `desiredCount`, Amazon ECS will spawn another instantiation of the task in the specified cluster.

Request Syntax

```

response = client.create_service(
    cluster='string',
    serviceName='string',
    taskDefinition='string',
    loadBalancers=[
        {
            'loadBalancerName': 'string',
            'containerName': 'string',
            'containerPort': 123
        },
    ],
    desiredCount=123,
    clientToken='string',

```

```
    role='string'  
)
```

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that you want to run your service on. If you do not specify a cluster, the default cluster is assumed.

- **serviceName** (*string*) – [REQUIRED]

The name of your service. Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed. Service names must be unique within a cluster, but you can have similarly named services in multiple clusters within a region or across multiple regions.

- **taskDefinition** (*string*) – [REQUIRED]

The family and revision (family:revision) or full Amazon Resource Name (ARN) of the task definition that you want to run in your service. If a revision is not specified, the latest ACTIVE revision is used.

- **loadBalancers** (*list*) – A list of load balancer objects, containing the load balancer name, the container name (as it appears in a container definition), and the container port to access from the load balancer.

– (*dict*) –

Details on a load balancer that is used with a service.

- * **loadBalancerName** (*string*) –

The name of the load balancer.

- * **containerName** (*string*) –

The name of the container to associate with the load balancer.

- * **containerPort** (*integer*) –

The port on the container to associate with the load balancer. This port must correspond to a `containerPort` in the service's task definition. Your container instances must allow ingress traffic on the `hostPort` of the port mapping.

- **desiredCount** (*integer*) – [REQUIRED]

The number of instantiations of the specified task definition that you would like to place and keep running on your cluster.

- **clientToken** (*string*) – Unique, case-sensitive identifier you provide to ensure the idempotency of the request. Up to 32 ASCII characters are allowed.
- **role** (*string*) – The name or full Amazon Resource Name (ARN) of the IAM role that allows your Amazon ECS container agent to make calls to your load balancer on your behalf. This parameter is only required if you are using a load balancer with your service.

Return type dict

Returns

Response Syntax

```
{  
    'service': {  
        'serviceArn': 'string',  
        'serviceName': 'string',  
        'clusterArn': 'string',  
        'loadBalancers': [  
            {  
                'loadBalancerArn': 'string',  
                'containerName': 'string',  
                'containerPort': integer,  
                'hostPort': integer,  
                'protocol': 'string',  
                'targetGroupArn': 'string'  
            }  
        ]  
    }  
}
```

```

        'loadBalancerName': 'string',
        'containerName': 'string',
        'containerPort': 123
    },
],
'status': 'string',
'desiredCount': 123,
'runningCount': 123,
'pendingCount': 123,
'taskDefinition': 'string',
'deployments': [
    {
        'id': 'string',
        'status': 'string',
        'taskDefinition': 'string',
        'desiredCount': 123,
        'pendingCount': 123,
        'runningCount': 123,
        'createdAt': datetime(2015, 1, 1),
        'updatedAt': datetime(2015, 1, 1)
    },
],
'roleArn': 'string',
'events': [
    {
        'id': 'string',
        'createdAt': datetime(2015, 1, 1),
        'message': 'string'
    },
],
]
}

```

Response Structure

- *(dict)* –
- **service** *(dict)* –

The full description of your service following the create call.

- * **serviceArn** *(string)* –

The Amazon Resource Name (ARN) that identifies the service. The ARN contains the `arn:aws:ecs` namespace, followed by the region of the service, the AWS account ID of the service owner, the `service` namespace, and then the service name. For example, `arn:aws:ecs:region:012345678910:service/my-service`.

- * **serviceName** *(string)* –

A user-generated string that you can use to identify your service.

- * **clusterArn** *(string)* –

The Amazon Resource Name (ARN) of the of the cluster that hosts the service.

- * **loadBalancers** *(list)* –

A list of load balancer objects, containing the load balancer name, the container name (as it appears in a container definition), and the container port to access from the load balancer.

- *(dict)* –

Details on a load balancer that is used with a service.

- **loadBalancerName** (*string*) –

The name of the load balancer.

- **containerName** (*string*) –

The name of the container to associate with the load balancer.

- **containerPort** (*integer*) –

The port on the container to associate with the load balancer. This port must correspond to a `containerPort` in the service's task definition. Your container instances must allow ingress traffic on the `hostPort` of the port mapping.

- * **status** (*string*) –

The status of the service. The valid values are `ACTIVE`, `DRAINING`, or `INACTIVE`.

- * **desiredCount** (*integer*) –

The desired number of instantiations of the task definition to keep running on the service. This value is specified when the service is created with `CreateService`, and it can be modified with `UpdateService`.

- * **runningCount** (*integer*) –

The number of tasks in the cluster that are in the `RUNNING` state.

- * **pendingCount** (*integer*) –

The number of tasks in the cluster that are in the `PENDING` state.

- * **taskDefinition** (*string*) –

The task definition to use for tasks in the service. This value is specified when the service is created with `CreateService`, and it can be modified with `UpdateService`.

- * **deployments** (*list*) –

The current state of deployments for the service.

- (*dict*) –

The details of an Amazon ECS service deployment.

- **id** (*string*) –

The ID of the deployment.

- **status** (*string*) –

The status of the deployment. Valid values are `PRIMARY` (for the most recent deployment), `ACTIVE` (for previous deployments that still have tasks running, but are being replaced with the `PRIMARY` deployment), and `INACTIVE` (for deployments that have been completely replaced).

- **taskDefinition** (*string*) –

The most recent task definition that was specified for the service to use.

- **desiredCount** (*integer*) –

The most recent desired count of tasks that was specified for the service to deploy and/or maintain.

- **pendingCount** (*integer*) –

The number of tasks in the deployment that are in the PENDING status.

- **runningCount** (*integer*) –

The number of tasks in the deployment that are in the RUNNING status.

- **createdAt** (*datetime*) –

The Unix time in seconds and milliseconds when the service was created.

- **updatedAt** (*datetime*) –

The Unix time in seconds and milliseconds when the service was last updated.

- * **roleArn** (*string*) –

The Amazon Resource Name (ARN) of the IAM role associated with the service that allows the Amazon ECS container agent to register container instances with a load balancer.

- * **events** (*list*) –

The event stream for your service. A maximum of 100 of the latest events are displayed.

- (*dict*) –

Details on an event associated with a service.

- **id** (*string*) –

The ID string of the event.

- **createdAt** (*datetime*) –

The Unix time in seconds and milliseconds when the event was triggered.

- **message** (*string*) –

The event message.

delete_cluster (***kwargs*)

Deletes the specified cluster. You must deregister all container instances from this cluster before you may delete it. You can list the container instances in a cluster with ListContainerInstances and deregister them with DeregisterContainerInstance .

Request Syntax

```
response = client.delete_cluster(
    cluster='string'
)
```

Parameters **cluster** (*string*) – [REQUIRED]

The short name or full Amazon Resource Name (ARN) of the cluster that you want to delete.

Return type dict

Returns

Response Syntax

```
{
  'cluster': {
    'clusterArn': 'string',
    'clusterName': 'string',
```

```
        'status': 'string',
        'registeredContainerInstancesCount': 123,
        'runningTasksCount': 123,
        'pendingTasksCount': 123,
        'activeServicesCount': 123
    }
}
```

Response Structure

- (*dict*) –
 - **cluster** (*dict*) –

The full description of the deleted cluster.

- * **clusterArn** (*string*) –

The Amazon Resource Name (ARN) that identifies the cluster. The ARN contains the `arn:aws:ecs` namespace, followed by the region of the cluster, the AWS account ID of the cluster owner, the `cluster` namespace, and then the cluster name. For example, `arn:aws:ecs:region:012345678910:cluster/test`.

- * **clusterName** (*string*) –

A user-generated string that you can use to identify your cluster.

- * **status** (*string*) –

The status of the cluster. The valid values are `ACTIVE` or `INACTIVE`. `ACTIVE` indicates that you can register container instances with the cluster and the associated instances can accept tasks.

- * **registeredContainerInstancesCount** (*integer*) –

The number of container instances registered into the cluster.

- * **runningTasksCount** (*integer*) –

The number of tasks in the cluster that are in the `RUNNING` state.

- * **pendingTasksCount** (*integer*) –

The number of tasks in the cluster that are in the `PENDING` state.

- * **activeServicesCount** (*integer*) –

The number of services that are running on the cluster in an `ACTIVE` state. You can view these services with `ListServices`.

delete_service (***kwargs*)

Deletes a specified service within a cluster.

Request Syntax

```
response = client.delete_service(
    cluster='string',
    service='string'
)
```

Parameters

- **cluster** (*string*) – The name of the cluster that hosts the service you want to delete.
- **service** (*string*) – [REQUIRED]

The name of the service you want to delete.

Return type dict

Returns

Response Syntax

```
{
  'service': {
    'serviceArn': 'string',
    'serviceName': 'string',
    'clusterArn': 'string',
    'loadBalancers': [
      {
        'loadBalancerName': 'string',
        'containerName': 'string',
        'containerPort': 123
      },
    ],
    'status': 'string',
    'desiredCount': 123,
    'runningCount': 123,
    'pendingCount': 123,
    'taskDefinition': 'string',
    'deployments': [
      {
        'id': 'string',
        'status': 'string',
        'taskDefinition': 'string',
        'desiredCount': 123,
        'pendingCount': 123,
        'runningCount': 123,
        'createdAt': datetime(2015, 1, 1),
        'updatedAt': datetime(2015, 1, 1)
      },
    ],
    'roleArn': 'string',
    'events': [
      {
        'id': 'string',
        'createdAt': datetime(2015, 1, 1),
        'message': 'string'
      },
    ],
  ]
}
```

Response Structure

- (*dict*) –

- **service** (*dict*) –

Details on a service within a cluster

- * **serviceArn** (*string*) –

The Amazon Resource Name (ARN) that identifies the service. The ARN contains the `arn:aws:ecs` namespace, followed by the region of the service, the AWS account ID of the service owner, the `service` namespace, and then the service name. For example, `arn:aws:ecs:region:012345678910:service/my-service`.

- * **serviceName** (*string*) –

A user-generated string that you can use to identify your service.

* **clusterArn** (*string*) –

The Amazon Resource Name (ARN) of the of the cluster that hosts the service.

* **loadBalancers** (*list*) –

A list of load balancer objects, containing the load balancer name, the container name (as it appears in a container definition), and the container port to access from the load balancer.

· (*dict*) –

Details on a load balancer that is used with a service.

· **loadBalancerName** (*string*) –

The name of the load balancer.

· **containerName** (*string*) –

The name of the container to associate with the load balancer.

· **containerPort** (*integer*) –

The port on the container to associate with the load balancer. This port must correspond to a `containerPort` in the service's task definition. Your container instances must allow ingress traffic on the `hostPort` of the port mapping.

* **status** (*string*) –

The status of the service. The valid values are `ACTIVE`, `DRAINING`, or `INACTIVE`.

* **desiredCount** (*integer*) –

The desired number of instantiations of the task definition to keep running on the service. This value is specified when the service is created with `CreateService`, and it can be modified with `UpdateService`.

* **runningCount** (*integer*) –

The number of tasks in the cluster that are in the `RUNNING` state.

* **pendingCount** (*integer*) –

The number of tasks in the cluster that are in the `PENDING` state.

* **taskDefinition** (*string*) –

The task definition to use for tasks in the service. This value is specified when the service is created with `CreateService`, and it can be modified with `UpdateService`.

* **deployments** (*list*) –

The current state of deployments for the service.

· (*dict*) –

The details of an Amazon ECS service deployment.

· **id** (*string*) –

The ID of the deployment.

· **status** (*string*) –

The status of the deployment. Valid values are `PRIMARY` (for the most recent deployment), `ACTIVE` (for previous deployments that still have tasks running, but are being replaced with the `PRIMARY` deployment), and `INACTIVE` (for deployments that have been completely replaced).

- **taskDefinition** (*string*) –
The most recent task definition that was specified for the service to use.
- **desiredCount** (*integer*) –
The most recent desired count of tasks that was specified for the service to deploy and/or maintain.
- **pendingCount** (*integer*) –
The number of tasks in the deployment that are in the PENDING status.
- **runningCount** (*integer*) –
The number of tasks in the deployment that are in the RUNNING status.
- **createdAt** (*datetime*) –
The Unix time in seconds and milliseconds when the service was created.
- **updatedAt** (*datetime*) –
The Unix time in seconds and milliseconds when the service was last updated.

* **roleArn** (*string*) –
The Amazon Resource Name (ARN) of the IAM role associated with the service that allows the Amazon ECS container agent to register container instances with a load balancer.

* **events** (*list*) –
The event stream for your service. A maximum of 100 of the latest events are displayed.

- (*dict*) –
Details on an event associated with a service.
- **id** (*string*) –
The ID string of the event.
- **createdAt** (*datetime*) –
The Unix time in seconds and milliseconds when the event was triggered.
- **message** (*string*) –
The event message.

deregister_container_instance (**kwargs)

Deregisters an Amazon ECS container instance from the specified cluster. This instance will no longer be available to run tasks.

If you intend to use the container instance for some other purpose after deregistration, you should stop all of the tasks running on the container instance before deregistration to avoid any orphaned tasks from consuming resources.

Deregistering a container instance removes the instance from a cluster, but it does not terminate the EC2 instance; if you are finished using the instance, be sure to terminate it in the Amazon EC2 console to stop billing.

Note: When you terminate a container instance, it is automatically deregistered from your cluster.

Request Syntax

```
response = client.deregister_container_instance(
    cluster='string',
    containerInstance='string',
    force=True|False
)
```

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that hosts the container instance you want to deregister. If you do not specify a cluster, the default cluster is assumed.
- **containerInstance** (*string*) – [REQUIRED]

The container instance UUID or full Amazon Resource Name (ARN) of the container instance you want to deregister. The ARN contains the `arn:aws:ecs` namespace, followed by the region of the container instance, the AWS account ID of the container instance owner, the `container-instance` namespace, and then the container instance UUID. For example, `arn:aws:ecs:region:aws_account_id:container-instance/container_instance_UUID`.

- **force** (*boolean*) – Force the deregistration of the container instance. If you have tasks running on the container instance when you deregister it with the `force` option, these tasks remain running and they will continue to pass Elastic Load Balancing load balancer health checks until you terminate the instance or the tasks stop through some other means, but they are orphaned (no longer monitored or accounted for by Amazon ECS). If an orphaned task on your container instance is part of an Amazon ECS service, then the service scheduler will start another copy of that task on a different container instance if possible.

Return type dict

Returns

Response Syntax

```
{
  'containerInstance': {
    'containerInstanceArn': 'string',
    'ec2InstanceId': 'string',
    'versionInfo': {
      'agentVersion': 'string',
      'agentHash': 'string',
      'dockerVersion': 'string'
    },
    'remainingResources': [
      {
        'name': 'string',
        'type': 'string',
        'doubleValue': 123.0,
        'longValue': 123,
        'integerValue': 123,
        'stringSetValue': [
          'string',
        ]
      }
    ],
    'registeredResources': [
      {
        'name': 'string',
```

```

        'type': 'string',
        'doubleValue': 123.0,
        'longValue': 123,
        'integerValue': 123,
        'stringSetValue': [
            'string',
        ],
    },
],
'status': 'string',
'agentConnected': True|False,
'runningTasksCount': 123,
'pendingTasksCount': 123,
'agentUpdateStatus': 'PENDING' | 'STAGING' | 'STAGED' | 'UPDATING' | 'UPDATED' | 'FA
}
}

```

Response Structure

- (*dict*) –

- **containerInstance** (*dict*) –

An Amazon EC2 instance that is running the Amazon ECS agent and has been registered with a cluster.

- * **containerInstanceArn** (*string*) –

The Amazon Resource Name (ARN) of the container instance. The ARN contains the `arn:aws:ecs` namespace, followed by the region of the container instance, the AWS account ID of the container instance owner, the `container-instance` namespace, and then the container instance UUID. For example, `arn:aws:ecs:region:aws_account_id:container-instance/container_instance_UUID`.

- * **ec2InstanceId** (*string*) –

The Amazon EC2 instance ID of the container instance.

- * **versionInfo** (*dict*) –

The version information for the Amazon ECS container agent and Docker daemon running on the container instance.

- **agentVersion** (*string*) –

The version number of the Amazon ECS container agent.

- **agentHash** (*string*) –

The Git commit hash for the Amazon ECS container agent build on the [amazon-ecs-agent](#) GitHub repository.

- **dockerVersion** (*string*) –

The Docker version running on the container instance.

- * **remainingResources** (*list*) –

The remaining resources of the container instance that are available for new tasks.

- (*dict*) –

Describes the resources available for a container instance.

- **name** (*string*) –

The name of the resource, such as CPU , MEMORY , PORTS , or a user-defined resource.

- **type** (*string*) –

The type of the resource, such as `INTEGER`, `DOUBLE`, `LONG`, or `STRINGSET`.

- **doubleValue** (*float*) –

When the `doubleValue` type is set, the value of the resource must be a double precision floating-point type.

- **longValue** (*integer*) –

When the `longValue` type is set, the value of the resource must be an extended precision floating-point type.

- **integerValue** (*integer*) –

When the `integerValue` type is set, the value of the resource must be an integer.

- **stringSetValue** (*list*) –

When the `stringSetValue` type is set, the value of the resource must be a string type.

- (*string*) –

- * **registeredResources** (*list*) –

The registered resources on the container instance that are in use by current tasks.

- (*dict*) –

Describes the resources available for a container instance.

- **name** (*string*) –

The name of the resource, such as `CPU`, `MEMORY`, `PORTS`, or a user-defined resource.

- **type** (*string*) –

The type of the resource, such as `INTEGER`, `DOUBLE`, `LONG`, or `STRINGSET`.

- **doubleValue** (*float*) –

When the `doubleValue` type is set, the value of the resource must be a double precision floating-point type.

- **longValue** (*integer*) –

When the `longValue` type is set, the value of the resource must be an extended precision floating-point type.

- **integerValue** (*integer*) –

When the `integerValue` type is set, the value of the resource must be an integer.

- **stringSetValue** (*list*) –

When the `stringSetValue` type is set, the value of the resource must be a string type.

- (*string*) –

- * **status** (*string*) –

The status of the container instance. The valid values are `ACTIVE` or `INACTIVE`. `ACTIVE` indicates that the container instance can accept tasks.

- * **agentConnected** (*boolean*) –

This parameter returns `true` if the agent is actually connected to Amazon ECS. Registered instances with an agent that may be un-

healthy or stopped will return `false`, and instances without a connected agent cannot accept placement request.

* **runningTasksCount** (*integer*) –

The number of tasks on the container instance that are in the `RUNNING` status.

* **pendingTasksCount** (*integer*) –

The number of tasks on the container instance that are in the `PENDING` status.

* **agentUpdateStatus** (*string*) –

The status of the most recent agent update. If an update has never been requested, this value is `NULL`.

deregister_task_definition (***kwargs*)

Deregisters the specified task definition by family and revision. Upon deregistration, the task definition is marked as `INACTIVE`. Existing tasks and services that reference an `INACTIVE` task definition continue to run without disruption. Existing services that reference an `INACTIVE` task definition can still scale up or down by modifying the service's desired count.

You cannot use an `INACTIVE` task definition to run new tasks or create new services, and you cannot update an existing service to reference an `INACTIVE` task definition (although there may be up to a 10 minute window following deregistration where these restrictions have not yet taken effect).

Request Syntax

```
response = client.deregister_task_definition(
    taskDefinition='string'
)
```

Parameters **taskDefinition** (*string*) – [REQUIRED]

The family and revision (`family:revision`) or full Amazon Resource Name (ARN) of the task definition that you want to deregister. You must specify a revision.

Return type dict

Returns

Response Syntax

```
{
  'taskDefinition': {
    'taskDefinitionArn': 'string',
    'containerDefinitions': [
      {
        'name': 'string',
        'image': 'string',
        'cpu': 123,
        'memory': 123,
        'links': [
          'string',
        ],
        'portMappings': [
          {
            'containerPort': 123,
            'hostPort': 123,
            'protocol': 'tcp' | 'udp'
          },
        ],
      },
    ],
  },
}
```

```
    ],
    'essential': True|False,
    'entryPoint': [
        'string',
    ],
    'command': [
        'string',
    ],
    'environment': [
        {
            'name': 'string',
            'value': 'string'
        },
    ],
    'mountPoints': [
        {
            'sourceVolume': 'string',
            'containerPath': 'string',
            'readOnly': True|False
        },
    ],
    'volumesFrom': [
        {
            'sourceContainer': 'string',
            'readOnly': True|False
        },
    ],
    ],
    },
    'family': 'string',
    'revision': 123,
    'volumes': [
        {
            'name': 'string',
            'host': {
                'sourcePath': 'string'
            }
        },
    ],
    'status': 'ACTIVE'|'INACTIVE'
}
}
```

Response Structure

- (*dict*) –
 - **taskDefinition** (*dict*) –

The full description of the deregistered task.

- * **taskDefinitionArn** (*string*) –

The full Amazon Resource Name (ARN) of the of the task definition.

- * **containerDefinitions** (*list*) –

A list of container definitions in JSON format that describe the different containers that make up your task. For more information on container definition parameters and defaults, see [Amazon ECS Task Definitions](#) in the *Amazon EC2 Container Service Developer Guide*.

- *(dict)* –

Container definitions are used in task definitions to describe the different containers that are launched as part of a task.

- **name** (*string*) –

The name of a container. If you are linking multiple containers together in a task definition, the `name` of one container can be entered in the `links` of another container to connect the containers.

- **image** (*string*) –

The image used to start a container. This string is passed directly to the Docker daemon. Images in the Docker Hub registry are available by default. Other repositories are specified with `*repository-url* /*image* :*tag*`.

- **cpu** (*integer*) –

The number of `cpu` units reserved for the container. A container instance has 1,024 `cpu` units for every CPU core. This parameter specifies the minimum amount of CPU to reserve for a container, and containers share unallocated CPU units with other containers on the instance with the same ratio as their allocated amount.

For example, if you run a single-container task on a single-core instance type with 512 CPU units specified for that container, and that is the only task running on the container instance, that container could use the full 1,024 CPU unit share at any given time. However, if you launched another copy of the same task on that container instance, each task would be guaranteed a minimum of 512 CPU units when needed, and each container could float to higher CPU usage if the other container was not using it, but if both tasks were 100% active all of the time, they would be limited to 512 CPU units.

The Docker daemon on the container instance uses the CPU value to calculate the relative CPU share ratios for running containers. For more information, see [CPU share constraint](#) in the Docker documentation. The minimum valid CPU share value that the Linux kernel will allow is 2; however, the CPU parameter is not required, and you can use CPU values below 2 in your container definitions. For CPU values below 2 (including null), the behavior varies based on your Amazon ECS container agent version:

- **Agent versions less than or equal to 1.1.0:** Null and zero CPU values are passed to Docker as 0, which Docker then converts to 1,024 CPU shares. CPU values of 1 are passed to Docker as 1, which the Linux kernel converts to 2 CPU shares.
- **Agent versions greater than or equal to 1.2.0:** Null, zero, and CPU values of 1 are passed to Docker as 2.
- **memory** (*integer*) –

The number of MiB of memory reserved for the container. If your container attempts to exceed the memory allocated here, the container is killed.

- **links** (*list*) –

The `link` parameter allows containers to communicate with each other without the need for port mappings, using the `name` parameter. The `name:internalName` construct is analogous to `name:alias` in Docker links. For more information on linking Docker containers, see <https://docs.docker.com/userguide/dockerlinks/>.

Warning: Containers that are collocated on a single container instance may be able to communicate with each other without requiring links or host port mappings. Network isolation is achieved on the container instance using security groups and VPC settings.

- (*string*) –

- **portMappings** (*list*) –

The list of port mappings for the container.

- (*dict*) –

Port mappings allow containers to access ports on the host container instance to send or receive traffic. Port mappings are specified as part of the container definition.

- **containerPort** (*integer*) –

The port number on the container that is bound to the user-specified or automatically assigned host port. If you specify a container port and not a host port, your container will automatically receive a host port in the ephemeral port range (for more information, see `hostPort`).

- **hostPort** (*integer*) –

The port number on the container instance to reserve for your container. You can specify a non-reserved host port for your container port mapping, or you can omit the `hostPort` (or set it to 0) while specifying a `containerPort` and your container will automatically receive a port in the ephemeral port range for your container instance operating system and Docker version.

The default ephemeral port range is 49153 to 65535, and this range is used for Docker versions prior to 1.6.0. For Docker version 1.6.0 and later, the Docker daemon tries to read the ephemeral port range from `/proc/sys/net/ipv4/ip_local_port_range`; if this kernel parameter is unavailable, the default ephemeral port range is used. You should not attempt to specify a host port in the ephemeral port range, since these are reserved for automatic assignment. In general, ports below 32768 are outside of the ephemeral port range.

The default reserved ports are 22 for SSH, the Docker ports 2375 and 2376, and the Amazon ECS Container Agent port 51678. Any host port that was previously specified in a running task is also reserved while the task is running (once a task stops, the host port is released). The current reserved ports are displayed in the `remainingResources` of `DescribeCon-`

tainerInstances output, and a container instance may have up to 50 reserved ports at a time, including the default reserved ports (automatically assigned ports do not count toward this limit).

- **protocol** (*string*) –

The protocol used for the port mapping. Valid values are `tcp` and `udp`. The default is `tcp`.

- **essential** (*boolean*) –

If the `essential` parameter of a container is marked as `true`, the failure of that container will stop the task. If the `essential` parameter of a container is marked as `false`, then its failure will not affect the rest of the containers in a task. If this parameter is omitted, a container is assumed to be `essential`.

Note: All tasks must have at least one essential container.

- **entryPoint** (*list*) –

Warning: Early versions of the Amazon ECS container agent do not properly handle `entryPoint` parameters. If you have problems using `entryPoint`, update your container agent or enter your commands and arguments as `command` array items instead.

The `ENTRYPOINT` that is passed to the container. For more information on the Docker `ENTRYPOINT` parameter, see <https://docs.docker.com/reference/builder/#entrypoint>.

- (*string*) –

- **command** (*list*) –

The `CMD` that is passed to the container. For more information on the Docker `CMD` parameter, see <https://docs.docker.com/reference/builder/#cmd>.

- (*string*) –

- **environment** (*list*) –

The environment variables to pass to a container.

- (*dict*) –

A key and value pair object.

- **name** (*string*) –

The name of the key value pair. For environment variables, this is the name of the environment variable.

- **value** (*string*) –

The value of the key value pair. For environment variables, this is the value of the environment variable.

- **mountPoints** (*list*) –

The mount points for data volumes in your container.

- (*dict*) –

Details on a volume mount point that is used in a container definition.

- **sourceVolume** (*string*) –
The name of the volume to mount.
- **containerPath** (*string*) –
The path on the container to mount the host volume at.
- **readOnly** (*boolean*) –
If this value is `true`, the container has read-only access to the volume. If this value is `false`, then the container can write to the volume. The default value is `false`.
- **volumesFrom** (*list*) –
Data volumes to mount from another container.
- (*dict*) –
Details on a data volume from another container.
- **sourceContainer** (*string*) –
The name of the container to mount volumes from.
- **readOnly** (*boolean*) –
If this value is `true`, the container has read-only access to the volume. If this value is `false`, then the container can write to the volume. The default value is `false`.
- * **family** (*string*) –
The family of your task definition. You can think of the `family` as the name of your task definition.
- * **revision** (*integer*) –
The revision of the task in a particular family. You can think of the revision as a version number of a task definition in a family. When you register a task definition for the first time, the revision is 1, and each time you register a new revision of a task definition in the same family, the revision value always increases by one (even if you have deregistered previous revisions in this family).
- * **volumes** (*list*) –
The list of volumes in a task. For more information on volume definition parameters and defaults, see [Amazon ECS Task Definitions](#) in the *Amazon EC2 Container Service Developer Guide*.
 - (*dict*) –
A data volume used in a task definition.
 - **name** (*string*) –
The name of the volume. This name is referenced in the `sourceVolume` parameter of container definition `mountPoints`.
 - **host** (*dict*) –
The path on the host container instance that is presented to the containers which access the volume. If this parameter is empty, then the Docker daemon assigns a host path for you.
 - **sourcePath** (*string*) –
The path on the host container instance that is presented to the container. If this parameter is empty, then the Docker daemon has assigned a host path for you.

* **status** (*string*) –
The status of the task definition.

describe_clusters (***kwargs*)
Describes one or more of your clusters.

Request Syntax

```
response = client.describe_clusters(
    clusters=[
        'string',
    ]
)
```

Parameters **clusters** (*list*) – A space-separated list of cluster names or full cluster Amazon Resource Name (ARN) entries. If you do not specify a cluster, the default cluster is assumed.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'clusters': [
        {
            'clusterArn': 'string',
            'clusterName': 'string',
            'status': 'string',
            'registeredContainerInstancesCount': 123,
            'runningTasksCount': 123,
            'pendingTasksCount': 123,
            'activeServicesCount': 123
        },
    ],
    'failures': [
        {
            'arn': 'string',
            'reason': 'string'
        },
    ],
}
```

Response Structure

- (*dict*) –
 - **clusters** (*list*) –

The list of clusters.

- * (*dict*) –

A regional grouping of one or more container instances on which you can run task requests. Each account receives a default cluster the first time you use the Amazon ECS service, but you may also create other clusters. Clusters may contain more than one instance type simultaneously.

- **clusterArn** (*string*) –

The Amazon Resource Name (ARN) that identifies the cluster. The ARN contains the `arn:aws:ecs` namespace, followed by the region of the cluster, the AWS account ID of the cluster owner, the `cluster` namespace, and then the cluster name. For example, `arn:aws:ecs:region :012345678910 :cluster/test`.

- **clusterName** (*string*) –

A user-generated string that you can use to identify your cluster.

- **status** (*string*) –

The status of the cluster. The valid values are `ACTIVE` or `INACTIVE`. `ACTIVE` indicates that you can register container instances with the cluster and the associated instances can accept tasks.

- **registeredContainerInstancesCount** (*integer*) –

The number of container instances registered into the cluster.

- **runningTasksCount** (*integer*) –

The number of tasks in the cluster that are in the `RUNNING` state.

- **pendingTasksCount** (*integer*) –

The number of tasks in the cluster that are in the `PENDING` state.

- **activeServicesCount** (*integer*) –

The number of services that are running on the cluster in an `ACTIVE` state. You can view these services with `ListServices`.

- **failures** (*list*) –

* (*dict*) –

A failed resource.

- **arn** (*string*) –

The Amazon Resource Name (ARN) of the failed resource.

- **reason** (*string*) –

The reason for the failure.

describe_container_instances (***kwargs*)

Describes Amazon EC2 Container Service container instances. Returns metadata about registered and remaining resources on each container instance requested.

Request Syntax

```
response = client.describe_container_instances(
    cluster='string',
    containerInstances=[
        'string',
    ]
)
```

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that hosts the container instances you want to describe. If you do not specify a cluster, the default cluster is assumed.

- **containerInstances** (*list*) – [REQUIRED]

A space-separated list of container instance UUIDs or full Amazon Resource Name (ARN) entries.

– (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'containerInstances': [
    {
      'containerInstanceArn': 'string',
      'ec2InstanceId': 'string',
      'versionInfo': {
        'agentVersion': 'string',
        'agentHash': 'string',
        'dockerVersion': 'string'
      },
      'remainingResources': [
        {
          'name': 'string',
          'type': 'string',
          'doubleValue': 123.0,
          'longValue': 123,
          'integerValue': 123,
          'stringSetValue': [
            'string',
          ]
        },
      ],
      'registeredResources': [
        {
          'name': 'string',
          'type': 'string',
          'doubleValue': 123.0,
          'longValue': 123,
          'integerValue': 123,
          'stringSetValue': [
            'string',
          ]
        },
      ],
      'status': 'string',
      'agentConnected': True|False,
      'runningTasksCount': 123,
      'pendingTasksCount': 123,
      'agentUpdateStatus': 'PENDING' | 'STAGING' | 'STAGED' | 'UPDATING' | 'UPDATED'
    },
  ],
  'failures': [
    {
      'arn': 'string',
      'reason': 'string'
    },
  ],
}
```

Response Structure

- *(dict)* –
 - **containerInstances** *(list)* –

The list of container instances.

- * *(dict)* –

An Amazon EC2 instance that is running the Amazon ECS agent and has been registered with a cluster.

- **containerInstanceArn** *(string)* –

The Amazon Resource Name (ARN) of the container instance. The ARN contains the `arn:aws:ecs` namespace, followed by the region of the container instance, the AWS account ID of the container instance owner, the `container-instance` namespace, and then the container instance UUID. For example, `arn:aws:ecs:region :aws_account_id :container-instance/container_instance_UUID`.

- **ec2InstanceId** *(string)* –

The Amazon EC2 instance ID of the container instance.

- **versionInfo** *(dict)* –

The version information for the Amazon ECS container agent and Docker daemon running on the container instance.

- **agentVersion** *(string)* –

The version number of the Amazon ECS container agent.

- **agentHash** *(string)* –

The Git commit hash for the Amazon ECS container agent build on the [amazon-ecs-agent](#) GitHub repository.

- **dockerVersion** *(string)* –

The Docker version running on the container instance.

- **remainingResources** *(list)* –

The remaining resources of the container instance that are available for new tasks.

- *(dict)* –

Describes the resources available for a container instance.

- **name** *(string)* –

The name of the resource, such as `CPU`, `MEMORY`, `PORTS`, or a user-defined resource.

- **type** *(string)* –

The type of the resource, such as `INTEGER`, `DOUBLE`, `LONG`, or `STRINGSET`.

- **doubleValue** *(float)* –

When the `doubleValue` type is set, the value of the resource must be a double precision floating-point type.

- **longValue** *(integer)* –

When the `longValue` type is set, the value of the resource must be an extended precision floating-point type.

- **integerValue** *(integer)* –

When the `integerValue` type is set, the value of the resource must be an integer.

- **stringSetValue** (*list*) –

When the `stringSetValue` type is set, the value of the resource must be a string type.

- (*string*) –
- **registeredResources** (*list*) –

The registered resources on the container instance that are in use by current tasks.

- (*dict*) –

Describes the resources available for a container instance.

- **name** (*string*) –

The name of the resource, such as CPU , MEMORY , PORTS , or a user-defined resource.

- **type** (*string*) –

The type of the resource, such as INTEGER , DOUBLE , LONG , or STRINGSET .

- **doubleValue** (*float*) –

When the `doubleValue` type is set, the value of the resource must be a double precision floating-point type.

- **longValue** (*integer*) –

When the `longValue` type is set, the value of the resource must be an extended precision floating-point type.

- **integerValue** (*integer*) –

When the `integerValue` type is set, the value of the resource must be an integer.

- **stringSetValue** (*list*) –

When the `stringSetValue` type is set, the value of the resource must be a string type.

- (*string*) –
- **status** (*string*) –

The status of the container instance. The valid values are ACTIVE or INACTIVE . ACTIVE indicates that the container instance can accept tasks.

- **agentConnected** (*boolean*) –

This parameter returns `true` if the agent is actually connected to Amazon ECS. Registered instances with an agent that may be unhealthy or stopped will return `false` , and instances without a connected agent cannot accept placement request.

- **runningTasksCount** (*integer*) –

The number of tasks on the container instance that are in the RUNNING status.

- **pendingTasksCount** (*integer*) –

The number of tasks on the container instance that are in the PENDING status.

- **agentUpdateStatus** (*string*) –

The status of the most recent agent update. If an update has never been requested, this value is `NULL`.

- **failures** (*list*) –
 - * (*dict*) –

A failed resource.

- **arn** (*string*) –

The Amazon Resource Name (ARN) of the failed resource.

- **reason** (*string*) –

The reason for the failure.

describe_services (***kwargs*)

Describes the specified services running in your cluster.

Request Syntax

```
response = client.describe_services(
    cluster='string',
    services=[
        'string',
    ]
)
```

Parameters

- **cluster** (*string*) – The name of the cluster that hosts the service you want to describe.
- **services** (*list*) – **[REQUIRED]**
A list of services you want to describe.
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'services': [
    {
      'serviceArn': 'string',
      'serviceName': 'string',
      'clusterArn': 'string',
      'loadBalancers': [
        {
          'loadBalancerName': 'string',
          'containerName': 'string',
          'containerPort': 123
        },
      ],
      'status': 'string',
      'desiredCount': 123,
      'runningCount': 123,
      'pendingCount': 123,
      'taskDefinition': 'string',
      'deployments': [
        {
          'id': 'string',
          'status': 'string',

```



```

        'taskDefinition': 'string',
        'desiredCount': 123,
        'pendingCount': 123,
        'runningCount': 123,
        'createdAt': datetime(2015, 1, 1),
        'updatedAt': datetime(2015, 1, 1)
    },
],
'roleArn': 'string',
'events': [
    {
        'id': 'string',
        'createdAt': datetime(2015, 1, 1),
        'message': 'string'
    },
]
},
],
'failures': [
    {
        'arn': 'string',
        'reason': 'string'
    },
]
}

```

Response Structure

- (*dict*) –

- **services** (*list*) –

The list of services described.

- * (*dict*) –

Details on a service within a cluster

- **serviceArn** (*string*) –

The Amazon Resource Name (ARN) that identifies the service. The ARN contains the `arn:aws:ecs` namespace, followed by the region of the service, the AWS account ID of the service owner, the `service` namespace, and then the service name. For example, `arn:aws:ecs:region :012345678910 :service/my-service`.

- **serviceName** (*string*) –

A user-generated string that you can use to identify your service.

- **clusterArn** (*string*) –

The Amazon Resource Name (ARN) of the of the cluster that hosts the service.

- **loadBalancers** (*list*) –

A list of load balancer objects, containing the load balancer name, the container name (as it appears in a container definition), and the container port to access from the load balancer.

- (*dict*) –

Details on a load balancer that is used with a service.

- **loadBalancerName** (*string*) –

The name of the load balancer.

- **containerName** (*string*) –

The name of the container to associate with the load balancer.

- **containerPort** (*integer*) –

The port on the container to associate with the load balancer. This port must correspond to a `containerPort` in the service's task definition. Your container instances must allow ingress traffic on the `hostPort` of the port mapping.

- **status** (*string*) –

The status of the service. The valid values are `ACTIVE`, `DRAINING`, or `INACTIVE`.

- **desiredCount** (*integer*) –

The desired number of instantiations of the task definition to keep running on the service. This value is specified when the service is created with `CreateService`, and it can be modified with `UpdateService`.

- **runningCount** (*integer*) –

The number of tasks in the cluster that are in the `RUNNING` state.

- **pendingCount** (*integer*) –

The number of tasks in the cluster that are in the `PENDING` state.

- **taskDefinition** (*string*) –

The task definition to use for tasks in the service. This value is specified when the service is created with `CreateService`, and it can be modified with `UpdateService`.

- **deployments** (*list*) –

The current state of deployments for the service.

- (*dict*) –

The details of an Amazon ECS service deployment.

- **id** (*string*) –

The ID of the deployment.

- **status** (*string*) –

The status of the deployment. Valid values are `PRIMARY` (for the most recent deployment), `ACTIVE` (for previous deployments that still have tasks running, but are being replaced with the `PRIMARY` deployment), and `INACTIVE` (for deployments that have been completely replaced).

- **taskDefinition** (*string*) –

The most recent task definition that was specified for the service to use.

- **desiredCount** (*integer*) –

The most recent desired count of tasks that was specified for the service to deploy and/or maintain.

- **pendingCount** (*integer*) –

The number of tasks in the deployment that are in the `PENDING` status.

- **runningCount** (*integer*) –
The number of tasks in the deployment that are in the `RUNNING` status.
- **createdAt** (*datetime*) –
The Unix time in seconds and milliseconds when the service was created.
- **updatedAt** (*datetime*) –
The Unix time in seconds and milliseconds when the service was last updated.
- **roleArn** (*string*) –
The Amazon Resource Name (ARN) of the IAM role associated with the service that allows the Amazon ECS container agent to register container instances with a load balancer.
- **events** (*list*) –
The event stream for your service. A maximum of 100 of the latest events are displayed.
- (*dict*) –
Details on an event associated with a service.
- **id** (*string*) –
The ID string of the event.
- **createdAt** (*datetime*) –
The Unix time in seconds and milliseconds when the event was triggered.
- **message** (*string*) –
The event message.
- **failures** (*list*) –
Any failures associated with the call.
- (*dict*) –
A failed resource.
- **arn** (*string*) –
The Amazon Resource Name (ARN) of the failed resource.
- **reason** (*string*) –
The reason for the failure.

describe_task_definition (***kwargs*)

Describes a task definition. You can specify a `family` and `revision` to find information on a specific task definition, or you can simply specify the family to find the latest `ACTIVE` revision in that family.

Note: You can only describe `INACTIVE` task definitions while an active task or service references them.

Request Syntax

```
response = client.describe_task_definition(
    taskDefinition='string'
)
```

Parameters `taskDefinition` (*string*) – **[REQUIRED]**

The family for the latest ACTIVE revision, family and revision (family:revision) for a specific revision in the family, or full Amazon Resource Name (ARN) of the task definition that you want to describe.

Return type dict

Returns

Response Syntax

```
{
  'taskDefinition': {
    'taskDefinitionArn': 'string',
    'containerDefinitions': [
      {
        'name': 'string',
        'image': 'string',
        'cpu': 123,
        'memory': 123,
        'links': [
          'string',
        ],
        'portMappings': [
          {
            'containerPort': 123,
            'hostPort': 123,
            'protocol': 'tcp'|'udp'
          },
        ],
        'essential': True|False,
        'entryPoint': [
          'string',
        ],
        'command': [
          'string',
        ],
        'environment': [
          {
            'name': 'string',
            'value': 'string'
          },
        ],
        'mountPoints': [
          {
            'sourceVolume': 'string',
            'containerPath': 'string',
            'readOnly': True|False
          },
        ],
        'volumesFrom': [
          {
            'sourceContainer': 'string',
            'readOnly': True|False
          },
        ],
      },
    ],
    'family': 'string',
    'revision': 123,
    'volumes': [
```

```

    {
        'name': 'string',
        'host': {
            'sourcePath': 'string'
        }
    },
],
'status': 'ACTIVE'|'INACTIVE'
}
}

```

Response Structure

- (dict) –
 - **taskDefinition** (dict) –

The full task definition description.

- * **taskDefinitionArn** (string) –

The full Amazon Resource Name (ARN) of the of the task definition.

- * **containerDefinitions** (list) –

A list of container definitions in JSON format that describe the different containers that make up your task. For more information on container definition parameters and defaults, see [Amazon ECS Task Definitions](#) in the *Amazon EC2 Container Service Developer Guide*.

- (dict) –

Container definitions are used in task definitions to describe the different containers that are launched as part of a task.

- **name** (string) –

The name of a container. If you are linking multiple containers together in a task definition, the `name` of one container can be entered in the `links` of another container to connect the containers.

- **image** (string) –

The image used to start a container. This string is passed directly to the Docker daemon. Images in the Docker Hub registry are available by default. Other repositories are specified with `*repository-url* /*image* :*tag*`.

- **cpu** (integer) –

The number of `cpu` units reserved for the container. A container instance has 1,024 `cpu` units for every CPU core. This parameter specifies the minimum amount of CPU to reserve for a container, and containers share unallocated CPU units with other containers on the instance with the same ratio as their allocated amount.

For example, if you run a single-container task on a single-core instance type with 512 CPU units specified for that container, and that is the only task running on the container instance, that container could use the full 1,024 CPU unit share at any given time. However, if you launched another copy of the same task on that container instance, each task would be

guaranteed a minimum of 512 CPU units when needed, and each container could float to higher CPU usage if the other container was not using it, but if both tasks were 100% active all of the time, they would be limited to 512 CPU units.

The Docker daemon on the container instance uses the CPU value to calculate the relative CPU share ratios for running containers. For more information, see [CPU share constraint](#) in the Docker documentation. The minimum valid CPU share value that the Linux kernel will allow is 2; however, the CPU parameter is not required, and you can use CPU values below 2 in your container definitions. For CPU values below 2 (including null), the behavior varies based on your Amazon ECS container agent version:

- **Agent versions less than or equal to 1.1.0:** Null and zero CPU values are passed to Docker as 0, which Docker then converts to 1,024 CPU shares. CPU values of 1 are passed to Docker as 1, which the Linux kernel converts to 2 CPU shares.
- **Agent versions greater than or equal to 1.2.0:** Null, zero, and CPU values of 1 are passed to Docker as 2.
- **memory** (*integer*) –

The number of MiB of memory reserved for the container. If your container attempts to exceed the memory allocated here, the container is killed.

- **links** (*list*) –

The `link` parameter allows containers to communicate with each other without the need for port mappings, using the `name` parameter. The `name:internalName` construct is analogous to `name:alias` in Docker links. For more information on linking Docker containers, see <https://docs.docker.com/userguide/dockerlinks/>.

Warning: Containers that are collocated on a single container instance may be able to communicate with each other without requiring links or host port mappings. Network isolation is achieved on the container instance using security groups and VPC settings.

- (*string*) –
- **portMappings** (*list*) –

The list of port mappings for the container.

- (*dict*) –

Port mappings allow containers to access ports on the host container instance to send or receive traffic. Port mappings are specified as part of the container definition.

- **containerPort** (*integer*) –

The port number on the container that is bound to the user-specified or automatically assigned host port. If you specify a container port and not a host port, your container will automatically receive a host port in the ephemeral port range (for more information, see `hostPort`).

- **hostPort** (*integer*) –

The port number on the container instance to reserve for your container. You can specify a non-reserved host port for your container port mapping, or you can omit the `hostPort` (or set it to 0) while specifying a `containerPort` and your container will automatically receive a port in the ephemeral port range for your container instance operating system and Docker version.

The default ephemeral port range is 49153 to 65535, and this range is used for Docker versions prior to 1.6.0. For Docker version 1.6.0 and later, the Docker daemon tries to read the ephemeral port range from `/proc/sys/net/ipv4/ip_local_port_range`; if this kernel parameter is unavailable, the default ephemeral port range is used. You should not attempt to specify a host port in the ephemeral port range, since these are reserved for automatic assignment. In general, ports below 32768 are outside of the ephemeral port range.

The default reserved ports are 22 for SSH, the Docker ports 2375 and 2376, and the Amazon ECS Container Agent port 51678. Any host port that was previously specified in a running task is also reserved while the task is running (once a task stops, the host port is released). The current reserved ports are displayed in the `remainingResources` of `DescribeContainerInstances` output, and a container instance may have up to 50 reserved ports at a time, including the default reserved ports (automatically assigned ports do not count toward this limit).

- **protocol** (*string*) –

The protocol used for the port mapping. Valid values are `tcp` and `udp`. The default is `tcp`.

- **essential** (*boolean*) –

If the `essential` parameter of a container is marked as `true`, the failure of that container will stop the task. If the `essential` parameter of a container is marked as `false`, then its failure will not affect the rest of the containers in a task. If this parameter is omitted, a container is assumed to be essential.

Note: All tasks must have at least one essential container.

- **entryPoint** (*list*) –

Warning: Early versions of the Amazon ECS container agent do not properly handle `entryPoint` parameters. If you have problems using `entryPoint`, update your container agent or enter your commands and arguments as `command array` items instead.

The `ENTRYPOINT` that is passed to the container. For more information on the Docker `ENTRYPOINT` parameter, see <https://docs.docker.com/reference/builder/#entrypoint>.

- (*string*) –

- **command** (*list*) –

The CMD that is passed to the container. For more information on the Docker CMD parameter, see <https://docs.docker.com/reference/builder/#cmd>.

- (*string*) –

- **environment** (*list*) –

The environment variables to pass to a container.

- (*dict*) –

A key and value pair object.

- **name** (*string*) –

The name of the key value pair. For environment variables, this is the name of the environment variable.

- **value** (*string*) –

The value of the key value pair. For environment variables, this is the value of the environment variable.

- **mountPoints** (*list*) –

The mount points for data volumes in your container.

- (*dict*) –

Details on a volume mount point that is used in a container definition.

- **sourceVolume** (*string*) –

The name of the volume to mount.

- **containerPath** (*string*) –

The path on the container to mount the host volume at.

- **readOnly** (*boolean*) –

If this value is `true`, the container has read-only access to the volume. If this value is `false`, then the container can write to the volume. The default value is `false`.

- **volumesFrom** (*list*) –

Data volumes to mount from another container.

- (*dict*) –

Details on a data volume from another container.

- **sourceContainer** (*string*) –

The name of the container to mount volumes from.

- **readOnly** (*boolean*) –

If this value is `true`, the container has read-only access to the volume. If this value is `false`, then the container can write to the volume. The default value is `false`.

- * **family** (*string*) –

The family of your task definition. You can think of the `family` as the name of your task definition.

- * **revision** (*integer*) –

The revision of the task in a particular family. You can think of the revision as a version number of a task definition in a family. When you register a task definition for the first time, the revision is 1, and each time you register a new revision of a task definition in the same

family, the revision value always increases by one (even if you have deregistered previous revisions in this family).

* **volumes** (*list*) –

The list of volumes in a task. For more information on volume definition parameters and defaults, see [Amazon ECS Task Definitions](#) in the *Amazon EC2 Container Service Developer Guide*.

· (*dict*) –

A data volume used in a task definition.

· **name** (*string*) –

The name of the volume. This name is referenced in the `sourceVolume` parameter of container definition `mountPoints`.

· **host** (*dict*) –

The path on the host container instance that is presented to the containers which access the volume. If this parameter is empty, then the Docker daemon assigns a host path for you.

· **sourcePath** (*string*) –

The path on the host container instance that is presented to the container. If this parameter is empty, then the Docker daemon has assigned a host path for you.

* **status** (*string*) –

The status of the task definition.

describe_tasks (***kwargs*)

Describes a specified task or tasks.

Request Syntax

```
response = client.describe_tasks(
    cluster='string',
    tasks=[
        'string',
    ]
)
```

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that hosts the task you want to describe. If you do not specify a cluster, the default cluster is assumed.
- **tasks** (*list*) – **[REQUIRED]**

A space-separated list of task UUIDs or full Amazon Resource Name (ARN) entries.

– (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'tasks': [
        {
            'taskArn': 'string',
```

```
'clusterArn': 'string',
'taskDefinitionArn': 'string',
'containerInstanceArn': 'string',
'overrides': {
  'containerOverrides': [
    {
      'name': 'string',
      'command': [
        'string',
      ],
      'environment': [
        {
          'name': 'string',
          'value': 'string'
        },
      ],
    },
  ],
},
],
'lastStatus': 'string',
'desiredStatus': 'string',
'containers': [
  {
    'containerArn': 'string',
    'taskArn': 'string',
    'name': 'string',
    'lastStatus': 'string',
    'exitCode': 123,
    'reason': 'string',
    'networkBindings': [
      {
        'bindIP': 'string',
        'containerPort': 123,
        'hostPort': 123,
        'protocol': 'tcp'|'udp'
      },
    ],
  },
],
'startedBy': 'string'
},
],
'failures': [
  {
    'arn': 'string',
    'reason': 'string'
  },
],
]
```

Response Structure

- (*dict*) –

- **tasks** (*list*) –

The list of tasks.

- * (*dict*) –

Details on a task in a cluster.

- **taskArn** (*string*) –

The Amazon Resource Name (ARN) of the task.

- **clusterArn** (*string*) –

The Amazon Resource Name (ARN) of the of the cluster that hosts the task.

- **taskDefinitionArn** (*string*) –

The Amazon Resource Name (ARN) of the of the task definition that creates the task.

- **containerInstanceArn** (*string*) –

The Amazon Resource Name (ARN) of the container instances that host the task.

- **overrides** (*dict*) –

One or more container overrides.

- **containerOverrides** (*list*) –

One or more container overrides sent to a task.

- (*dict*) –

The overrides that should be sent to a container.

- **name** (*string*) –

The name of the container that receives the override.

- **command** (*list*) –

The command to send to the container that overrides the default command from the Docker image or the task definition.

- (*string*) –

- **environment** (*list*) –

The environment variables to send to the container. You can add new environment variables, which are added to the container at launch, or you can override the existing environment variables from the Docker image or the task definition.

- (*dict*) –

A key and value pair object.

- **name** (*string*) –

The name of the key value pair. For environment variables, this is the name of the environment variable.

- **value** (*string*) –

The value of the key value pair. For environment variables, this is the value of the environment variable.

- **lastStatus** (*string*) –

The last known status of the task.

- **desiredStatus** (*string*) –

The desired status of the task.

- **containers** (*list*) –

The containers associated with the task.

- (*dict*) –

A docker container that is part of a task.

- **containerArn** (*string*) –

The Amazon Resource Name (ARN) of the container.

- **taskArn** (*string*) –
The Amazon Resource Name (ARN) of the task.
- **name** (*string*) –
The name of the container.
- **lastStatus** (*string*) –
The last known status of the container.
- **exitCode** (*integer*) –
The exit code returned from the container.
- **reason** (*string*) –
A short (255 max characters) human-readable string to provide additional detail about a running or stopped container.
- **networkBindings** (*list*) –
The network bindings associated with the container.
- (*dict*) –
Details on the network bindings between a container and its host container instance.
- **bindIP** (*string*) –
The IP address that the container is bound to on the container instance.
- **containerPort** (*integer*) –
The port number on the container that is be used with the network binding.
- **hostPort** (*integer*) –
The port number on the host that is used with the network binding.
- **protocol** (*string*) –
The protocol used for the network binding.
- **startedBy** (*string*) –
The tag specified when a task is started. If the task is started by an Amazon ECS service, then the `startedBy` parameter contains the deployment ID of the service that starts it.
- **failures** (*list*) –
 - * (*dict*) –
A failed resource.
 - **arn** (*string*) –
The Amazon Resource Name (ARN) of the failed resource.
 - **reason** (*string*) –
The reason for the failure.

discover_poll_endpoint (***kwargs*)

Note: This action is only used by the Amazon EC2 Container Service agent, and it is not intended for use outside of the agent.

Returns an endpoint for the Amazon EC2 Container Service agent to poll for updates.

Request Syntax

```
response = client.discover_poll_endpoint(
    containerInstance='string',
    cluster='string'
)
```

Parameters

- **containerInstance** (*string*) – The container instance UUID or full Amazon Resource Name (ARN) of the container instance. The ARN contains the `arn:aws:ecs:region:aws_account_id:container-instance/container_instance_UUID`.
- **cluster** (*string*) – The cluster that the container instance belongs to.

Return type dict

Returns

Response Syntax

```
{
    'endpoint': 'string',
    'telemetryEndpoint': 'string'
}
```

Response Structure

- (*dict*) –
 - **endpoint** (*string*) –
The endpoint for the Amazon ECS agent to poll.
 - **telemetryEndpoint** (*string*) –
The telemetry endpoint for the Amazon ECS agent.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type L{botocore.paginate.Paginator}

Returns A paginator object.

get_waiter (*waiter_name*)

list_clusters (**kwargs)

Returns a list of existing clusters.

Request Syntax

```
response = client.list_clusters(  
    nextToken='string',  
    maxResults=123  
)
```

Parameters

- **nextToken** (*string*) – The `nextToken` value returned from a previous paginated `ListClusters` request where `maxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `nextToken` value. This value is `null` when there are no more results to return.
- **maxResults** (*integer*) – The maximum number of cluster results returned by `ListClusters` in paginated output. When this parameter is used, `ListClusters` only returns `maxResults` results in a single page along with a `nextToken` response element. The remaining results of the initial request can be seen by sending another `ListClusters` request with the returned `nextToken` value. This value can be between 1 and 100. If this parameter is not used, then `ListClusters` returns up to 100 results and a `nextToken` value if applicable.

Return type dict

Returns

Response Syntax

```
{  
    'clusterArns': [  
        'string',  
    ],  
    'nextToken': 'string'  
}
```

Response Structure

- (*dict*) –
 - **clusterArns** (*list*) –
The list of full Amazon Resource Name (ARN) entries for each cluster associated with your account.
 - * (*string*) –
 - **nextToken** (*string*) –
The `nextToken` value to include in a future `ListClusters` request. When the results of a `ListClusters` request exceed `maxResults`, this value can be used to retrieve the next page of results. This value is `null` when there are no more results to return.

list_container_instances (**kwargs)

Returns a list of container instances in a specified cluster.

Request Syntax

```
response = client.list_container_instances(
    cluster='string',
    nextToken='string',
    maxResults=123
)
```

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that hosts the container instances you want to list. If you do not specify a cluster, the default cluster is assumed..
- **nextToken** (*string*) – The nextToken value returned from a previous paginated ListContainerInstances request where maxResults was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the nextToken value. This value is null when there are no more results to return.
- **maxResults** (*integer*) – The maximum number of container instance results returned by ListContainerInstances in paginated output. When this parameter is used, ListContainerInstances only returns maxResults results in a single page along with a nextToken response element. The remaining results of the initial request can be seen by sending another ListContainerInstances request with the returned nextToken value. This value can be between 1 and 100. If this parameter is not used, then ListContainerInstances returns up to 100 results and a nextToken value if applicable.

Return type dict

Returns

Response Syntax

```
{
  'containerInstanceArns': [
    'string',
  ],
  'nextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **containerInstanceArns** (*list*) –

The list of container instance full Amazon Resource Name (ARN) entries for each container instance associated with the specified cluster.

 - * (*string*) –
 - **nextToken** (*string*) –

The nextToken value to include in a future ListContainerInstances request. When the results of a ListContainerInstances request exceed maxResults, this value can be used to retrieve the next page of results. This value is null when there are no more results to return.

list_services (***kwargs*)

Lists the services that are running in a specified cluster.

Request Syntax

```
response = client.list_services(  
    cluster='string',  
    nextToken='string',  
    maxResults=123  
)
```

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that hosts the services you want to list. If you do not specify a cluster, the default cluster is assumed..
- **nextToken** (*string*) – The `nextToken` value returned from a previous paginated `ListServices` request where `maxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `nextToken` value. This value is `null` when there are no more results to return.
- **maxResults** (*integer*) – The maximum number of container instance results returned by `ListServices` in paginated output. When this parameter is used, `ListServices` only returns `maxResults` results in a single page along with a `nextToken` response element. The remaining results of the initial request can be seen by sending another `ListServices` request with the returned `nextToken` value. This value can be between 1 and 100. If this parameter is not used, then `ListServices` returns up to 100 results and a `nextToken` value if applicable.

Return type dict

Returns

Response Syntax

```
{  
    'serviceArns': [  
        'string',  
    ],  
    'nextToken': 'string'  
}
```

Response Structure

- (*dict*) –
 - **serviceArns** (*list*) –
The list of full Amazon Resource Name (ARN) entries for each service associated with the specified cluster.
 * (*string*) –
 - **nextToken** (*string*) –
The `nextToken` value to include in a future `ListServices` request. When the results of a `ListServices` request exceed `maxResults`, this value can be used to retrieve the next page of results. This value is `null` when there are no more results to return.

list_task_definition_families (**kwargs)

Returns a list of task definition families that are registered to your account (which may include task definition families that no longer have any `ACTIVE` task definitions). You can filter the results with the `familyPrefix` parameter.

Request Syntax


```
response = client.list_task_definition_families(
    familyPrefix='string',
    nextToken='string',
    maxResults=123
)
```

Parameters

- **familyPrefix** (*string*) – The familyPrefix is a string that is used to filter the results of ListTaskDefinitionFamilies . If you specify a familyPrefix , only task definition family names that begin with the familyPrefix string are returned.
- **nextToken** (*string*) – The nextToken value returned from a previous paginated ListTaskDefinitionFamilies request where maxResults was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the nextToken value. This value is null when there are no more results to return.
- **maxResults** (*integer*) – The maximum number of task definition family results returned by ListTaskDefinitionFamilies in paginated output. When this parameter is used, ListTaskDefinitions only returns maxResults results in a single page along with a nextToken response element. The remaining results of the initial request can be seen by sending another ListTaskDefinitionFamilies request with the returned nextToken value. This value can be between 1 and 100. If this parameter is not used, then ListTaskDefinitionFamilies returns up to 100 results and a nextToken value if applicable.

Return type dict

Returns

Response Syntax

```
{
  'families': [
    'string',
  ],
  'nextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **families** (*list*) –

The list of task definition family names that match the ListTaskDefinitionFamilies request.

* (*string*) –
 - **nextToken** (*string*) –

The nextToken value to include in a future ListTaskDefinitionFamilies request. When the results of a ListTaskDefinitionFamilies request exceed maxResults , this value can be used to retrieve the next page of results. This value is null when there are no more results to return.

list_task_definitions (**kwargs)

Returns a list of task definitions that are registered to your account. You can filter the results by family name with the familyPrefix parameter or by status with the status parameter.

Request Syntax

```
response = client.list_task_definitions(
    familyPrefix='string',
    status='ACTIVE'|'INACTIVE',
    sort='ASC'|'DESC',
    nextToken='string',
    maxResults=123
)
```

Parameters

- **familyPrefix** (*string*) – The full family name that you want to filter the `ListTaskDefinitions` results with. Specifying a `familyPrefix` will limit the listed task definitions to task definition revisions that belong to that family.
- **status** (*string*) – The task definition status that you want to filter the `ListTaskDefinitions` results with. By default, only `ACTIVE` task definitions are listed. By setting this parameter to `INACTIVE`, you can view task definitions that are `INACTIVE` as long as an active task or service still references them. If you paginate the resulting output, be sure to keep the `status` value constant in each subsequent request.
- **sort** (*string*) – The order in which to sort the results. Valid values are `ASC` and `DESC`. By default (`ASC`), task definitions are listed lexicographically by family name and in ascending numerical order by revision so that the newest task definitions in a family are listed last. Setting this parameter to `DESC` reverses the sort order on family name and revision so that the newest task definitions in a family are listed first.
- **nextToken** (*string*) – The `nextToken` value returned from a previous paginated `ListTaskDefinitions` request where `maxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `nextToken` value. This value is `null` when there are no more results to return.
- **maxResults** (*integer*) – The maximum number of task definition results returned by `ListTaskDefinitions` in paginated output. When this parameter is used, `ListTaskDefinitions` only returns `maxResults` results in a single page along with a `nextToken` response element. The remaining results of the initial request can be seen by sending another `ListTaskDefinitions` request with the returned `nextToken` value. This value can be between 1 and 100. If this parameter is not used, then `ListTaskDefinitions` returns up to 100 results and a `nextToken` value if applicable.

Return type dict

Returns

Response Syntax

```
{
    'taskDefinitionArns': [
        'string',
    ],
    'nextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **taskDefinitionArns** (*list*) –

The list of task definition Amazon Resource Name (ARN) entries for the

ListTaskDefinitions request.

* (*string*) –
– **nextToken** (*string*) –

The nextToken value to include in a future ListTaskDefinitions request. When the results of a ListTaskDefinitions request exceed maxResults, this value can be used to retrieve the next page of results. This value is null when there are no more results to return.

list_tasks (**kwargs)

Returns a list of tasks for a specified cluster. You can filter the results by family name, by a particular container instance, or by the desired status of the task with the family, containerInstance, and desiredStatus parameters.

Request Syntax

```
response = client.list_tasks(
    cluster='string',
    containerInstance='string',
    family='string',
    nextToken='string',
    maxResults=123,
    startedBy='string',
    serviceName='string',
    desiredStatus='RUNNING' | 'PENDING' | 'STOPPED'
)
```

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that hosts the tasks you want to list. If you do not specify a cluster, the default cluster is assumed..
- **containerInstance** (*string*) – The container instance UUID or full Amazon Resource Name (ARN) of the container instance that you want to filter the ListTasks results with. Specifying a containerInstance will limit the results to tasks that belong to that container instance.
- **family** (*string*) – The name of the family that you want to filter the ListTasks results with. Specifying a family will limit the results to tasks that belong to that family.
- **nextToken** (*string*) – The nextToken value returned from a previous paginated ListTasks request where maxResults was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the nextToken value. This value is null when there are no more results to return.
- **maxResults** (*integer*) – The maximum number of task results returned by ListTasks in paginated output. When this parameter is used, ListTasks only returns maxResults results in a single page along with a nextToken response element. The remaining results of the initial request can be seen by sending another ListTasks request with the returned nextToken value. This value can be between 1 and 100. If this parameter is not used, then ListTasks returns up to 100 results and a nextToken value if applicable.
- **startedBy** (*string*) – The startedBy value that you want to filter the task results with. Specifying a startedBy value will limit the results to tasks that were started with that value.
- **serviceName** (*string*) – The name of the service that you want to filter the ListTasks results with. Specifying a serviceName will limit the results to tasks that belong to that service.

- **desiredStatus** (*string*) – The task status that you want to filter the `ListTasks` results with. Specifying a `desiredStatus` of `STOPPED` will limit the results to tasks that are in the `STOPPED` status, which can be useful for debugging tasks that are not starting properly or have died or finished. The default status filter is `RUNNING`.

Return type dict

Returns

Response Syntax

```
{
    'taskArns': [
        'string',
    ],
    'nextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **taskArns** (*list*) –
The list of task Amazon Resource Name (ARN) entries for the `ListTasks` request.
 - * (*string*) –
 - **nextToken** (*string*) –
The `nextToken` value to include in a future `ListTasks` request. When the results of a `ListTasks` request exceed `maxResults`, this value can be used to retrieve the next page of results. This value is `null` when there are no more results to return.

register_container_instance (**kwargs)

Note: This action is only used by the Amazon EC2 Container Service agent, and it is not intended for use outside of the agent.

Registers an Amazon EC2 instance into the specified cluster. This instance will become available to place containers on.

Request Syntax

```
response = client.register_container_instance(
    cluster='string',
    instanceIdentityDocument='string',
    instanceIdentityDocumentSignature='string',
    totalResources=[
        {
            'name': 'string',
            'type': 'string',
            'doubleValue': 123.0,
            'longValue': 123,
            'integerValue': 123,
            'stringValue': [
                'string',
            ]
        },
    ],
)
```

```

versionInfo={
    'agentVersion': 'string',
    'agentHash': 'string',
    'dockerVersion': 'string'
},
containerInstanceArn='string'
)

```

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that you want to register your container instance with. If you do not specify a cluster, the default cluster is assumed..
- **instanceIdentityDocument** (*string*) – The instance identity document for the Amazon EC2 instance to register. This document can be found by running the following command from the instance: `curl http://169.254.169.254/latest/dynamic/instance-identity/document/`
- **instanceIdentityDocumentSignature** (*string*) – The instance identity document signature for the Amazon EC2 instance to register. This signature can be found by running the following command from the instance: `curl http://169.254.169.254/latest/dynamic/instance-identity/signature/`
- **totalResources** (*list*) – The resources available on the instance.
 - (*dict*) –

Describes the resources available for a container instance.

* **name** (*string*) –

The name of the resource, such as CPU , MEMORY , PORTS , or a user-defined resource.

* **type** (*string*) –

The type of the resource, such as INTEGER , DOUBLE , LONG , or STRINGSET .

* **doubleValue** (*float*) –

When the `doubleValue` type is set, the value of the resource must be a double precision floating-point type.

* **longValue** (*integer*) –

When the `longValue` type is set, the value of the resource must be an extended precision floating-point type.

* **integerValue** (*integer*) –

When the `integerValue` type is set, the value of the resource must be an integer.

* **stringSetValue** (*list*) –

When the `stringSetValue` type is set, the value of the resource must be a string type.

· (*string*) –

- **versionInfo** (*dict*) – The version information for the Amazon ECS container agent and Docker daemon running on the container instance.
 - **agentVersion** (*string*) –

The version number of the Amazon ECS container agent.
 - **agentHash** (*string*) –

The Git commit hash for the Amazon ECS container agent build on the [amazon-ecs-agent](#) GitHub repository.

- **dockerVersion** (*string*) –

The Docker version running on the container instance.

- **containerInstanceArn** (*string*) – The Amazon Resource Name (ARN) of the container instance (if it was previously registered).

Return type dict

Returns

Response Syntax

```
{
  'containerInstance': {
    'containerInstanceArn': 'string',
    'ec2InstanceId': 'string',
    'versionInfo': {
      'agentVersion': 'string',
      'agentHash': 'string',
      'dockerVersion': 'string'
    },
    'remainingResources': [
      {
        'name': 'string',
        'type': 'string',
        'doubleValue': 123.0,
        'longValue': 123,
        'integerValue': 123,
        'stringSetValue': [
          'string',
        ]
      },
    ],
    'registeredResources': [
      {
        'name': 'string',
        'type': 'string',
        'doubleValue': 123.0,
        'longValue': 123,
        'integerValue': 123,
        'stringSetValue': [
          'string',
        ]
      },
    ],
    'status': 'string',
    'agentConnected': True|False,
    'runningTasksCount': 123,
    'pendingTasksCount': 123,
    'agentUpdateStatus': 'PENDING' | 'STAGING' | 'STAGED' | 'UPDATING' | 'UPDATED' | 'FA
  }
}
```

Response Structure

- (*dict*) –
 - **containerInstance** (*dict*) –

An Amazon EC2 instance that is running the Amazon ECS agent and has been registered with a cluster.

- * **containerInstanceArn** (*string*) –

The Amazon Resource Name (ARN) of the container instance. The ARN contains the `arn:aws:ecs` namespace, followed by the region of the container instance, the AWS account ID of the container instance owner, the `container-instance` namespace, and then the container instance UUID. For example, `arn:aws:ecs:region:aws_account_id:container-instance/container_instance_UUID`.

* **ec2InstanceId** (*string*) –

The Amazon EC2 instance ID of the container instance.

* **versionInfo** (*dict*) –

The version information for the Amazon ECS container agent and Docker daemon running on the container instance.

· **agentVersion** (*string*) –

The version number of the Amazon ECS container agent.

· **agentHash** (*string*) –

The Git commit hash for the Amazon ECS container agent build on the [amazon-ecs-agent](#) GitHub repository.

· **dockerVersion** (*string*) –

The Docker version running on the container instance.

* **remainingResources** (*list*) –

The remaining resources of the container instance that are available for new tasks.

· (*dict*) –

Describes the resources available for a container instance.

· **name** (*string*) –

The name of the resource, such as `CPU`, `MEMORY`, `PORTS`, or a user-defined resource.

· **type** (*string*) –

The type of the resource, such as `INTEGER`, `DOUBLE`, `LONG`, or `STRINGSET`.

· **doubleValue** (*float*) –

When the `doubleValue` type is set, the value of the resource must be a double precision floating-point type.

· **longValue** (*integer*) –

When the `longValue` type is set, the value of the resource must be an extended precision floating-point type.

· **integerValue** (*integer*) –

When the `integerValue` type is set, the value of the resource must be an integer.

· **stringSetValue** (*list*) –

When the `stringSetValue` type is set, the value of the resource must be a string type.

· (*string*) –

* **registeredResources** (*list*) –

The registered resources on the container instance that are in use by current tasks.

· (*dict*) –

Describes the resources available for a container instance.

- **name** (*string*) –

The name of the resource, such as CPU , MEMORY , PORTS , or a user-defined resource.

- **type** (*string*) –

The type of the resource, such as INTEGER , DOUBLE , LONG , or STRINGSET .

- **doubleValue** (*float*) –

When the doubleValue type is set, the value of the resource must be a double precision floating-point type.

- **longValue** (*integer*) –

When the longValue type is set, the value of the resource must be an extended precision floating-point type.

- **integerValue** (*integer*) –

When the integerValue type is set, the value of the resource must be an integer.

- **stringSetValue** (*list*) –

When the stringSetValue type is set, the value of the resource must be a string type.

- (*string*) –

- * **status** (*string*) –

The status of the container instance. The valid values are ACTIVE or INACTIVE . ACTIVE indicates that the container instance can accept tasks.

- * **agentConnected** (*boolean*) –

This parameter returns `true` if the agent is actually connected to Amazon ECS. Registered instances with an agent that may be unhealthy or stopped will return `false` , and instances without a connected agent cannot accept placement request.

- * **runningTasksCount** (*integer*) –

The number of tasks on the container instance that are in the RUNNING status.

- * **pendingTasksCount** (*integer*) –

The number of tasks on the container instance that are in the PENDING status.

- * **agentUpdateStatus** (*string*) –

The status of the most recent agent update. If an update has never been requested, this value is NULL .

register_task_definition (***kwargs*)

Registers a new task definition from the supplied `family` and `containerDefinitions` . Optionally, you can add data volumes to your containers with the `volumes` parameter. For more information on task definition parameters and defaults, see [Amazon ECS Task Definitions](#) in the *Amazon EC2 Container Service Developer Guide* .

Request Syntax

```
response = client.register_task_definition(  
    family='string',
```



```

containerDefinitions=[
    {
        'name': 'string',
        'image': 'string',
        'cpu': 123,
        'memory': 123,
        'links': [
            'string',
        ],
        'portMappings': [
            {
                'containerPort': 123,
                'hostPort': 123,
                'protocol': 'tcp'|'udp'
            },
        ],
        'essential': True|False,
        'entryPoint': [
            'string',
        ],
        'command': [
            'string',
        ],
        'environment': [
            {
                'name': 'string',
                'value': 'string'
            },
        ],
        'mountPoints': [
            {
                'sourceVolume': 'string',
                'containerPath': 'string',
                'readOnly': True|False
            },
        ],
        'volumesFrom': [
            {
                'sourceContainer': 'string',
                'readOnly': True|False
            },
        ],
    },
],
volumes=[
    {
        'name': 'string',
        'host': {
            'sourcePath': 'string'
        }
    },
]
)

```

Parameters

- **family** (*string*) – [REQUIRED]

You must specify a `family` for a task definition, which allows you to track multiple versions of the same task definition. You can think of the `family` as

a name for your task definition. Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.

- **containerDefinitions** (*list*) – [REQUIRED]

A list of container definitions in JSON format that describe the different containers that make up your task.

- (*dict*) –

Container definitions are used in task definitions to describe the different containers that are launched as part of a task.

- * **name** (*string*) –

The name of a container. If you are linking multiple containers together in a task definition, the `name` of one container can be entered in the `links` of another container to connect the containers.

- * **image** (*string*) –

The image used to start a container. This string is passed directly to the Docker daemon. Images in the Docker Hub registry are available by default. Other repositories are specified with `*repository-url* /*image* :*tag*`.

- * **cpu** (*integer*) –

The number of `cpu` units reserved for the container. A container instance has 1,024 `cpu` units for every CPU core. This parameter specifies the minimum amount of CPU to reserve for a container, and containers share unallocated CPU units with other containers on the instance with the same ratio as their allocated amount.

For example, if you run a single-container task on a single-core instance type with 512 CPU units specified for that container, and that is the only task running on the container instance, that container could use the full 1,024 CPU unit share at any given time. However, if you launched another copy of the same task on that container instance, each task would be guaranteed a minimum of 512 CPU units when needed, and each container could float to higher CPU usage if the other container was not using it, but if both tasks were 100% active all of the time, they would be limited to 512 CPU units.

The Docker daemon on the container instance uses the CPU value to calculate the relative CPU share ratios for running containers. For more information, see [CPU share constraint](#) in the Docker documentation. The minimum valid CPU share value that the Linux kernel will allow is 2; however, the CPU parameter is not required, and you can use CPU values below 2 in your container definitions. For CPU values below 2 (including null), the behavior varies based on your Amazon ECS container agent version:

- **Agent versions less than or equal to 1.1.0:** Null and zero CPU values are passed to Docker as 0, which Docker then converts to 1,024 CPU shares. CPU values of 1 are passed to Docker as 1, which the Linux kernel converts to 2 CPU shares.
- **Agent versions greater than or equal to 1.2.0:** Null, zero, and CPU values of 1 are passed to Docker as 2.

- * **memory** (*integer*) –

The number of MiB of memory reserved for the container. If your container attempts to exceed the memory allocated here, the con-

tainer is killed.

* **links** (*list*) –

The `link` parameter allows containers to communicate with each other without the need for port mappings, using the `name` parameter. The `name:internalName` construct is analogous to `name:alias` in Docker links. For more information on linking Docker containers, see <https://docs.docker.com/userguide/dockerlinks/>.

Warning: Containers that are colocated on a single container instance may be able to communicate with each other without requiring links or host port mappings. Network isolation is achieved on the container instance using security groups and VPC settings.

· (*string*) –

* **portMappings** (*list*) –

The list of port mappings for the container.

· (*dict*) –

Port mappings allow containers to access ports on the host container instance to send or receive traffic. Port mappings are specified as part of the container definition.

· **containerPort** (*integer*) –

The port number on the container that is bound to the user-specified or automatically assigned host port. If you specify a container port and not a host port, your container will automatically receive a host port in the ephemeral port range (for more information, see `hostPort`).

· **hostPort** (*integer*) –

The port number on the container instance to reserve for your container. You can specify a non-reserved host port for your container port mapping, or you can omit the `hostPort` (or set it to 0) while specifying a `containerPort` and your container will automatically receive a port in the ephemeral port range for your container instance operating system and Docker version.

The default ephemeral port range is 49153 to 65535, and this range is used for Docker versions prior to 1.6.0. For Docker version 1.6.0 and later, the Docker daemon tries to read the ephemeral port range from `/proc/sys/net/ipv4/ip_local_port_range`; if this kernel parameter is unavailable, the default ephemeral port range is used. You should not attempt to specify a host port in the ephemeral port range, since these are reserved for automatic assignment. In general, ports below 32768 are outside of the ephemeral port range.

The default reserved ports are 22 for SSH, the Docker ports 2375 and 2376, and the Amazon ECS Container Agent port 51678. Any host port that was previously specified in a running task is also reserved while the task is running (once a task stops, the host port is released). The current reserved ports are

displayed in the `remainingResources` of `DescribeContainerInstances` output, and a container instance may have up to 50 reserved ports at a time, including the default reserved ports (automatically assigned ports do not count toward this limit).

- **protocol** (*string*) –

The protocol used for the port mapping. Valid values are `tcp` and `udp`. The default is `tcp`.

- * **essential** (*boolean*) –

If the `essential` parameter of a container is marked as `true`, the failure of that container will stop the task. If the `essential` parameter of a container is marked as `false`, then its failure will not affect the rest of the containers in a task. If this parameter is omitted, a container is assumed to be essential.

Note: All tasks must have at least one essential container.

- * **entryPoint** (*list*) –

Warning: Early versions of the Amazon ECS container agent do not properly handle `entryPoint` parameters. If you have problems using `entryPoint`, update your container agent or enter your commands and arguments as `command` array items instead.

The `ENTRYPOINT` that is passed to the container. For more information on the Docker `ENTRYPOINT` parameter, see <https://docs.docker.com/reference/builder/#entrypoint>.

- (*string*) –

- * **command** (*list*) –

The `CMD` that is passed to the container. For more information on the Docker `CMD` parameter, see <https://docs.docker.com/reference/builder/#cmd>.

- (*string*) –

- * **environment** (*list*) –

The environment variables to pass to a container.

- (*dict*) –

A key and value pair object.

- **name** (*string*) –

The name of the key value pair. For environment variables, this is the name of the environment variable.

- **value** (*string*) –

The value of the key value pair. For environment variables, this is the value of the environment variable.

- * **mountPoints** (*list*) –

The mount points for data volumes in your container.

- (*dict*) –

Details on a volume mount point that is used in a container definition.

- **sourceVolume** (*string*) –
The name of the volume to mount.
- **containerPath** (*string*) –
The path on the container to mount the host volume at.
- **readOnly** (*boolean*) –
If this value is `true`, the container has read-only access to the volume. If this value is `false`, then the container can write to the volume. The default value is `false`.
- * **volumesFrom** (*list*) –
Data volumes to mount from another container.
 - (*dict*) –
Details on a data volume from another container.
 - **sourceContainer** (*string*) –
The name of the container to mount volumes from.
 - **readOnly** (*boolean*) –
If this value is `true`, the container has read-only access to the volume. If this value is `false`, then the container can write to the volume. The default value is `false`.
- **volumes** (*list*) – A list of volume definitions in JSON format that containers in your task may use.
 - (*dict*) –
A data volume used in a task definition.
 - * **name** (*string*) –
The name of the volume. This name is referenced in the `sourceVolume` parameter of container definition `mountPoints`.
 - * **host** (*dict*) –
The path on the host container instance that is presented to the containers which access the volume. If this parameter is empty, then the Docker daemon assigns a host path for you.
 - **sourcePath** (*string*) –
The path on the host container instance that is presented to the container. If this parameter is empty, then the Docker daemon has assigned a host path for you.

Return type dict

Returns

Response Syntax

```
{
  'taskDefinition': {
    'taskDefinitionArn': 'string',
    'containerDefinitions': [
      {
        'name': 'string',
        'image': 'string',
        'cpu': 123,
        'memory': 123,
        'links': [
```

```
        'string',
    ],
    'portMappings': [
        {
            'containerPort': 123,
            'hostPort': 123,
            'protocol': 'tcp'|'udp'
        },
    ],
    'essential': True|False,
    'entryPoint': [
        'string',
    ],
    'command': [
        'string',
    ],
    'environment': [
        {
            'name': 'string',
            'value': 'string'
        },
    ],
    'mountPoints': [
        {
            'sourceVolume': 'string',
            'containerPath': 'string',
            'readOnly': True|False
        },
    ],
    'volumesFrom': [
        {
            'sourceContainer': 'string',
            'readOnly': True|False
        },
    ],
    ],
    },
    'family': 'string',
    'revision': 123,
    'volumes': [
        {
            'name': 'string',
            'host': {
                'sourcePath': 'string'
            }
        },
    ],
    'status': 'ACTIVE'|'INACTIVE'
}
}
```

Response Structure

- (*dict*) –

- **taskDefinition** (*dict*) –

Details of a task definition.

- * **taskDefinitionArn** (*string*) –

The full Amazon Resource Name (ARN) of the of the task defini-

tion.

* **containerDefinitions** (*list*) –

A list of container definitions in JSON format that describe the different containers that make up your task. For more information on container definition parameters and defaults, see [Amazon ECS Task Definitions](#) in the *Amazon EC2 Container Service Developer Guide*

· (*dict*) –

Container definitions are used in task definitions to describe the different containers that are launched as part of a task.

· **name** (*string*) –

The name of a container. If you are linking multiple containers together in a task definition, the `name` of one container can be entered in the `links` of another container to connect the containers.

· **image** (*string*) –

The image used to start a container. This string is passed directly to the Docker daemon. Images in the Docker Hub registry are available by default. Other repositories are specified with `*repository-url* /*image* :*tag*`.

· **cpu** (*integer*) –

The number of `cpu` units reserved for the container. A container instance has 1,024 `cpu` units for every CPU core. This parameter specifies the minimum amount of CPU to reserve for a container, and containers share unallocated CPU units with other containers on the instance with the same ratio as their allocated amount.

For example, if you run a single-container task on a single-core instance type with 512 CPU units specified for that container, and that is the only task running on the container instance, that container could use the full 1,024 CPU unit share at any given time. However, if you launched another copy of the same task on that container instance, each task would be guaranteed a minimum of 512 CPU units when needed, and each container could float to higher CPU usage if the other container was not using it, but if both tasks were 100% active all of the time, they would be limited to 512 CPU units.

The Docker daemon on the container instance uses the CPU value to calculate the relative CPU share ratios for running containers. For more information, see [CPU share constraint](#) in the Docker documentation. The minimum valid CPU share value that the Linux kernel will allow is 2; however, the CPU parameter is not required, and you can use CPU values below 2 in your container definitions. For CPU values below 2 (including null), the behavior varies based on your Amazon ECS container agent version:

- **Agent versions less than or equal to 1.1.0:** Null and zero CPU values are passed to Docker as 0, which Docker then converts to 1,024 CPU shares. CPU values of 1 are passed to Docker as 1, which the Linux kernel converts to 2 CPU

shares.

- **Agent versions greater than or equal to 1.2.0:** Null, zero, and CPU values of 1 are passed to Docker as 2.
- **memory** (*integer*) –

The number of MiB of memory reserved for the container. If your container attempts to exceed the memory allocated here, the container is killed.

- **links** (*list*) –

The `link` parameter allows containers to communicate with each other without the need for port mappings, using the `name` parameter. The `name:internalName` construct is analogous to `name:alias` in Docker links. For more information on linking Docker containers, see <https://docs.docker.com/userguide/dockerlinks/>.

Warning: Containers that are collocated on a single container instance may be able to communicate with each other without requiring links or host port mappings. Network isolation is achieved on the container instance using security groups and VPC settings.

- (*string*) –
- **portMappings** (*list*) –

The list of port mappings for the container.

- (*dict*) –

Port mappings allow containers to access ports on the host container instance to send or receive traffic. Port mappings are specified as part of the container definition.

- **containerPort** (*integer*) –

The port number on the container that is bound to the user-specified or automatically assigned host port. If you specify a container port and not a host port, your container will automatically receive a host port in the ephemeral port range (for more information, see `hostPort`).

- **hostPort** (*integer*) –

The port number on the container instance to reserve for your container. You can specify a non-reserved host port for your container port mapping, or you can omit the `hostPort` (or set it to 0) while specifying a `containerPort` and your container will automatically receive a port in the ephemeral port range for your container instance operating system and Docker version.

The default ephemeral port range is 49153 to 65535, and this range is used for Docker versions prior to 1.6.0. For Docker version 1.6.0 and later, the Docker daemon tries to read the ephemeral port range from `/proc/sys/net/ipv4/ip_local_port_range`; if this kernel parameter is unavailable, the default ephemeral port range is used. You should not attempt to specify a host port in the ephemeral port range, since these are reserved for

automatic assignment. In general, ports below 32768 are outside of the ephemeral port range.

The default reserved ports are 22 for SSH, the Docker ports 2375 and 2376, and the Amazon ECS Container Agent port 51678. Any host port that was previously specified in a running task is also reserved while the task is running (once a task stops, the host port is released). The current reserved ports are displayed in the `remainingResources` of `DescribeContainerInstances` output, and a container instance may have up to 50 reserved ports at a time, including the default reserved ports (automatically assigned ports do not count toward this limit).

- **protocol** (*string*) –

The protocol used for the port mapping. Valid values are `tcp` and `udp`. The default is `tcp`.

- **essential** (*boolean*) –

If the `essential` parameter of a container is marked as `true`, the failure of that container will stop the task. If the `essential` parameter of a container is marked as `false`, then its failure will not affect the rest of the containers in a task. If this parameter is omitted, a container is assumed to be essential.

Note: All tasks must have at least one essential container.

- **entryPoint** (*list*) –

Warning: Early versions of the Amazon ECS container agent do not properly handle `entryPoint` parameters. If you have problems using `entryPoint`, update your container agent or enter your commands and arguments as `command` array items instead.

The `ENTRYPOINT` that is passed to the container. For more information on the Docker `ENTRYPOINT` parameter, see <https://docs.docker.com/reference/builder/#entrypoint>.

- (*string*) –
- **command** (*list*) –

The `CMD` that is passed to the container. For more information on the Docker `CMD` parameter, see <https://docs.docker.com/reference/builder/#cmd>.

- (*string*) –
- **environment** (*list*) –

The environment variables to pass to a container.

- (*dict*) –

A key and value pair object.

- **name** (*string*) –

The name of the key value pair. For environment variables, this is the name of the environment variable.

- **value** (*string*) –

The value of the key value pair. For environment variables, this is the value of the environment variable.

- **mountPoints** (*list*) –

The mount points for data volumes in your container.

- (*dict*) –

Details on a volume mount point that is used in a container definition.

- **sourceVolume** (*string*) –

The name of the volume to mount.

- **containerPath** (*string*) –

The path on the container to mount the host volume at.

- **readOnly** (*boolean*) –

If this value is `true`, the container has read-only access to the volume. If this value is `false`, then the container can write to the volume. The default value is `false`.

- **volumesFrom** (*list*) –

Data volumes to mount from another container.

- (*dict*) –

Details on a data volume from another container.

- **sourceContainer** (*string*) –

The name of the container to mount volumes from.

- **readOnly** (*boolean*) –

If this value is `true`, the container has read-only access to the volume. If this value is `false`, then the container can write to the volume. The default value is `false`.

- * **family** (*string*) –

The family of your task definition. You can think of the `family` as the name of your task definition.

- * **revision** (*integer*) –

The revision of the task in a particular family. You can think of the revision as a version number of a task definition in a family. When you register a task definition for the first time, the revision is `1`, and each time you register a new revision of a task definition in the same family, the revision value always increases by one (even if you have deregistered previous revisions in this family).

- * **volumes** (*list*) –

The list of volumes in a task. For more information on volume definition parameters and defaults, see [Amazon ECS Task Definitions](#) in the *Amazon EC2 Container Service Developer Guide*.

- (*dict*) –

A data volume used in a task definition.

- **name** (*string*) –

The name of the volume. This name is referenced in the `sourceVolume` parameter of container definition `mountPoints`.

- **host** (*dict*) –

The path on the host container instance that is presented to the containers which access the volume. If this parameter is empty, then the Docker daemon assigns a host path for you.

- **sourcePath** (*string*) –

The path on the host container instance that is presented to the container. If this parameter is empty, then the Docker daemon has assigned a host path for you.

- * **status** (*string*) –

The status of the task definition.

run_task (***kwargs*)

Start a task using random placement and the default Amazon ECS scheduler. If you want to use your own scheduler or place a task on a specific container instance, use `StartTask` instead.

Warning: The `count` parameter is limited to 10 tasks per call.

Request Syntax

```
response = client.run_task(
    cluster='string',
    taskDefinition='string',
    overrides={
        'containerOverrides': [
            {
                'name': 'string',
                'command': [
                    'string',
                ],
                'environment': [
                    {
                        'name': 'string',
                        'value': 'string'
                    },
                ],
            },
        ],
    },
    count=123,
    startedBy='string'
)
```

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that you want to run your task on. If you do not specify a cluster, the default cluster is assumed..
- **taskDefinition** (*string*) – **[REQUIRED]**

The family and revision (`family:revision`) or full Amazon Resource Name (ARN) of the task definition that you want to run. If a revision is not specified, the latest ACTIVE revision is used.

- **overrides** (*dict*) – A list of container overrides in JSON format that specify the name of a container in the specified task definition and the overrides it should receive. You can override the default command for a container (that is specified in the task definition or Docker image) with a `command` override. You can also override existing environment variables (that are specified in the task definition

or Docker image) on a container or add new environment variables to it with an `environment override`.

Note: A total of 8192 characters are allowed for overrides. This limit includes the JSON formatting characters of the override structure.

– **containerOverrides** (*list*) –

One or more container overrides sent to a task.

* (*dict*) –

The overrides that should be sent to a container.

· **name** (*string*) –

The name of the container that receives the override.

· **command** (*list*) –

The command to send to the container that overrides the default command from the Docker image or the task definition.

· (*string*) –

· **environment** (*list*) –

The environment variables to send to the container. You can add new environment variables, which are added to the container at launch, or you can override the existing environment variables from the Docker image or the task definition.

· (*dict*) –

A key and value pair object.

· **name** (*string*) –

The name of the key value pair. For environment variables, this is the name of the environment variable.

· **value** (*string*) –

The value of the key value pair. For environment variables, this is the value of the environment variable.

- **count** (*integer*) – The number of instantiations of the specified task that you would like to place on your cluster.

Warning: The `count` parameter is limited to 10 tasks per call.

- **startedBy** (*string*) – An optional tag specified when a task is started. For example if you automatically trigger a task to run a batch process job, you could apply a unique identifier for that job to your task with the `startedBy` parameter. You can then identify which tasks belong to that job by filtering the results of a `ListTasks` call with the `startedBy` value.

If a task is started by an Amazon ECS service, then the `startedBy` parameter contains the deployment ID of the service that starts it.

Return type `dict`

Returns

Response Syntax

```
{
  'tasks': [
    {
      'taskArn': 'string',
```

```

'clusterArn': 'string',
'taskDefinitionArn': 'string',
'containerInstanceArn': 'string',
'overrides': {
    'containerOverrides': [
        {
            'name': 'string',
            'command': [
                'string',
            ],
            'environment': [
                {
                    'name': 'string',
                    'value': 'string'
                },
            ]
        },
    ]
},
'lastStatus': 'string',
'desiredStatus': 'string',
'containers': [
    {
        'containerArn': 'string',
        'taskArn': 'string',
        'name': 'string',
        'lastStatus': 'string',
        'exitCode': 123,
        'reason': 'string',
        'networkBindings': [
            {
                'bindIP': 'string',
                'containerPort': 123,
                'hostPort': 123,
                'protocol': 'tcp'|'udp'
            },
        ],
    },
],
'startedBy': 'string'
},
'failures': [
    {
        'arn': 'string',
        'reason': 'string'
    },
]
}

```

Response Structure

- *(dict)* –
- **tasks** (*list*) –

A full description of the tasks that were run. Each task that was successfully placed on your cluster will be described here.

- * *(dict)* –

Details on a task in a cluster.

- **taskArn** (*string*) –
The Amazon Resource Name (ARN) of the task.
- **clusterArn** (*string*) –
The Amazon Resource Name (ARN) of the of the cluster that hosts the task.
- **taskDefinitionArn** (*string*) –
The Amazon Resource Name (ARN) of the of the task definition that creates the task.
- **containerInstanceArn** (*string*) –
The Amazon Resource Name (ARN) of the container instances that host the task.
- **overrides** (*dict*) –
One or more container overrides.
- **containerOverrides** (*list*) –
One or more container overrides sent to a task.
(*dict*) –
The overrides that should be sent to a container.
- **name** (*string*) –
The name of the container that receives the override.
- **command** (*list*) –
The command to send to the container that overrides the default command from the Docker image or the task definition.
(*string*) –
- **environment** (*list*) –
The environment variables to send to the container. You can add new environment variables, which are added to the container at launch, or you can override the existing environment variables from the Docker image or the task definition.
(*dict*) –
A key and value pair object.
- **name** (*string*) –
The name of the key value pair. For environment variables, this is the name of the environment variable.
- **value** (*string*) –
The value of the key value pair. For environment variables, this is the value of the environment variable.
- **lastStatus** (*string*) –
The last known status of the task.
- **desiredStatus** (*string*) –
The desired status of the task.
- **containers** (*list*) –
The containers associated with the task.
(*dict*) –
A docker container that is part of a task.

- **containerArn** (*string*) –
The Amazon Resource Name (ARN) of the container.
- **taskArn** (*string*) –
The Amazon Resource Name (ARN) of the task.
- **name** (*string*) –
The name of the container.
- **lastStatus** (*string*) –
The last known status of the container.
- **exitCode** (*integer*) –
The exit code returned from the container.
- **reason** (*string*) –
A short (255 max characters) human-readable string to provide additional detail about a running or stopped container.
- **networkBindings** (*list*) –
The network bindings associated with the container.
- (*dict*) –
Details on the network bindings between a container and its host container instance.
- **bindIP** (*string*) –
The IP address that the container is bound to on the container instance.
- **containerPort** (*integer*) –
The port number on the container that is be used with the network binding.
- **hostPort** (*integer*) –
The port number on the host that is used with the network binding.
- **protocol** (*string*) –
The protocol used for the network binding.
- **startedBy** (*string*) –
The tag specified when a task is started. If the task is started by an Amazon ECS service, then the `startedBy` parameter contains the deployment ID of the service that starts it.
- **failures** (*list*) –
Any failed tasks from your `RunTask` action are listed here.
- (*dict*) –
A failed resource.
- **arn** (*string*) –
The Amazon Resource Name (ARN) of the failed resource.
- **reason** (*string*) –
The reason for the failure.

start_task (***kwargs*)

Starts a new task from the specified task definition on the specified container instance or instances. If you want to use the default Amazon ECS scheduler to place your task, use `RunTask` instead.

Warning: The list of container instances to start tasks on is limited to 10.

Request Syntax

```
response = client.start_task(
    cluster='string',
    taskDefinition='string',
    overrides={
        'containerOverrides': [
            {
                'name': 'string',
                'command': [
                    'string',
                ],
                'environment': [
                    {
                        'name': 'string',
                        'value': 'string'
                    },
                ]
            },
        ]
    },
    containerInstances=[
        'string',
    ],
    startedBy='string'
)
```

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that you want to start your task on. If you do not specify a cluster, the default cluster is assumed..
- **taskDefinition** (*string*) – **[REQUIRED]**

The family and revision (`family:revision`) or full Amazon Resource Name (ARN) of the task definition that you want to start. If a revision is not specified, the latest ACTIVE revision is used.

- **overrides** (*dict*) – A list of container overrides in JSON format that specify the name of a container in the specified task definition and the overrides it should receive. You can override the default command for a container (that is specified in the task definition or Docker image) with a `command` override. You can also override existing environment variables (that are specified in the task definition or Docker image) on a container or add new environment variables to it with an `environment` override.

Note: A total of 8192 characters are allowed for overrides. This limit includes the JSON formatting characters of the override structure.

– containerOverrides (*list*) –

One or more container overrides sent to a task.

* (*dict*) –

The overrides that should be sent to a container.

· name (*string*) –

The name of the container that receives the override.

- **command** (*list*) –

The command to send to the container that overrides the default command from the Docker image or the task definition.

- (*string*) –

- **environment** (*list*) –

The environment variables to send to the container. You can add new environment variables, which are added to the container at launch, or you can override the existing environment variables from the Docker image or the task definition.

- (*dict*) –

A key and value pair object.

- **name** (*string*) –

The name of the key value pair. For environment variables, this is the name of the environment variable.

- **value** (*string*) –

The value of the key value pair. For environment variables, this is the value of the environment variable.

- **containerInstances** (*list*) – **[REQUIRED]**

The container instance UUIDs or full Amazon Resource Name (ARN) entries for the container instances on which you would like to place your task.

Warning: The list of container instances to start tasks on is limited to 10.

- (*string*) –

- **startedBy** (*string*) – An optional tag specified when a task is started. For example if you automatically trigger a task to run a batch process job, you could apply a unique identifier for that job to your task with the `startedBy` parameter. You can then identify which tasks belong to that job by filtering the results of a `ListTasks` call with the `startedBy` value.

If a task is started by an Amazon ECS service, then the `startedBy` parameter contains the deployment ID of the service that starts it.

Return type dict

Returns

Response Syntax

```
{
  'tasks': [
    {
      'taskArn': 'string',
      'clusterArn': 'string',
      'taskDefinitionArn': 'string',
      'containerInstanceArn': 'string',
      'overrides': {
        'containerOverrides': [
          {
            'name': 'string',
            'command': [
              'string',
            ],
            'environment': [
              {

```

```
        'name': 'string',
        'value': 'string'
    },
    ],
    },
    ],
    'lastStatus': 'string',
    'desiredStatus': 'string',
    'containers': [
        {
            'containerArn': 'string',
            'taskArn': 'string',
            'name': 'string',
            'lastStatus': 'string',
            'exitCode': 123,
            'reason': 'string',
            'networkBindings': [
                {
                    'bindIP': 'string',
                    'containerPort': 123,
                    'hostPort': 123,
                    'protocol': 'tcp'|'udp'
                },
            ],
        },
    ],
    'startedBy': 'string'
},
],
'failures': [
    {
        'arn': 'string',
        'reason': 'string'
    },
],
]
```

Response Structure

- (*dict*) –
 - **tasks** (*list*) –

A full description of the tasks that were started. Each task that was successfully placed on your container instances will be described here.

* (*dict*) –

Details on a task in a cluster.

- **taskArn** (*string*) –
The Amazon Resource Name (ARN) of the task.
- **clusterArn** (*string*) –
The Amazon Resource Name (ARN) of the of the cluster that hosts the task.
- **taskDefinitionArn** (*string*) –
The Amazon Resource Name (ARN) of the of the task definition that creates the task.
- **containerInstanceArn** (*string*) –

The Amazon Resource Name (ARN) of the container instances that host the task.

- **overrides** (*dict*) –

One or more container overrides.

- **containerOverrides** (*list*) –

One or more container overrides sent to a task.

- (*dict*) –

The overrides that should be sent to a container.

- **name** (*string*) –

The name of the container that receives the override.

- **command** (*list*) –

The command to send to the container that overrides the default command from the Docker image or the task definition.

- (*string*) –

- **environment** (*list*) –

The environment variables to send to the container. You can add new environment variables, which are added to the container at launch, or you can override the existing environment variables from the Docker image or the task definition.

- (*dict*) –

A key and value pair object.

- **name** (*string*) –

The name of the key value pair. For environment variables, this is the name of the environment variable.

- **value** (*string*) –

The value of the key value pair. For environment variables, this is the value of the environment variable.

- **lastStatus** (*string*) –

The last known status of the task.

- **desiredStatus** (*string*) –

The desired status of the task.

- **containers** (*list*) –

The containers associated with the task.

- (*dict*) –

A docker container that is part of a task.

- **containerArn** (*string*) –

The Amazon Resource Name (ARN) of the container.

- **taskArn** (*string*) –

The Amazon Resource Name (ARN) of the task.

- **name** (*string*) –

The name of the container.

- **lastStatus** (*string*) –

The last known status of the container.

- **exitCode** (*integer*) –

The exit code returned from the container.

- **reason** (*string*) –
A short (255 max characters) human-readable string to provide additional detail about a running or stopped container.
- **networkBindings** (*list*) –
The network bindings associated with the container.
- (*dict*) –
Details on the network bindings between a container and its host container instance.
- **bindIP** (*string*) –
The IP address that the container is bound to on the container instance.
- **containerPort** (*integer*) –
The port number on the container that is be used with the network binding.
- **hostPort** (*integer*) –
The port number on the host that is used with the network binding.
- **protocol** (*string*) –
The protocol used for the network binding.
- **startedBy** (*string*) –
The tag specified when a task is started. If the task is started by an Amazon ECS service, then the `startedBy` parameter contains the deployment ID of the service that starts it.
- **failures** (*list*) –
Any failed tasks from your `StartTask` action are listed here.
 - * (*dict*) –
A failed resource.
 - **arn** (*string*) –
The Amazon Resource Name (ARN) of the failed resource.
 - **reason** (*string*) –
The reason for the failure.

stop_task (***kwargs*)
Stops a running task.

Request Syntax

```
response = client.stop_task(  
    cluster='string',  
    task='string'  
)
```

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that hosts the task you want to stop. If you do not specify a cluster, the default cluster is assumed..
- **task** (*string*) – **[REQUIRED]**
The task UUIDs or full Amazon Resource Name (ARN) entry of the task you would like to stop.

Return type dict

Returns

Response Syntax

```

{
    'task': {
        'taskArn': 'string',
        'clusterArn': 'string',
        'taskDefinitionArn': 'string',
        'containerInstanceArn': 'string',
        'overrides': {
            'containerOverrides': [
                {
                    'name': 'string',
                    'command': [
                        'string',
                    ],
                    'environment': [
                        {
                            'name': 'string',
                            'value': 'string'
                        },
                    ],
                },
            ],
        },
        'lastStatus': 'string',
        'desiredStatus': 'string',
        'containers': [
            {
                'containerArn': 'string',
                'taskArn': 'string',
                'name': 'string',
                'lastStatus': 'string',
                'exitCode': 123,
                'reason': 'string',
                'networkBindings': [
                    {
                        'bindIP': 'string',
                        'containerPort': 123,
                        'hostPort': 123,
                        'protocol': 'tcp' | 'udp'
                    },
                ],
            },
        ],
        'startedBy': 'string'
    }
}

```

Response Structure

• (*dict*) –

– **task** (*dict*) –

Details on a task in a cluster.

* **taskArn** (*string*) –

The Amazon Resource Name (ARN) of the task.

* **clusterArn** (*string*) –

The Amazon Resource Name (ARN) of the of the cluster that hosts the task.

* **taskDefinitionArn** (*string*) –

The Amazon Resource Name (ARN) of the of the task definition that creates the task.

* **containerInstanceArn** (*string*) –

The Amazon Resource Name (ARN) of the container instances that host the task.

* **overrides** (*dict*) –

One or more container overrides.

· **containerOverrides** (*list*) –

One or more container overrides sent to a task.

· (*dict*) –

The overrides that should be sent to a container.

· **name** (*string*) –

The name of the container that receives the override.

· **command** (*list*) –

The command to send to the container that overrides the default command from the Docker image or the task definition.

· (*string*) –

· **environment** (*list*) –

The environment variables to send to the container. You can add new environment variables, which are added to the container at launch, or you can override the existing environment variables from the Docker image or the task definition.

· (*dict*) –

A key and value pair object.

· **name** (*string*) –

The name of the key value pair. For environment variables, this is the name of the environment variable.

· **value** (*string*) –

The value of the key value pair. For environment variables, this is the value of the environment variable.

* **lastStatus** (*string*) –

The last known status of the task.

* **desiredStatus** (*string*) –

The desired status of the task.

* **containers** (*list*) –

The containers associated with the task.

· (*dict*) –

A docker container that is part of a task.

· **containerArn** (*string*) –

The Amazon Resource Name (ARN) of the container.

- **taskArn** (*string*) –
The Amazon Resource Name (ARN) of the task.
- **name** (*string*) –
The name of the container.
- **lastStatus** (*string*) –
The last known status of the container.
- **exitCode** (*integer*) –
The exit code returned from the container.
- **reason** (*string*) –
A short (255 max characters) human-readable string to provide additional detail about a running or stopped container.
- **networkBindings** (*list*) –
The network bindings associated with the container.
- (*dict*) –
Details on the network bindings between a container and its host container instance.
- **bindIP** (*string*) –
The IP address that the container is bound to on the container instance.
- **containerPort** (*integer*) –
The port number on the container that is be used with the network binding.
- **hostPort** (*integer*) –
The port number on the host that is used with the network binding.
- **protocol** (*string*) –
The protocol used for the network binding.
- * **startedBy** (*string*) –
The tag specified when a task is started. If the task is started by an Amazon ECS service, then the `startedBy` parameter contains the deployment ID of the service that starts it.

submit_container_state_change (**kwargs)

Note: This action is only used by the Amazon EC2 Container Service agent, and it is not intended for use outside of the agent.

Sent to acknowledge that a container changed states.

Request Syntax

```
response = client.submit_container_state_change(
    cluster='string',
    task='string',
    containerName='string',
    status='string',
    exitCode=123,
    reason='string',
```

```
networkBindings=[
    {
        'bindIP': 'string',
        'containerPort': 123,
        'hostPort': 123,
        'protocol': 'tcp' | 'udp'
    },
]
```

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that hosts the container.
- **task** (*string*) – The task UUID or full Amazon Resource Name (ARN) of the task that hosts the container.
- **containerName** (*string*) – The name of the container.
- **status** (*string*) – The status of the state change request.
- **exitCode** (*integer*) – The exit code returned for the state change request.
- **reason** (*string*) – The reason for the state change request.
- **networkBindings** (*list*) – The network bindings of the container.
 - (*dict*) –

Details on the network bindings between a container and its host container instance.

* **bindIP** (*string*) –

The IP address that the container is bound to on the container instance.

* **containerPort** (*integer*) –

The port number on the container that is be used with the network binding.

* **hostPort** (*integer*) –

The port number on the host that is used with the network binding.

* **protocol** (*string*) –

The protocol used for the network binding.

Return type dict

Returns

Response Syntax

```
{
    'acknowledgment': 'string'
}
```

Response Structure

- (*dict*) –
 - **acknowledgment** (*string*) –
Acknowledgement of the state change.

submit_task_state_change (***kwargs*)

Note: This action is only used by the Amazon EC2 Container Service agent, and it is not intended for use outside of the agent.

Sent to acknowledge that a task changed states.

Request Syntax

```
response = client.submit_task_state_change(
    cluster='string',
    task='string',
    status='string',
    reason='string'
)
```

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that hosts the task.
- **task** (*string*) – The task UUID or full Amazon Resource Name (ARN) of the task in the state change request.
- **status** (*string*) – The status of the state change request.
- **reason** (*string*) – The reason for the state change request.

Return type dict

Returns

Response Syntax

```
{
    'acknowledgment': 'string'
}
```

Response Structure

- (*dict*) –
 - **acknowledgment** (*string*) –
Acknowledgement of the state change.

update_container_agent (***kwargs*)

Updates the Amazon ECS container agent on a specified container instance. Updating the Amazon ECS container agent does not interrupt running tasks or services on the container instance. The process for updating the agent differs depending on whether your container instance was launched with the Amazon ECS-optimized AMI or another operating system.

UpdateContainerAgent requires the Amazon ECS-optimized AMI or Amazon Linux with the `ecs-init` service installed and running. For help updating the Amazon ECS container agent on other operating systems, see [Manually Updating the Amazon ECS Container Agent](#) in the *Amazon EC2 Container Service Developer Guide*.

Request Syntax

```
response = client.update_container_agent(
    cluster='string',
    containerInstance='string'
)
```

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that your container instance is running on. If you do not specify a cluster, the default cluster is assumed.
- **containerInstance** (*string*) – **[REQUIRED]**

The container instance UUID or full Amazon Resource Name (ARN) entries for the container instance on which you would like to update the Amazon ECS container agent.

Return type dict

Returns

Response Syntax

```
{
  'containerInstance': {
    'containerInstanceArn': 'string',
    'ec2InstanceId': 'string',
    'versionInfo': {
      'agentVersion': 'string',
      'agentHash': 'string',
      'dockerVersion': 'string'
    },
    'remainingResources': [
      {
        'name': 'string',
        'type': 'string',
        'doubleValue': 123.0,
        'longValue': 123,
        'integerValue': 123,
        'stringSetValue': [
          'string',
        ]
      }
    ],
    'registeredResources': [
      {
        'name': 'string',
        'type': 'string',
        'doubleValue': 123.0,
        'longValue': 123,
        'integerValue': 123,
        'stringSetValue': [
          'string',
        ]
      }
    ],
    'status': 'string',
    'agentConnected': True|False,
    'runningTasksCount': 123,
    'pendingTasksCount': 123,
    'agentUpdateStatus': 'PENDING' | 'STAGING' | 'STAGED' | 'UPDATING' | 'UPDATED' | 'FA
  }
}
```

Response Structure

- (*dict*) –

- **containerInstance** (*dict*) –

An Amazon EC2 instance that is running the Amazon ECS agent and has been registered with a cluster.

- * **containerInstanceArn** (*string*) –

The Amazon Resource Name (ARN) of the container instance. The ARN contains the `arn:aws:ecs` namespace, followed by the re-

gion of the container instance, the AWS account ID of the container instance owner, the `container-instance` namespace, and then the container instance UUID. For example, `arn:aws:ecs:region:aws_account_id:container-instance/container_instance_UUID`.

* **ec2InstanceId** (*string*) –

The Amazon EC2 instance ID of the container instance.

* **versionInfo** (*dict*) –

The version information for the Amazon ECS container agent and Docker daemon running on the container instance.

· **agentVersion** (*string*) –

The version number of the Amazon ECS container agent.

· **agentHash** (*string*) –

The Git commit hash for the Amazon ECS container agent build on the [amazon-ecs-agent](#) GitHub repository.

· **dockerVersion** (*string*) –

The Docker version running on the container instance.

* **remainingResources** (*list*) –

The remaining resources of the container instance that are available for new tasks.

· (*dict*) –

Describes the resources available for a container instance.

· **name** (*string*) –

The name of the resource, such as CPU , MEMORY , PORTS , or a user-defined resource.

· **type** (*string*) –

The type of the resource, such as INTEGER , DOUBLE , LONG , or STRINGSET .

· **doubleValue** (*float*) –

When the `doubleValue` type is set, the value of the resource must be a double precision floating-point type.

· **longValue** (*integer*) –

When the `longValue` type is set, the value of the resource must be an extended precision floating-point type.

· **integerValue** (*integer*) –

When the `integerValue` type is set, the value of the resource must be an integer.

· **stringSetValue** (*list*) –

When the `stringSetValue` type is set, the value of the resource must be a string type.

· (*string*) –

* **registeredResources** (*list*) –

The registered resources on the container instance that are in use by current tasks.

· (*dict*) –

Describes the resources available for a container instance.

- **name** (*string*) –
The name of the resource, such as CPU , MEMORY , PORTS , or a user-defined resource.
- **type** (*string*) –
The type of the resource, such as INTEGER , DOUBLE , LONG , or STRINGSET .
- **doubleValue** (*float*) –
When the doubleValue type is set, the value of the resource must be a double precision floating-point type.
- **longValue** (*integer*) –
When the longValue type is set, the value of the resource must be an extended precision floating-point type.
- **integerValue** (*integer*) –
When the integerValue type is set, the value of the resource must be an integer.
- **stringSetValue** (*list*) –
When the stringSetValue type is set, the value of the resource must be a string type.
- (*string*) –
- * **status** (*string*) –
The status of the container instance. The valid values are ACTIVE or INACTIVE . ACTIVE indicates that the container instance can accept tasks.
- * **agentConnected** (*boolean*) –
This parameter returns `true` if the agent is actually connected to Amazon ECS. Registered instances with an agent that may be unhealthy or stopped will return `false` , and instances without a connected agent cannot accept placement request.
- * **runningTasksCount** (*integer*) –
The number of tasks on the container instance that are in the RUNNING status.
- * **pendingTasksCount** (*integer*) –
The number of tasks on the container instance that are in the PENDING status.
- * **agentUpdateStatus** (*string*) –
The status of the most recent agent update. If an update has never been requested, this value is NULL .

update_service (***kwargs*)

Modify the desired count or task definition used in a service.

You can add to or subtract from the number of instantiations of a task definition in a service by specifying the cluster that the service is running in and a new `desiredCount` parameter.

You can use `UpdateService` to modify your task definition and deploy a new version of your service, one task at a time. If you modify the task definition with `UpdateService` , Amazon ECS spawns a task with the new version of the task definition and then stops an old task after the new version is running. Because `UpdateService` starts a new version of the task before stopping an old version, your cluster must have capacity to support one more instantiation of the task when `UpdateService` is run. If your

cluster cannot support another instantiation of the task used in your service, you can reduce the desired count of your service by one before modifying the task definition.

Request Syntax

```
response = client.update_service(
    cluster='string',
    service='string',
    desiredCount=123,
    taskDefinition='string'
)
```

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that your service is running on. If you do not specify a cluster, the default cluster is assumed.
- **service** (*string*) – **[REQUIRED]**
The name of the service that you want to update.
- **desiredCount** (*integer*) – The number of instantiations of the task that you would like to place and keep running in your service.
- **taskDefinition** (*string*) – The family and revision (family:revision) or full Amazon Resource Name (ARN) of the task definition that you want to run in your service. If a revision is not specified, the latest ACTIVE revision is used. If you modify the task definition with UpdateService, Amazon ECS spawns a task with the new version of the task definition and then stops an old task after the new version is running.

Return type dict

Returns

Response Syntax

```
{
  'service': {
    'serviceArn': 'string',
    'serviceName': 'string',
    'clusterArn': 'string',
    'loadBalancers': [
      {
        'loadBalancerName': 'string',
        'containerName': 'string',
        'containerPort': 123
      },
    ],
    'status': 'string',
    'desiredCount': 123,
    'runningCount': 123,
    'pendingCount': 123,
    'taskDefinition': 'string',
    'deployments': [
      {
        'id': 'string',
        'status': 'string',
        'taskDefinition': 'string',
        'desiredCount': 123,
        'pendingCount': 123,
        'runningCount': 123,
        'createdAt': datetime(2015, 1, 1),
```

```
        'updatedAt': datetime(2015, 1, 1)
    },
],
'roleArn': 'string',
'events': [
    {
        'id': 'string',
        'createdAt': datetime(2015, 1, 1),
        'message': 'string'
    },
]
}
}
```

Response Structure

- *(dict)* –
 - **service** (*dict*) –

The full description of your service following the update call.

- * **serviceArn** (*string*) –

The Amazon Resource Name (ARN) that identifies the service. The ARN contains the `arn:aws:ecs` namespace, followed by the region of the service, the AWS account ID of the service owner, the `service` namespace, and then the service name. For example, `arn:aws:ecs:region:012345678910:service/my-service`.

- * **serviceName** (*string*) –

A user-generated string that you can use to identify your service.

- * **clusterArn** (*string*) –

The Amazon Resource Name (ARN) of the of the cluster that hosts the service.

- * **loadBalancers** (*list*) –

A list of load balancer objects, containing the load balancer name, the container name (as it appears in a container definition), and the container port to access from the load balancer.

- (*dict*) –

Details on a load balancer that is used with a service.

- **loadBalancerName** (*string*) –

The name of the load balancer.

- **containerName** (*string*) –

The name of the container to associate with the load balancer.

- **containerPort** (*integer*) –

The port on the container to associate with the load balancer. This port must correspond to a `containerPort` in the service's task definition. Your container instances must allow ingress traffic on the `hostPort` of the port mapping.

- * **status** (*string*) –

The status of the service. The valid values are `ACTIVE`, `DRAINING`, or `INACTIVE`.

- * **desiredCount** (*integer*) –

The desired number of instantiations of the task definition to keep

running on the service. This value is specified when the service is created with `CreateService`, and it can be modified with `UpdateService`.

* **runningCount** (*integer*) –

The number of tasks in the cluster that are in the `RUNNING` state.

* **pendingCount** (*integer*) –

The number of tasks in the cluster that are in the `PENDING` state.

* **taskDefinition** (*string*) –

The task definition to use for tasks in the service. This value is specified when the service is created with `CreateService`, and it can be modified with `UpdateService`.

* **deployments** (*list*) –

The current state of deployments for the service.

· (*dict*) –

The details of an Amazon ECS service deployment.

· **id** (*string*) –

The ID of the deployment.

· **status** (*string*) –

The status of the deployment. Valid values are `PRIMARY` (for the most recent deployment), `ACTIVE` (for previous deployments that still have tasks running, but are being replaced with the `PRIMARY` deployment), and `INACTIVE` (for deployments that have been completely replaced).

· **taskDefinition** (*string*) –

The most recent task definition that was specified for the service to use.

· **desiredCount** (*integer*) –

The most recent desired count of tasks that was specified for the service to deploy and/or maintain.

· **pendingCount** (*integer*) –

The number of tasks in the deployment that are in the `PENDING` status.

· **runningCount** (*integer*) –

The number of tasks in the deployment that are in the `RUNNING` status.

· **createdAt** (*datetime*) –

The Unix time in seconds and milliseconds when the service was created.

· **updatedAt** (*datetime*) –

The Unix time in seconds and milliseconds when the service was last updated.

* **roleArn** (*string*) –

The Amazon Resource Name (ARN) of the IAM role associated with the service that allows the Amazon ECS container agent to register container instances with a load balancer.

* **events** (*list*) –

The event stream for your service. A maximum of 100 of the latest events are displayed.

- *(dict)* –

Details on an event associated with a service.

- **id** (*string*) –

The ID string of the event.

- **createdAt** (*datetime*) –

The Unix time in seconds and milliseconds when the event was triggered.

- **message** (*string*) –

The event message.

Paginators

The available paginators are:

- `ECS.Paginator.ListClusters`
- `ECS.Paginator.ListContainerInstances`
- `ECS.Paginator.ListServices`
- `ECS.Paginator.ListTaskDefinitionFamilies`
- `ECS.Paginator.ListTaskDefinitions`
- `ECS.Paginator.ListTasks`

class `ECS.Paginator.ListClusters`

```
paginator = client.get_paginator('list_clusters')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `ECS.Client.list_clusters()`.

Request Syntax

```
response_iterator = paginator.paginate(
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'clusterArns': [
    'string',
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **clusterArns** (*list*) –

The list of full Amazon Resource Name (ARN) entries for each cluster associated with your account.

 - * (*string*) –
 - **NextToken** (*string*) –

A token to resume pagination.

class `ECS.Paginator.ListContainerInstances`

```
paginator = client.get_paginator('list_container_instances')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `ECS.Client.list_container_instances()`.

Request Syntax

```
response_iterator = paginator.paginate(
    cluster='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that hosts the container instances you want to list. If you do not specify a cluster, the default cluster is assumed..
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –
The size of each page.
- **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
    'containerInstanceArns': [
        'string',
    ],
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **containerInstanceArns** (*list*) –
The list of container instance full Amazon Resource Name (ARN) entries for each container instance associated with the specified cluster.
 - * (*string*) –
 - **NextToken** (*string*) –
A token to resume pagination.

class `ECS.Paginator.ListServices`

```
paginator = client.get_paginator('list_services')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `ECS.Client.list_services()`.

Request Syntax

```
response_iterator = paginator.paginate(
    cluster='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that hosts the services you want to list. If you do not specify a cluster, the default cluster is assumed..
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

– **PageSize** (*integer*) –

The size of each page.

– **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'serviceArns': [
    'string',
  ],
  'NextToken': 'string'
}
```

Response Structure

• (*dict*) –

– **serviceArns** (*list*) –

The list of full Amazon Resource Name (ARN) entries for each service associated with the specified cluster.

* (*string*) –

– **NextToken** (*string*) –

A token to resume pagination.

class ECS.Paginator.**ListTaskDefinitionFamilies**

```
paginator = client.get_paginator('list_task_definition_families')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `ECS.Client.list_task_definition_families()`.

Request Syntax

```
response_iterator = paginator.paginate(
    familyPrefix='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **familyPrefix** (*string*) – The `familyPrefix` is a string that is used to filter the results of `ListTaskDefinitionFamilies`. If you specify a `familyPrefix`, only task definition family names that begin with the `familyPrefix` string are returned.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'families': [
    'string',
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **families** (*list*) –
The list of task definition family names that match the `ListTaskDefinitionFamilies` request.
 - * (*string*) –
 - **NextToken** (*string*) –
A token to resume pagination.

class `ECS.Paginator.ListTaskDefinitions`

```
paginator = client.get_paginator('list_task_definitions')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `ECS.Client.list_task_definitions()`.

Request Syntax

```
response_iterator = paginator.paginate(
    familyPrefix='string',
    status='ACTIVE'|'INACTIVE',
    sort='ASC'|'DESC',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **familyPrefix** (*string*) – The full family name that you want to filter the `ListTaskDefinitions` results with. Specifying a `familyPrefix` will limit the listed task definitions to task definition revisions that belong to that family.
- **status** (*string*) – The task definition status that you want to filter the `ListTaskDefinitions` results with. By default, only `ACTIVE` task definitions are listed. By setting this parameter to `INACTIVE`, you can view task definitions that are `INACTIVE` as long as an active task or service still references them. If you paginate the resulting output, be sure to keep the `status` value constant in each subsequent request.
- **sort** (*string*) – The order in which to sort the results. Valid values are `ASC` and `DESC`. By default (`ASC`), task definitions are listed lexicographically by family name and in ascending numerical order by revision so that the newest task definitions in a family are listed last. Setting this parameter to `DESC` reverses the sort order on family name and revision so that the newest task definitions in a family are listed first.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict**Returns****Response Syntax**

```
{
  'taskDefinitionArns': [
    'string',
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **taskDefinitionArns** (*list*) –
The list of task definition Amazon Resource Name (ARN) entries for the `ListTaskDefinitions` request.
* (*string*) –
 - **NextToken** (*string*) –
A token to resume pagination.

class `ECS.Paginator.ListTasks`

```
paginator = client.get_paginator('list_tasks')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `ECS.Client.list_tasks()`.

Request Syntax

```
response_iterator = paginator.paginate(
    cluster='string',
    containerInstance='string',
    family='string',
    startedBy='string',
    serviceName='string',
    desiredStatus='RUNNING'|'PENDING'|'STOPPED',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that hosts the tasks you want to list. If you do not specify a cluster, the default cluster is assumed..
- **containerInstance** (*string*) – The container instance UUID or full Amazon Resource Name (ARN) of the container instance that you want to filter the `ListTasks` results with. Specifying a `containerInstance` will limit the results to tasks that belong to that container instance.
- **family** (*string*) – The name of the family that you want to filter the `ListTasks` results with. Specifying a `family` will limit the results to tasks that belong to that family.
- **startedBy** (*string*) – The `startedBy` value that you want to filter the task results with. Specifying a `startedBy` value will limit the results to tasks that were started with that value.
- **serviceName** (*string*) – The name of the service that you want to filter the `ListTasks` results with. Specifying a `serviceName` will limit the results to tasks that belong to that service.
- **desiredStatus** (*string*) – The task status that you want to filter the `ListTasks` results with. Specifying a `desiredStatus` of `STOPPED` will limit the results to tasks that are in the `STOPPED` status, which can be useful for debugging tasks that are not starting properly or have died or finished. The default status filter is `RUNNING`.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken`

from a previous response.

Return type dict

Returns

Response Syntax

```
{
    'taskArns': [
        'string',
    ],
    'NextToken': 'string'
}
```

Response Structure

- *(dict)* –
 - **taskArns** (*list*) –

The list of task Amazon Resource Name (ARN) entries for the ListTasks request.

 - * *(string)* –
 - **NextToken** (*string*) –

A token to resume pagination.

Waiters

The available waiters are:

- `ECS.Waiter.ServicesInactive`
- `ECS.Waiter.ServicesStable`
- `ECS.Waiter.TasksRunning`
- `ECS.Waiter.TasksStopped`

class `ECS.Waiter.ServicesInactive`

```
waiter = client.get_waiter('services_inactive')
```

wait (***kwargs*)

Polls `ECS.Client.describe_services()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(
    cluster='string',
    services=[
        'string',
    ]
)
```

Parameters

- **cluster** (*string*) – The name of the cluster that hosts the service you want to describe.

- **services** (*list*) – [REQUIRED]

A list of services you want to describe.
– (*string*) –

Returns None

class ECS.Waiter.**ServicesStable**

```
waiter = client.get_waiter('services_stable')
```

wait (***kwargs*)

Polls `ECS.Client.describe_services()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(  
    cluster='string',  
    services=[  
        'string',  
    ]  
)
```

Parameters

- **cluster** (*string*) – The name of the cluster that hosts the service you want to describe.
- **services** (*list*) – [REQUIRED]
A list of services you want to describe.
– (*string*) –

Returns None

class ECS.Waiter.**TasksRunning**

```
waiter = client.get_waiter('tasks_running')
```

wait (***kwargs*)

Polls `ECS.Client.describe_tasks()` every 6 seconds until a successful state is reached. An error is returned after 100 failed checks.

Request Syntax

```
waiter.wait(  
    cluster='string',  
    tasks=[  
        'string',  
    ]  
)
```

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that hosts the task you want to describe. If you do not specify a cluster, the default cluster is assumed.
- **tasks** (*list*) – [REQUIRED]
A space-separated list of task UUIDs or full Amazon Resource Name (ARN) entries.

– (string) –

Returns None

class `ECS.Waiter.TasksStopped`

```
waiter = client.get_waiter('tasks_stopped')
```

wait (***kwargs*)

Polls `ECS.Client.describe_tasks()` every 6 seconds until a successful state is reached. An error is returned after 100 failed checks.

Request Syntax

```
waiter.wait(
    cluster='string',
    tasks=[
        'string',
    ]
)
```

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that hosts the task you want to describe. If you do not specify a cluster, the default cluster is assumed.
- **tasks** (*list*) – **[REQUIRED]**

A space-separated list of task UUIDs or full Amazon Resource Name (ARN) entries.

– (string) –

Returns None

EFS

Table of Contents

- [EFS](#)
 - [Client](#)

Client

class `EFS.Client`

A low-level client representing Amazon Elastic File System (EFS):

```
import boto3
```

```
client = boto3.client('efs')
```

These are the available methods:

- `can_paginate()`
- `create_file_system()`
- `create_mount_target()`
- `create_tags()`

- `delete_file_system()`
- `delete_mount_target()`
- `delete_tags()`
- `describe_file_systems()`
- `describe_mount_target_security_groups()`
- `describe_mount_targets()`
- `describe_tags()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_waiter()`
- `modify_mount_target_security_groups()`

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you’d normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

create_file_system (***kwargs*)

Creates a new, empty file system. The operation requires a creation token in the request that Amazon EFS uses to ensure idempotent creation (calling the operation with same creation token has no effect). If a file system does not currently exist that is owned by the caller’s AWS account with the specified creation token, this operation does the following:

- Creates a new, empty file system. The file system will have an Amazon EFS assigned ID, and an initial lifecycle state “creating”.
- Returns with the description of the created file system.

Otherwise, this operation returns a `FileSystemAlreadyExists` error with the ID of the existing file system.

Note: For basic use cases, you can use a randomly generated UUID for the creation token.

The idempotent operation allows you to retry a `CreateFileSystem` call without risk of creating an extra file system. This can happen when an initial call fails in a way that leaves it uncertain whether or not a file system was actually created. An example might be that a transport level timeout occurred or your connection was reset. As long as you use the same creation token, if the initial call had succeeded in creating a file system, the client can learn of its existence from the `FileSystemAlreadyExists` error.

Note: The `CreateFileSystem` call returns while the file system’s lifecycle state is still “creating”. You can check the file system creation status by calling the `DescribeFileSystems` API, which among other things returns the file system state.

After the file system is fully created, Amazon EFS sets its lifecycle state to “available”, at which point you can create one or more mount targets for the file system (`CreateMountTarget`) in your VPC. You mount your Amazon EFS file system on an EC2 instances in your VPC via the mount target. For more information, see [Amazon EFS: How it Works](#)

This operation requires permission for the `elasticfilesystem:CreateFileSystem` action.

Request Syntax

```
response = client.create_file_system(
    CreationToken='string'
```

)

Parameters **CreationToken** (*string*) – [REQUIRED]

String of up to 64 ASCII characters. Amazon EFS uses this to ensure idempotent creation.

Return type dict**Returns****Response Syntax**

```
{
  'OwnerId': 'string',
  'CreationToken': 'string',
  'FileSystemId': 'string',
  'CreationTime': datetime(2015, 1, 1),
  'LifeCycleState': 'creating'|'available'|'deleting'|'deleted',
  'Name': 'string',
  'NumberOfMountTargets': 123,
  'SizeInBytes': {
    'Value': 123,
    'Timestamp': datetime(2015, 1, 1)
  }
}
```

Response Structure

- (*dict*) –

This object provides description of a file system.

- **OwnerId** (*string*) –

The AWS account that created the file system. If the file system was created by an IAM user, the parent account to which the user belongs is the owner.

- **CreationToken** (*string*) –

Opaque string specified in the request.

- **FileSystemId** (*string*) –

The file system ID assigned by Amazon EFS.

- **CreationTime** (*datetime*) –

The time at which the file system was created, in seconds, since 1970-01-01T00:00:00Z.

- **LifeCycleState** (*string*) –

A predefined string value that indicates the lifecycle phase of the file system.

- **Name** (*string*) –

You can add tags to a file system (see `CreateTags`) including a “Name” tag. If the file system has a “Name” tag, Amazon EFS returns the value in this field.

- **NumberOfMountTargets** (*integer*) –

The current number of mount targets (see `CreateMountTarget`) the file system has.

- **SizeInBytes** (*dict*) –

This object provides the latest known metered size of data stored in the file system, in bytes, in its `Value` field, and the time at which that size was determined in its `Timestamp` field. The `Timestamp` value is the integer number of seconds since 1970-01-01T00:00:00Z. Note that the value does not represent the size of a consistent snapshot of the file system, but it is eventually consistent when there are no writes to the file system. That is, the value will represent actual size only if the file system is not modified for a period longer than a couple of hours. Otherwise, the value is not the exact size the file system was at any instant in time.

* **Value** (*integer*) –

The latest known metered size, in bytes, of data stored in the file system.

* **Timestamp** (*datetime*) –

The time at which the size of data, returned in the `Value` field, was determined. The value is the integer number of seconds since 1970-01-01T00:00:00Z.

create_mount_target (***kwargs*)

Creates a mount target for a file system. You can then mount the file system on EC2 instances via the mount target.

You can create one mount target in each Availability Zone in your VPC. All EC2 instances in a VPC within a given Availability Zone share a single mount target for a given file system. If you have multiple subnets in an Availability Zone, you create a mount target in one of the subnets. EC2 instances do not need to be in the same subnet as the mount target in order to access their file system. For more information, see [Amazon EFS: How it Works](#).

In the request, you also specify a file system ID for which you are creating the mount target and the file system's lifecycle state must be "available" (see `DescribeFileSystems`).

In the request, you also provide a subnet ID, which serves several purposes:

- It determines the VPC in which Amazon EFS creates the mount target.
- It determines the Availability Zone in which Amazon EFS creates the mount target.
- It determines the IP address range from which Amazon EFS selects the IP address of the mount target if you don't specify an IP address in the request.

After creating the mount target, Amazon EFS returns a response that includes, a `MountTargetId` and an `IpAddress`. You use this IP address when mounting the file system in an EC2 instance. You can also use the mount target's DNS name when mounting the file system. The EC2 instance on which you mount the file system via the mount target can resolve the mount target's DNS name to its IP address. For more information, see [How it Works: Implementation Overview](#).

Note that you can create mount targets for a file system in only one VPC, and there can be only one mount target per Availability Zone. That is, if the file system already has one or more mount targets created for it, the request to add another mount target must meet the following requirements:

- The subnet specified in the request must belong to the same VPC as the subnets of the existing mount targets.
- The subnet specified in the request must not be in the same Availability Zone as any of the subnets of the existing mount targets.

If the request satisfies the requirements, Amazon EFS does the following:

- Creates a new mount target in the specified subnet.
- Also creates a new network interface in the subnet as follows: * If the request provides an `IpAddress`, Amazon EFS assigns that IP address to the network interface. Otherwise, Amazon EFS assigns a free address in the subnet (in the same way that the Amazon EC2 `CreateNetworkInterface` call does when a request does not specify a primary private IP address).

- If the request provides `SecurityGroups`, this network interface is associated with those security groups. Otherwise, it belongs to the default security group for the subnet's VPC.
- Assigns the description "Mount target `*fsmt-id*` for file system `*fs-id*`" where `*fsmt-id*` is the mount target ID, and `*fs-id*` is the `FileSystemId`.
- Sets the `requesterManaged` property of the network interface to "true", and the `requesterId` value to "EFS".

Each Amazon EFS mount target has one corresponding requester-managed EC2 network interface. After the network interface is created, Amazon EFS sets the `NetworkInterfaceId` field in the mount target's description to the network interface ID, and the `IpAddress` field to its address. If network interface creation fails, the entire `CreateMountTarget` operation fails.

Note: The `CreateMountTarget` call returns only after creating the network interface, but while the mount target state is still "creating". You can check the mount target creation status by calling the `DescribeFileSystems` API, which among other things returns the mount target state.

We recommend you create a mount target in each of the Availability Zones. There are cost considerations for using a file system in an Availability Zone through a mount target created in another Availability Zone. For more information, go to [Amazon EFS](#) product detail page. In addition, by always using a mount target local to the instance's Availability Zone, you eliminate a partial failure scenario; if the Availability Zone in which your mount target is created goes down, then you won't be able to access your file system through that mount target.

This operation requires permission for the following action on the file system:

- `elasticfilesystem:CreateMountTarget`

This operation also requires permission for the following Amazon EC2 actions:

- `ec2:DescribeSubnets`
- `ec2:DescribeNetworkInterfaces`
- `ec2:CreateNetworkInterface`

Request Syntax

```
response = client.create_mount_target(
    FileSystemId='string',
    SubnetId='string',
    IpAddress='string',
    SecurityGroups=[
        'string',
    ]
)
```

Parameters

- **FileSystemId** (*string*) – [REQUIRED]
The ID of the file system for which to create the mount target.
- **SubnetId** (*string*) – [REQUIRED]
The ID of the subnet to add the mount target in.
- **IpAddress** (*string*) – A valid IPv4 address within the address range of the specified subnet.
- **SecurityGroups** (*list*) – Up to 5 VPC security group IDs, of the form "sg-xxxxxxx". These must be for the same VPC as subnet specified.
– (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'OwnerId': 'string',
    'MountTargetId': 'string',
    'FileSystemId': 'string',
    'SubnetId': 'string',
    'LifecycleState': 'creating'|'available'|'deleting'|'deleted',
    'IpAddress': 'string',
    'NetworkInterfaceId': 'string'
}
```

Response Structure

- (*dict*) –

This object provides description of a mount target.

- **OwnerId** (*string*) –

The AWS account ID that owns the resource.

- **MountTargetId** (*string*) –

The system-assigned mount target ID.

- **FileSystemId** (*string*) –

The ID of the file system for which the mount target is intended.

- **SubnetId** (*string*) –

The ID of the subnet that the mount target is in.

- **LifecycleState** (*string*) –

The lifecycle state the mount target is in.

- **IpAddress** (*string*) –

The address at which the file system may be mounted via the mount target.

- **NetworkInterfaceId** (*string*) –

The ID of the network interface that Amazon EFS created when it created the mount target.

create_tags (***kwargs*)

Creates or overwrites tags associated with a file system. Each tag is a key-value pair. If a tag key specified in the request already exists on the file system, this operation overwrites its value with the value provided in the request. If you add the “Name” tag to your file system, Amazon EFS returns it in the response to the DescribeFileSystems API.

This operation requires permission for the `elasticfilesystem:CreateTags` action.

Request Syntax

```
response = client.create_tags(
    FileSystemId='string',
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
)
```

Parameters

- **FileSystemId** (*string*) – [REQUIRED]

String. The ID of the file system whose tags you want to modify. This operation modifies only the tags and not the file system.

- **Tags** (*list*) – [REQUIRED]

An array of Tag objects to add. Each Tag object is a key-value pair.

- (*dict*) –

A tag is a pair of key and value. The allowed characters in keys and values are letters, whitespace, and numbers, representable in UTF-8, and the characters '+', '-', '=', ':', '_', '.', and '/'.

- * **Key** (*string*) – [REQUIRED]

Tag key, a string. The key must not start with “aws:”.

- * **Value** (*string*) – [REQUIRED]

Value of the tag key.

Returns None

delete_file_system (***kwargs*)

Deletes a file system, permanently severing access to its contents. Upon return, the file system no longer exists and you will not be able to access any contents of the deleted file system.

You cannot delete a file system that is in use. That is, if the file system has any mount targets, you must first delete them. For more information, see DescribeMountTargets and DeleteMountTarget .

Note: The DeleteFileSystem call returns while the file system state is still “deleting”. You can check the file system deletion status by calling the DescribeFileSystems API, which returns a list of file systems in your account. If you pass file system ID or creation token for the deleted file system, the DescribeFileSystems will return a 404 “FileSystemNotFound” error.

This operation requires permission for the `elasticfilesystem:DeleteFileSystem` action.

Request Syntax

```
response = client.delete_file_system(
    FileSystemId='string'
)
```

Parameters `FileSystemId` (*string*) – [REQUIRED]

The ID of the file system you want to delete.

Returns None

delete_mount_target (***kwargs*)

Deletes the specified mount target.

This operation forcibly breaks any mounts of the file system via the mount target being deleted, which might disrupt instances or applications using those mounts. To avoid applications getting cut off abruptly, you might consider unmounting any mounts of the mount target, if feasible. The operation also deletes the associated network interface. Uncommitted writes may be lost, but breaking a mount target using this operation does not corrupt the file system itself. The file system you created remains. You can mount an EC2 instance in your VPC using another mount target.

This operation requires permission for the following action on the file system:

- `elasticfilesystem:DeleteMountTarget`

Note: The DeleteMountTarget call returns while the mount target state is still “deleting”. You can check the mount target deletion by calling the DescribeMountTargets API, which returns a list of mount target descriptions for the given file system.

The operation also requires permission for the following Amazon EC2 action on the mount target's network interface:

- `ec2:DeleteNetworkInterface`

Request Syntax

```
response = client.delete_mount_target(
    MountTargetId='string'
)
```

Parameters `MountTargetId` (*string*) – [REQUIRED]

String. The ID of the mount target to delete.

Returns None

`delete_tags` (***kwargs*)

Deletes the specified tags from a file system. If the `DeleteTags` request includes a tag key that does not exist, Amazon EFS ignores it; it is not an error. For more information about tags and related restrictions, go to [Tag Restrictions](#) in the *AWS Billing and Cost Management User Guide*.

This operation requires permission for the `elasticfilesystem:DeleteTags` action.

Request Syntax

```
response = client.delete_tags(
    FileSystemId='string',
    TagKeys=[
        'string',
    ]
)
```

Parameters

- **`FileSystemId`** (*string*) – [REQUIRED]

String. The ID of the file system whose tags you want to delete.

- **`TagKeys`** (*list*) – [REQUIRED]

A list of tag keys to delete.

– (*string*) –

Returns None

`describe_file_systems` (***kwargs*)

Returns the description of a specific Amazon EFS file system if either the file system `CreationToken` or the `FileSystemId` is provided; otherwise, returns descriptions of all file systems owned by the caller's AWS account in the AWS region of the endpoint that you're calling.

When retrieving all file system descriptions, you can optionally specify the `MaxItems` parameter to limit the number of descriptions in a response. If more file system descriptions remain, Amazon EFS returns a `NextMarker`, an opaque token, in the response. In this case, you should send a subsequent request with the `Marker` request parameter set to the value of `NextMarker`.

So to retrieve a list of your file system descriptions, the expected usage of this API is an iterative process of first calling `DescribeFileSystems` without the `Marker` and then continuing to call it with the `Marker` parameter set to the value of the `NextMarker` from the previous response until the response has no `NextMarker`.

Note that the implementation may return fewer than `MaxItems` file system descriptions while still including a `NextMarker` value.

The order of file systems returned in the response of one `DescribeFileSystems` call, and the order of file systems returned across the responses of a multi-call iteration, is unspecified.

This operation requires permission for the `elasticfilesystem:DescribeFileSystems` action.

Request Syntax

```
response = client.describe_file_systems(
    MaxItems=123,
    Marker='string',
    CreationToken='string',
    FileSystemId='string'
)
```

Parameters

- **MaxItems** (*integer*) – Optional integer. Specifies the maximum number of file systems to return in the response. This parameter value must be greater than 0. The number of items Amazon EFS returns will be the minimum of the `MaxItems` parameter specified in the request and the service’s internal maximum number of items per page.
- **Marker** (*string*) – Optional string. Opaque pagination token returned from a previous `DescribeFileSystems` operation. If present, specifies to continue the list from where the returning call had left off.
- **CreationToken** (*string*) – Optional string. Restricts the list to the file system with this creation token (you specify a creation token at the time of creating an Amazon EFS file system).
- **FileSystemId** (*string*) – Optional string. File system ID whose description you want to retrieve.

Return type dict

Returns

Response Syntax

```
{
    'Marker': 'string',
    'FileSystems': [
        {
            'OwnerId': 'string',
            'CreationToken': 'string',
            'FileSystemId': 'string',
            'CreationTime': datetime(2015, 1, 1),
            'LifecycleState': 'creating'|'available'|'deleting'|'deleted',
            'Name': 'string',
            'NumberOfMountTargets': 123,
            'SizeInBytes': {
                'Value': 123,
                'Timestamp': datetime(2015, 1, 1)
            }
        },
    ],
    'NextMarker': 'string'
}
```

Response Structure

- (*dict*) –
 - **Marker** (*string*) –
A string, present if provided by caller in the request.
 - **FileSystems** (*list*) –
An array of file system descriptions.

* (*dict*) –

This object provides description of a file system.

- **OwnerId** (*string*) –

The AWS account that created the file system. If the file system was created by an IAM user, the parent account to which the user belongs is the owner.

- **CreationToken** (*string*) –

Opaque string specified in the request.

- **FileSystemId** (*string*) –

The file system ID assigned by Amazon EFS.

- **CreationTime** (*datetime*) –

The time at which the file system was created, in seconds, since 1970-01-01T00:00:00Z.

- **LifeCycleState** (*string*) –

A predefined string value that indicates the lifecycle phase of the file system.

- **Name** (*string*) –

You can add tags to a file system (see `CreateTags`) including a “Name” tag. If the file system has a “Name” tag, Amazon EFS returns the value in this field.

- **NumberOfMountTargets** (*integer*) –

The current number of mount targets (see `CreateMountTarget`) the file system has.

- **SizeInBytes** (*dict*) –

This object provides the latest known metered size of data stored in the file system, in bytes, in its `Value` field, and the time at which that size was determined in its `Timestamp` field. The `Timestamp` value is the integer number of seconds since 1970-01-01T00:00:00Z. Note that the value does not represent the size of a consistent snapshot of the file system, but it is eventually consistent when there are no writes to the file system. That is, the value will represent actual size only if the file system is not modified for a period longer than a couple of hours. Otherwise, the value is not the exact size the file system was at any instant in time.

- **Value** (*integer*) –

The latest known metered size, in bytes, of data stored in the file system.

- **Timestamp** (*datetime*) –

The time at which the size of data, returned in the `Value` field, was determined. The value is the integer number of seconds since 1970-01-01T00:00:00Z.

- **NextMarker** (*string*) –

A string, present if there are more file systems than returned in the response. You can use the `NextMarker` in the subsequent request to fetch the descriptions.

describe_mount_target_security_groups (***kwargs*)

Returns the security groups currently in effect for a mount target. This operation requires that the network interface of the mount target has been created and the life cycle state of the mount target is not “deleted”.

This operation requires permissions for the following actions:

- `elasticfilesystem:DescribeMountTargetSecurityGroups` action on the mount target’s file system.
- `ec2:DescribeNetworkInterfaceAttribute` action on the mount target’s network interface.

Request Syntax

```
response = client.describe_mount_target_security_groups(
    MountTargetId='string'
)
```

Parameters `MountTargetId` (*string*) – [REQUIRED]

The ID of the mount target whose security groups you want to retrieve.

Return type dict

Returns**Response Syntax**

```
{
    'SecurityGroups': [
        'string',
    ]
}
```

Response Structure

- (*dict*) –
 - **SecurityGroups** (*list*) –

An array of security groups.

* (*string*) –

describe_mount_targets (***kwargs*)

Returns the descriptions of all the current mount targets, or a specific mount target, for a file system. When requesting all of the current mount targets, the order of mount targets returned in the response is unspecified.

This operation requires permission for the `elasticfilesystem:DescribeMountTargets` action, on either the file system id that you specify in `FileSystemId`, or on the file system of the mount target that you specify in `MountTargetId`.

Request Syntax

```
response = client.describe_mount_targets(
    MaxItems=123,
    Marker='string',
    FileSystemId='string',
    MountTargetId='string'
)
```

Parameters

- **MaxItems** (*integer*) – Optional. Maximum number of mount targets to return in the response. It must be an integer with a value greater than zero.

- **Marker** (*string*) – Optional. String. Opaque pagination token returned from a previous `DescribeMountTargets` operation. If present, it specifies to continue the list from where the previous returning call left off.
- **FileSystemId** (*string*) – Optional. String. The ID of the file system whose mount targets you want to list. It must be included in your request if `MountTargetId` is not included.
- **MountTargetId** (*string*) – Optional. String. The ID of the mount target that you want to have described. It must be included in your request if `FileSystemId` is not included.

Return type dict

Returns

Response Syntax

```
{
    'Marker': 'string',
    'MountTargets': [
        {
            'OwnerId': 'string',
            'MountTargetId': 'string',
            'FileSystemId': 'string',
            'SubnetId': 'string',
            'LifecycleState': 'creating'|'available'|'deleting'|'deleted',
            'IpAddress': 'string',
            'NetworkInterfaceId': 'string'
        },
    ],
    'NextMarker': 'string'
}
```

Response Structure

- (*dict*) –
 - **Marker** (*string*) –

If the request included the `Marker`, the response returns that value in this field.
 - **MountTargets** (*list*) –

Returns the file system's mount targets as an array of `MountTargetDescription` objects.

 - * (*dict*) –

This object provides description of a mount target.

 - **OwnerId** (*string*) –

The AWS account ID that owns the resource.
 - **MountTargetId** (*string*) –

The system-assigned mount target ID.
 - **FileSystemId** (*string*) –

The ID of the file system for which the mount target is intended.
 - **SubnetId** (*string*) –

The ID of the subnet that the mount target is in.
 - **LifecycleState** (*string*) –

The lifecycle state the mount target is in.

- **IpAddress** (*string*) –
The address at which the file system may be mounted via the mount target.
- **NetworkInterfaceId** (*string*) –
The ID of the network interface that Amazon EFS created when it created the mount target.
- **NextMarker** (*string*) –
If a value is present, there are more mount targets to return. In a subsequent request, you can provide `Marker` in your request with this value to retrieve the next set of mount targets.

describe_tags (***kwargs*)

Returns the tags associated with a file system. The order of tags returned in the response of one `DescribeTags` call, and the order of tags returned across the responses of a multi-call iteration (when using pagination), is unspecified.

This operation requires permission for the `elasticfilesystem:DescribeTags` action.

Request Syntax

```
response = client.describe_tags(
    MaxItems=123,
    Marker='string',
    FileSystemId='string'
)
```

Parameters

- **MaxItems** (*integer*) – Optional. Maximum number of file system tags to return in the response. It must be an integer with a value greater than zero.
- **Marker** (*string*) – Optional. String. Opaque pagination token returned from a previous `DescribeTags` operation. If present, it specifies to continue the list from where the previous call left off.
- **FileSystemId** (*string*) – **[REQUIRED]**

The ID of the file system whose tag set you want to retrieve.

Return type dict

Returns

Response Syntax

```
{
    'Marker': 'string',
    'Tags': [
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
    'NextMarker': 'string'
}
```

Response Structure

- (*dict*) –
 - **Marker** (*string*) –
If the request included a `Marker`, the response returns that value in this field.

– **Tags** (*list*) –

Returns tags associated with the file system as an array of `Tag` objects.

* (*dict*) –

A tag is a pair of key and value. The allowed characters in keys and values are letters, whitespace, and numbers, representable in UTF-8, and the characters '+', '-', '=', ':', '_', ':', and '/'.

· **Key** (*string*) –

Tag key, a string. The key must not start with “aws:”.

· **Value** (*string*) –

Value of the tag key.

– **NextMarker** (*string*) –

If a value is present, there are more tags to return. In a subsequent request, you can provide the value of `NextMarker` as the value of the `Marker` parameter in your next request to retrieve the next set of tags.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method’s model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you’d normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_waiter (*waiter_name*)

modify_mount_target_security_groups (***kwargs*)

Modifies the set of security groups in effect for a mount target.

When you create a mount target, Amazon EFS also creates a new network interface (see `CreateMountTarget`). This operation replaces the security groups in effect for the network interface associated with a mount target, with the `SecurityGroups` provided in the request. This operation requires that the network interface of the mount target has been created and the life cycle state of the mount target is not “deleted”.

The operation requires permissions for the following actions:

- `elasticfilesystem:ModifyMountTargetSecurityGroups` action on the mount target’s file system.
- `ec2:ModifyNetworkInterfaceAttribute` action on the mount target’s network interface.

Request Syntax

```
response = client.modify_mount_target_security_groups(
    MountTargetId='string',
    SecurityGroups=[
        'string',
    ]
)
```

Parameters

- **MountTargetId** (*string*) – [REQUIRED]
The ID of the mount target whose security groups you want to modify.
- **SecurityGroups** (*list*) – An array of up to five VPC security group IDs.
– (*string*) –

Returns None

ElastiCache

Table of Contents

- [ElastiCache](#)
 - [Client](#)
 - [Paginators](#)
 - [Waiters](#)

Client

class `ElastiCache.Client`

A low-level client representing Amazon ElastiCache:

```
import boto3
```

```
client = boto3.client('elasticache')
```

These are the available methods:

- `add_tags_to_resource()`
- `authorize_cache_security_group_ingress()`
- `can_paginate()`
- `copy_snapshot()`
- `create_cache_cluster()`
- `create_cache_parameter_group()`
- `create_cache_security_group()`
- `create_cache_subnet_group()`
- `create_replication_group()`
- `create_snapshot()`
- `delete_cache_cluster()`
- `delete_cache_parameter_group()`
- `delete_cache_security_group()`
- `delete_cache_subnet_group()`
- `delete_replication_group()`
- `delete_snapshot()`

- `describe_cache_clusters()`
- `describe_cache_engine_versions()`
- `describe_cache_parameter_groups()`
- `describe_cache_parameters()`
- `describe_cache_security_groups()`
- `describe_cache_subnet_groups()`
- `describe_engine_default_parameters()`
- `describe_events()`
- `describe_replication_groups()`
- `describe_reserved_cache_nodes()`
- `describe_reserved_cache_nodes_offerings()`
- `describe_snapshots()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_waiter()`
- `list_tags_for_resource()`
- `modify_cache_cluster()`
- `modify_cache_parameter_group()`
- `modify_cache_subnet_group()`
- `modify_replication_group()`
- `purchase_reserved_cache_nodes_offering()`
- `reboot_cache_cluster()`
- `remove_tags_from_resource()`
- `reset_cache_parameter_group()`
- `revoke_cache_security_group_ingress()`

`add_tags_to_resource(**kwargs)`

The *AddTagsToResource* action adds up to 10 cost allocation tags to the named resource. A *cost allocation tag* is a key-value pair where the key and value are case-sensitive. Cost allocation tags can be used to categorize and track your AWS costs.

When you apply tags to your ElastiCache resources, AWS generates a cost allocation report as a comma-separated value (CSV) file with your usage and costs aggregated by your tags. You can apply tags that represent business categories (such as cost centers, application names, or owners) to organize your costs across multiple services. For more information, see [Using Cost Allocation Tags in Amazon ElastiCache](#).

Request Syntax

```
response = client.add_tags_to_resource(
    ResourceName='string',
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **ResourceName** (*string*) – **[REQUIRED]**

The name of the resource to which the tags are to be added, for example `arn:aws:elasticache:us-west-2:0123456789:cluster:myCluster`.

- **Tags** (*list*) – **[REQUIRED]**

A list of cost allocation tags to be added to this resource. A tag is a key-value pair. A tag key must be accompanied by a tag value.

– (dict) –

A cost allocation Tag that can be added to an ElastiCache cluster or replication group. Tags are composed of a Key/Value pair. A tag with a null Value is permitted.

* **Key** (*string*) –

The key for the tag.

* **Value** (*string*) –

The tag’s value. May not be null.

Return type dict

Returns

Response Syntax

```
{
  'TagList': [
    {
      'Key': 'string',
      'Value': 'string'
    },
  ]
}
```

Response Structure

• (dict) –

Represents the output from the *AddTagsToResource* , *ListTagsOnResource* , and *RemoveTagsFromResource* actions.

– **TagList** (*list*) –

A list of cost allocation tags as key-value pairs.

* (dict) –

A cost allocation Tag that can be added to an ElastiCache cluster or replication group. Tags are composed of a Key/Value pair. A tag with a null Value is permitted.

· **Key** (*string*) –

The key for the tag.

· **Value** (*string*) –

The tag’s value. May not be null.

authorize_cache_security_group_ingress (**kwargs)

The *AuthorizeCacheSecurityGroupIngress* action allows network ingress to a cache security group. Applications using ElastiCache must be running on Amazon EC2, and Amazon EC2 security groups are used as the authorization mechanism.

Note: You cannot authorize ingress from an Amazon EC2 security group in one region to an ElastiCache cluster in another region.

Request Syntax

```
response = client.authorize_cache_security_group_ingress(
    CacheSecurityGroupName='string',
    EC2SecurityGroupName='string',
    EC2SecurityGroupOwnerId='string'
)
```

Parameters

- **CacheSecurityGroupName** (*string*) – **[REQUIRED]**
The cache security group which will allow network ingress.
- **EC2SecurityGroupName** (*string*) – **[REQUIRED]**
The Amazon EC2 security group to be authorized for ingress to the cache security group.
- **EC2SecurityGroupOwnerId** (*string*) – **[REQUIRED]**
The AWS account number of the Amazon EC2 security group owner. Note that this is not the same thing as an AWS access key ID - you must provide a valid AWS account number for this parameter.

Return type dict**Returns****Response Syntax**

```
{
  'CacheSecurityGroup': {
    'OwnerId': 'string',
    'CacheSecurityGroupName': 'string',
    'Description': 'string',
    'EC2SecurityGroups': [
      {
        'Status': 'string',
        'EC2SecurityGroupName': 'string',
        'EC2SecurityGroupOwnerId': 'string'
      },
    ]
  }
}
```

Response Structure

- (*dict*) –

Represents the output of one of the following actions:

- *AuthorizeCacheSecurityGroupIngress*
- *CreateCacheSecurityGroup*
- *RevokeCacheSecurityGroupIngress*
- **CacheSecurityGroup** (*dict*) –

Represents the output of one of the following actions:

- * *AuthorizeCacheSecurityGroupIngress*
- * *CreateCacheSecurityGroup*
- * *RevokeCacheSecurityGroupIngress*
- * **OwnerId** (*string*) –

The AWS account ID of the cache security group owner.

- * **CacheSecurityGroupName** (*string*) –

The name of the cache security group.

- * **Description** (*string*) –

The description of the cache security group.

- * **EC2SecurityGroups** (*list*) –

A list of Amazon EC2 security groups that are associated with this cache security group.

- *(dict)* –
Provides ownership and status information for an Amazon EC2 security group.
- **Status** (*string*) –
The status of the Amazon EC2 security group.
- **EC2SecurityGroupName** (*string*) –
The name of the Amazon EC2 security group.
- **EC2SecurityGroupOwnerId** (*string*) –
The AWS account ID of the Amazon EC2 security group owner.

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

copy_snapshot (***kwargs*)

The *CopySnapshot* action makes a copy of an existing snapshot.

Request Syntax

```
response = client.copy_snapshot(
    SourceSnapshotName='string',
    TargetSnapshotName='string'
)
```

Parameters

- **SourceSnapshotName** (*string*) – [REQUIRED]
The name of an existing snapshot from which to copy.
- **TargetSnapshotName** (*string*) – [REQUIRED]
A name for the copied snapshot.

Return type dict

Returns

Response Syntax

```
{
  'Snapshot': {
    'SnapshotName': 'string',
    'CacheClusterId': 'string',
    'SnapshotStatus': 'string',
    'SnapshotSource': 'string',
    'CacheNodeType': 'string',
    'Engine': 'string',
    'EngineVersion': 'string',
    'NumCacheNodes': 123,
    'PreferredAvailabilityZone': 'string',
    'CacheClusterCreateTime': datetime(2015, 1, 1),
    'PreferredMaintenanceWindow': 'string',
    'TopicArn': 'string',
```

```
'Port': 123,
'CacheParameterGroupName': 'string',
'CacheSubnetGroupName': 'string',
'VpcId': 'string',
'AutoMinorVersionUpgrade': True|False,
'SnapshotRetentionLimit': 123,
'SnapshotWindow': 'string',
'NodeSnapshots': [
    {
        'CacheNodeId': 'string',
        'CacheSize': 'string',
        'CacheNodeCreateTime': datetime(2015, 1, 1),
        'SnapshotCreateTime': datetime(2015, 1, 1)
    },
]
}
```

Response Structure

- *(dict)* –

Represents a copy of an entire cache cluster as of the time when the snapshot was taken.

- **Snapshot** (*dict*) –

Represents a copy of an entire cache cluster as of the time when the snapshot was taken.

- * **SnapshotName** (*string*) –

The name of a snapshot. For an automatic snapshot, the name is system-generated; for a manual snapshot, this is the user-provided name.

- * **CacheClusterId** (*string*) –

The user-supplied identifier of the source cache cluster.

- * **SnapshotStatus** (*string*) –

The status of the snapshot. Valid values: `creating|available|restoring|copying|deleting`.

- * **SnapshotSource** (*string*) –

Indicates whether the snapshot is from an automatic backup (`automated`) or was created manually (`manual`).

- * **CacheNodeType** (*string*) –

The name of the compute and memory capacity node type for the source cache cluster.

Valid node types are as follows:

- General purpose: * Current generation: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
- Previous generation: `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
- Compute optimized: `cache.c1.xlarge`
- Memory optimized * Current generation: `cache.r3.large`, `cache.r3.xlarge`,

```

cache.r3.2xlarge    ,    cache.r3.4xlarge    ,
cache.r3.8xlarge
· Previous generation:    cache.m2.xlarge    ,
cache.m2.2xlarge , cache.m2.4xlarge

```

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#).

* **Engine** (*string*) –

The name of the cache engine (*memcached* or *redis*) used by the source cache cluster.

* **EngineVersion** (*string*) –

The version of the cache engine version that is used by the source cache cluster.

* **NumCacheNodes** (*integer*) –

The number of cache nodes in the source cache cluster.

For clusters running Redis, this value must be 1. For clusters running Memcached, this value must be between 1 and 20.

* **PreferredAvailabilityZone** (*string*) –

The name of the Availability Zone in which the source cache cluster is located.

* **CacheClusterCreateTime** (*datetime*) –

The date and time when the source cache cluster was created.

* **PreferredMaintenanceWindow** (*string*) –

Specifies the weekly time range during which maintenance on the cache cluster is performed. It is specified as a range in the format ddd:hh24:mi-ddd:hh24:mi (24H Clock UTC). The minimum maintenance window is a 60 minute period. Valid values for ddd are:

- sun
- mon
- tue
- wed
- thu
- fri
- sat

Example: sun:05:00-sun:09:00

* **TopicArn** (*string*) –

The Amazon Resource Name (ARN) for the topic used by the source cache cluster for publishing notifications.

* **Port** (*integer*) –

The port number used by each cache nodes in the source cache cluster.

* **CacheParameterGroupName** (*string*) –

The cache parameter group that is associated with the source cache cluster.

* **CacheSubnetGroupName** (*string*) –

The name of the cache subnet group associated with the source cache cluster.

* **VpcId** (*string*) –

The Amazon Virtual Private Cloud identifier (VPC ID) of the cache subnet group for the source cache cluster.

* **AutoMinorVersionUpgrade** (*boolean*) –

This parameter is currently disabled.

* **SnapshotRetentionLimit** (*integer*) –

For an automatic snapshot, the number of days for which ElastiCache will retain the snapshot before deleting it.

For manual snapshots, this field reflects the *SnapshotRetentionLimit* for the source cache cluster when the snapshot was created. This field is otherwise ignored: Manual snapshots do not expire, and can only be deleted using the *DeleteSnapshot* action.

Important If the value of *SnapshotRetentionLimit* is set to zero (0), backups are turned off.

* **SnapshotWindow** (*string*) –

The daily time range during which ElastiCache takes daily snapshots of the source cache cluster.

* **NodeSnapshots** (*list*) –

A list of the cache nodes in the source cache cluster.

· (*dict*) –

Represents an individual cache node in a snapshot of a cache cluster.

· **CacheNodeId** (*string*) –

The cache node identifier for the node in the source cache cluster.

· **CacheSize** (*string*) –

The size of the cache on the source cache node.

· **CacheNodeCreateTime** (*datetime*) –

The date and time when the cache node was created in the source cache cluster.

· **SnapshotCreateTime** (*datetime*) –

The date and time when the source node's metadata and cache data set was obtained for the snapshot.

create_cache_cluster (***kwargs*)

The *CreateCacheCluster* action creates a cache cluster. All nodes in the cache cluster run the same protocol-compliant cache engine software, either Memcached or Redis.

Request Syntax

```
response = client.create_cache_cluster(
    CacheClusterId='string',
    ReplicationGroupId='string',
```

```

AZMode='single-az'|'cross-az',
PreferredAvailabilityZone='string',
PreferredAvailabilityZones=[
    'string',
],
NumCacheNodes=123,
CacheNodeType='string',
Engine='string',
EngineVersion='string',
CacheParameterGroupName='string',
CacheSubnetGroupName='string',
CacheSecurityGroupNames=[
    'string',
],
SecurityGroupIds=[
    'string',
],
Tags=[
    {
        'Key': 'string',
        'Value': 'string'
    },
],
SnapshotArns=[
    'string',
],
SnapshotName='string',
PreferredMaintenanceWindow='string',
Port=123,
NotificationTopicArn='string',
AutoMinorVersionUpgrade=True|False,
SnapshotRetentionLimit=123,
SnapshotWindow='string'
)

```

Parameters

- **CacheClusterId** (*string*) – [REQUIRED]

The node group identifier. This parameter is stored as a lowercase string.

Constraints:

- A name must contain from 1 to 20 alphanumeric characters or hyphens.
 - The first character must be a letter.
 - A name cannot end with a hyphen or contain two consecutive hyphens.
- **ReplicationGroupId** (*string*) – The ID of the replication group to which this cache cluster should belong. If this parameter is specified, the cache cluster will be added to the specified replication group as a read replica; otherwise, the cache cluster will be a standalone primary that is not part of any replication group.

If the specified replication group is Multi-AZ enabled and the availability zone is not specified, the cache cluster will be created in availability zones that provide the best spread of read replicas across availability zones.

Note: This parameter is only valid if the `Engine` parameter is `redis`.

- **AZMode** (*string*) – Specifies whether the nodes in this Memcached node group are created in a single Availability Zone or created across multiple Availability Zones in the cluster's region.

This parameter is only supported for Memcached cache clusters.

If the `AZMode` and `PreferredAvailabilityZones` are not specified, `ElastiCache` assumes `single-az` mode.

- **PreferredAvailabilityZone** (*string*) – The EC2 Availability Zone in which the cache cluster will be created.

All nodes belonging to this Memcached cache cluster are placed in the preferred Availability Zone. If you want to create your nodes across multiple Availability Zones, use `PreferredAvailabilityZones`.

Default: System chosen Availability Zone.

- **PreferredAvailabilityZones** (*list*) – A list of the Availability Zones in which cache nodes will be created. The order of the zones in the list is not important.

This option is only supported on Memcached.

Note: If you are creating your cache cluster in an Amazon VPC (recommended) you can only locate nodes in Availability Zones that are associated with the subnets in the selected subnet group.

The number of Availability Zones listed must equal the value of `NumCacheNodes`.

If you want all the nodes in the same Availability Zone, use `PreferredAvailabilityZone` instead, or repeat the Availability Zone multiple times in the list.

Default: System chosen Availability Zones.

Example: One Memcached node in each of three different Availability Zones:

`PreferredAvailabilityZones.member.1=us-west-2aPreferredAvailabilityZone`

Example: All three Memcached nodes in one Availability Zone:

`PreferredAvailabilityZones.member.1=us-west-2aPreferredAvailabilityZone`

– (*string*) –

- **NumCacheNodes** (*integer*) – The initial number of cache nodes that the cache cluster will have.

For clusters running Redis, this value must be 1. For clusters running Memcached, this value must be between 1 and 20.

If you need more than 20 nodes for your Memcached cluster, please fill out the `ElastiCache Limit Increase Request` form at <http://aws.amazon.com/contact-us/elasticache-node-limit-request/>.

- **CacheNodeType** (*string*) – The compute and memory capacity of the nodes in the node group.

Valid node types are as follows:

- General purpose: * Current generation: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
- Previous generation: `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
- Compute optimized: `cache.c1.xlarge`
- Memory optimized * Current generation: `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`, `cache.r3.4xlarge`, `cache.r3.8xlarge`
- Previous generation: `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#).

- **Engine** (*string*) – The name of the cache engine to be used for this cache cluster.

Valid values for this parameter are:

`memcached | redis`

- **EngineVersion** (*string*) – The version number of the cache engine to be used for this cache cluster. To view the supported cache engine versions, use the `DescribeCacheEngineVersions` action.
- **CacheParameterGroupName** (*string*) – The name of the parameter group to associate with this cache cluster. If this argument is omitted, the default parameter group for the specified engine is used.
- **CacheSubnetGroupName** (*string*) – The name of the subnet group to be used for the cache cluster.

Use this parameter only when you are creating a cache cluster in an Amazon Virtual Private Cloud (VPC).

- **CacheSecurityGroupNames** (*list*) – A list of security group names to associate with this cache cluster.

Use this parameter only when you are creating a cache cluster outside of an Amazon Virtual Private Cloud (VPC).

– (*string*) –

- **SecurityGroupIds** (*list*) – One or more VPC security groups associated with the cache cluster.

Use this parameter only when you are creating a cache cluster in an Amazon Virtual Private Cloud (VPC).

– (*string*) –

- **Tags** (*list*) – A list of cost allocation tags to be added to this resource. A tag is a key-value pair. A tag key must be accompanied by a tag value.

– (*dict*) –

A cost allocation Tag that can be added to an ElastiCache cluster or replication group. Tags are composed of a Key/Value pair. A tag with a null Value is permitted.

* **Key** (*string*) –

The key for the tag.

* **Value** (*string*) –

The tag's value. May not be null.

- **SnapshotArns** (*list*) – A single-element string list containing an Amazon Resource Name (ARN) that uniquely identifies a Redis RDB snapshot file stored in Amazon S3. The snapshot file will be used to populate the node group. The Amazon S3 object name in the ARN cannot contain any commas.

Note: This parameter is only valid if the `Engine` parameter is `redis`.

Example of an Amazon S3 ARN: `arn:aws:s3:::my_bucket/snapshot1.rdb`

– (*string*) –

- **SnapshotName** (*string*) – The name of a snapshot from which to restore data into the new node group. The snapshot status changes to `restoring` while the

new node group is being created.

Note: This parameter is only valid if the `Engine` parameter is `redis`.

- **PreferredMaintenanceWindow** (*string*) – Specifies the weekly time range during which maintenance on the cache cluster is performed. It is specified as a range in the format `ddd:hh24:mi-ddd:hh24:mi` (24H Clock UTC). The minimum maintenance window is a 60 minute period. Valid values for `ddd` are:
 - `sun`
 - `mon`
 - `tue`
 - `wed`
 - `thu`
 - `fri`
 - `sat`

Example: `sun:05:00-sun:09:00`

- **Port** (*integer*) – The port number on which each of the cache nodes will accept connections.
- **NotificationTopicArn** (*string*) – The Amazon Resource Name (ARN) of the Amazon Simple Notification Service (SNS) topic to which notifications will be sent.

Note: The Amazon SNS topic owner must be the same as the cache cluster owner.

-
- **AutoMinorVersionUpgrade** (*boolean*) – This parameter is currently disabled.
 - **SnapshotRetentionLimit** (*integer*) – The number of days for which ElastiCache will retain automatic snapshots before deleting them. For example, if you set `SnapshotRetentionLimit` to `5`, then a snapshot that was taken today will be retained for 5 days before being deleted.

Note: This parameter is only valid if the `Engine` parameter is `redis`.

Default: 0 (i.e., automatic backups are disabled for this cache cluster).

- **SnapshotWindow** (*string*) – The daily time range (in UTC) during which ElastiCache will begin taking a daily snapshot of your node group.

Example: `05:00-09:00`

If you do not specify this parameter, then ElastiCache will automatically choose an appropriate time range.

Note: This parameter is only valid if the `Engine` parameter is `redis`.

Return type dict

Returns

Response Syntax

```
{
  'CacheCluster': {
    'CacheClusterId': 'string',
    'ConfigurationEndpoint': {
      'Address': 'string',
      'Port': 123
    },
    'ClientDownloadLandingPage': 'string',
    'CacheNodeType': 'string',
    'Engine': 'string',
    'EngineVersion': 'string',
```

```

'CacheClusterStatus': 'string',
'NumCacheNodes': 123,
'PreferredAvailabilityZone': 'string',
'CacheClusterCreateTime': datetime(2015, 1, 1),
'PreferredMaintenanceWindow': 'string',
'PendingModifiedValues': {
    'NumCacheNodes': 123,
    'CacheNodeIdsToRemove': [
        'string',
    ],
    'EngineVersion': 'string'
},
'NotificationConfiguration': {
    'TopicArn': 'string',
    'TopicStatus': 'string'
},
'CacheSecurityGroups': [
    {
        'CacheSecurityGroupName': 'string',
        'Status': 'string'
    },
],
'CacheParameterGroup': {
    'CacheParameterGroupName': 'string',
    'ParameterApplyStatus': 'string',
    'CacheNodeIdsToReboot': [
        'string',
    ]
},
'CacheSubnetGroupName': 'string',
'CacheNodes': [
    {
        'CacheNodeId': 'string',
        'CacheNodeStatus': 'string',
        'CacheNodeCreateTime': datetime(2015, 1, 1),
        'Endpoint': {
            'Address': 'string',
            'Port': 123
        },
        'ParameterGroupStatus': 'string',
        'SourceCacheNodeId': 'string',
        'CustomerAvailabilityZone': 'string'
    },
],
'AutoMinorVersionUpgrade': True|False,
'SecurityGroups': [
    {
        'SecurityGroupId': 'string',
        'Status': 'string'
    },
],
'ReplicationGroupId': 'string',
'SnapshotRetentionLimit': 123,
'SnapshotWindow': 'string'
}
}

```

Response Structure

- (*dict*) –

Contains all of the attributes of a specific cache cluster.

- **CacheCluster** (*dict*) –

Contains all of the attributes of a specific cache cluster.

- * **CacheClusterId** (*string*) –

The user-supplied identifier of the cache cluster. This identifier is a unique key that identifies a cache cluster.

- * **ConfigurationEndpoint** (*dict*) –

Represents the information required for client programs to connect to a cache node.

- **Address** (*string*) –

The DNS hostname of the cache node.

- **Port** (*integer*) –

The port number that the cache engine is listening on.

- * **ClientDownloadLandingPage** (*string*) –

The URL of the web page where you can download the latest ElastiCache client library.

- * **CacheNodeType** (*string*) –

The name of the compute and memory capacity node type for the cache cluster.

Valid node types are as follows:

- General purpose: * Current generation: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
- Previous generation: `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
- Compute optimized: `cache.c1.xlarge`
- Memory optimized * Current generation: `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`, `cache.r3.4xlarge`, `cache.r3.8xlarge`
- Previous generation: `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#).

- * **Engine** (*string*) –

The name of the cache engine (*memcached* or *redis*) to be used for this cache cluster.

- * **EngineVersion** (*string*) –

The version of the cache engine version that is used in this cache cluster.

* **CacheClusterStatus** (*string*) –

The current state of this cache cluster, one of the following values: *available* , *creating* , *deleted* , *deleting* , *incompatible-network* , *modifying* , *rebooting cache cluster nodes* , *restore-failed* , or *snapshotting* .

* **NumCacheNodes** (*integer*) –

The number of cache nodes in the cache cluster.

For clusters running Redis, this value must be 1. For clusters running Memcached, this value must be between 1 and 20.

* **PreferredAvailabilityZone** (*string*) –

The name of the Availability Zone in which the cache cluster is located or “Multiple” if the cache nodes are located in different Availability Zones.

* **CacheClusterCreateTime** (*datetime*) –

The date and time when the cache cluster was created.

* **PreferredMaintenanceWindow** (*string*) –

Specifies the weekly time range during which maintenance on the cache cluster is performed. It is specified as a range in the format ddd:hh24:mi-ddd:hh24:mi (24H Clock UTC). The minimum maintenance window is a 60 minute period. Valid values for ddd are:

- sun
- mon
- tue
- wed
- thu
- fri
- sat

Example: sun:05:00-sun:09:00

* **PendingModifiedValues** (*dict*) –

A group of settings that will be applied to the cache cluster in the future, or that are currently being applied.

- **NumCacheNodes** (*integer*) –

The new number of cache nodes for the cache cluster.

For clusters running Redis, this value must be 1. For clusters running Memcached, this value must be between 1 and 20.

- **CacheNodeIdsToRemove** (*list*) –

A list of cache node IDs that are being removed (or will be removed) from the cache cluster. A node ID is a numeric identifier (0001, 0002, etc.).

- (*string*) –

- **EngineVersion** (*string*) –

The new cache engine version that the cache cluster will run.

* **NotificationConfiguration** (*dict*) –

Describes a notification topic and its status. Notification topics are used for publishing ElastiCache events to subscribers using Amazon Simple Notification Service (SNS).

- **TopicArn** (*string*) –
The Amazon Resource Name (ARN) that identifies the topic.
- **TopicStatus** (*string*) –
The current state of the topic.
- * **CacheSecurityGroups** (*list*) –
A list of cache security group elements, composed of name and status sub-elements.
 - (*dict*) –
Represents a cache cluster's status within a particular cache security group.
 - **CacheSecurityGroupName** (*string*) –
The name of the cache security group.
 - **Status** (*string*) –
The membership status in the cache security group. The status changes when a cache security group is modified, or when the cache security groups assigned to a cache cluster are modified.
- * **CacheParameterGroup** (*dict*) –
The status of the cache parameter group.
 - **CacheParameterGroupName** (*string*) –
The name of the cache parameter group.
 - **ParameterApplyStatus** (*string*) –
The status of parameter updates.
 - **CacheNodeIdsToReboot** (*list*) –
A list of the cache node IDs which need to be rebooted for parameter changes to be applied. A node ID is a numeric identifier (0001, 0002, etc.).
 - (*string*) –
- * **CacheSubnetGroupName** (*string*) –
The name of the cache subnet group associated with the cache cluster.
- * **CacheNodes** (*list*) –
A list of cache nodes that are members of the cache cluster.
 - (*dict*) –
Represents an individual cache node within a cache cluster. Each cache node runs its own instance of the cluster's protocol-compliant caching software - either Memcached or Redis.

Valid node types are as follows:
 - General purpose: * Current generation: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
 - Previous generation: `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
 - Compute optimized: `cache.c1.xlarge`

- **Memory optimized** * **Current generation:**
`cache.r3.large` , `cache.r3.xlarge` ,
`cache.r3.2xlarge` , `cache.r3.4xlarge` ,
`cache.r3.8xlarge`
- **Previous generation:** `cache.m2.xlarge` ,
`cache.m2.2xlarge` , `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#) .

- **CacheNodeId** (*string*) –

The cache node identifier. A node ID is a numeric identifier (0001, 0002, etc.). The combination of cluster ID and node ID uniquely identifies every cache node used in a customer's AWS account.

- **CacheNodeStatus** (*string*) –

The current state of this cache node.

- **CacheNodeCreateTime** (*datetime*) –

The date and time when the cache node was created.

- **Endpoint** (*dict*) –

The hostname for connecting to this cache node.

- **Address** (*string*) –

The DNS hostname of the cache node.

- **Port** (*integer*) –

The port number that the cache engine is listening on.

- **ParameterGroupStatus** (*string*) –

The status of the parameter group applied to this cache node.

- **SourceCacheNodeId** (*string*) –

The ID of the primary node to which this read replica node is synchronized. If this field is empty, then this node is not associated with a primary cache cluster.

- **CustomerAvailabilityZone** (*string*) –

The Availability Zone where this node was created and now resides.

- * **AutoMinorVersionUpgrade** (*boolean*) –

This parameter is currently disabled.

- * **SecurityGroups** (*list*) –

A list of VPC Security Groups associated with the cache cluster.

- (*dict*) –

Represents a single cache security group and its status.

- **SecurityGroupId** (*string*) –

The identifier of the cache security group.

- **Status** (*string*) –

The status of the cache security group membership. The status changes whenever a cache security group is modified, or when the cache security groups assigned to a cache cluster are modified.

- * **ReplicationGroupId** (*string*) –

The replication group to which this cache cluster belongs. If this field is empty, the cache cluster is not associated with any replication group.

- * **SnapshotRetentionLimit** (*integer*) –

The number of days for which ElastiCache will retain automatic cache cluster snapshots before deleting them. For example, if you set *SnapshotRetentionLimit* to 5, then a snapshot that was taken today will be retained for 5 days before being deleted.

Important If the value of *SnapshotRetentionLimit* is set to zero (0), backups are turned off.

- * **SnapshotWindow** (*string*) –

The daily time range (in UTC) during which ElastiCache will begin taking a daily snapshot of your cache cluster.

Example: 05:00-09:00

create_cache_parameter_group (***kwargs*)

The *CreateCacheParameterGroup* action creates a new cache parameter group. A cache parameter group is a collection of parameters that you apply to all of the nodes in a cache cluster.

Request Syntax

```
response = client.create_cache_parameter_group(
    CacheParameterGroupName='string',
    CacheParameterGroupFamily='string',
    Description='string'
)
```

Parameters

- **CacheParameterGroupName** (*string*) – [REQUIRED]

A user-specified name for the cache parameter group.

- **CacheParameterGroupFamily** (*string*) – [REQUIRED]

The name of the cache parameter group family the cache parameter group can be used with.

Valid values are: memcached1.4 | redis2.6 | redis2.8

- **Description** (*string*) – [REQUIRED]

A user-specified description for the cache parameter group.

Return type dict

Returns

Response Syntax

```
{
    'CacheParameterGroup': {
        'CacheParameterGroupName': 'string',
```



```

        'CacheParameterGroupFamily': 'string',
        'Description': 'string'
    }
}

```

Response Structure

- (dict) –

Represents the output of a *CreateCacheParameterGroup* action.

- **CacheParameterGroup** (dict) –

Represents the output of a *CreateCacheParameterGroup* action.

- * **CacheParameterGroupName** (string) –

The name of the cache parameter group.

- * **CacheParameterGroupFamily** (string) –

The name of the cache parameter group family that this cache parameter group is compatible with.

- * **Description** (string) –

The description for this cache parameter group.

create_cache_security_group (**kwargs)

The *CreateCacheSecurityGroup* action creates a new cache security group. Use a cache security group to control access to one or more cache clusters.

Cache security groups are only used when you are creating a cache cluster outside of an Amazon Virtual Private Cloud (VPC). If you are creating a cache cluster inside of a VPC, use a cache subnet group instead. For more information, see [CreateCacheSubnetGroup](#).

Request Syntax

```

response = client.create_cache_security_group(
    CacheSecurityGroupName='string',
    Description='string'
)

```

Parameters

- **CacheSecurityGroupName** (string) – [REQUIRED]

A name for the cache security group. This value is stored as a lowercase string.

Constraints: Must contain no more than 255 alphanumeric characters. Cannot be the word “Default”.

Example: mysecuritygroup

- **Description** (string) – [REQUIRED]

A description for the cache security group.

Return type dict

Returns

Response Syntax

```

{
    'CacheSecurityGroup': {
        'OwnerId': 'string',
        'CacheSecurityGroupName': 'string',
        'Description': 'string',
        'EC2SecurityGroups': [

```

```
        {
            'Status': 'string',
            'EC2SecurityGroupName': 'string',
            'EC2SecurityGroupOwnerId': 'string'
        },
    ]
}
```

Response Structure

- (*dict*) –

Represents the output of one of the following actions:

- *AuthorizeCacheSecurityGroupIngress*
- *CreateCacheSecurityGroup*
- *RevokeCacheSecurityGroupIngress*
- **CacheSecurityGroup** (*dict*) –

Represents the output of one of the following actions:

- * *AuthorizeCacheSecurityGroupIngress*
- * *CreateCacheSecurityGroup*
- * *RevokeCacheSecurityGroupIngress*
- * **OwnerId** (*string*) –

The AWS account ID of the cache security group owner.

- * **CacheSecurityGroupName** (*string*) –

The name of the cache security group.

- * **Description** (*string*) –

The description of the cache security group.

- * **EC2SecurityGroups** (*list*) –

A list of Amazon EC2 security groups that are associated with this cache security group.

- (*dict*) –

Provides ownership and status information for an Amazon EC2 security group.

- **Status** (*string*) –

The status of the Amazon EC2 security group.

- **EC2SecurityGroupName** (*string*) –

The name of the Amazon EC2 security group.

- **EC2SecurityGroupOwnerId** (*string*) –

The AWS account ID of the Amazon EC2 security group owner.

create_cache_subnet_group (***kwargs*)

The *CreateCacheSubnetGroup* action creates a new cache subnet group.

Use this parameter only when you are creating a cluster in an Amazon Virtual Private Cloud (VPC).

Request Syntax

```
response = client.create_cache_subnet_group(
    CacheSubnetGroupName='string',
    CacheSubnetGroupDescription='string',
```

```

SubnetIds=[
    'string',
]
)

```

Parameters

- **CacheSubnetGroupName** (*string*) – [REQUIRED]

A name for the cache subnet group. This value is stored as a lowercase string.

Constraints: Must contain no more than 255 alphanumeric characters or hyphens.

Example: mysubnetgroup

- **CacheSubnetGroupDescription** (*string*) – [REQUIRED]

A description for the cache subnet group.

- **SubnetIds** (*list*) – [REQUIRED]

A list of VPC subnet IDs for the cache subnet group.

– (*string*) –

Return type dict

Returns

Response Syntax

```

{
    'CacheSubnetGroup': {
        'CacheSubnetGroupName': 'string',
        'CacheSubnetGroupDescription': 'string',
        'VpcId': 'string',
        'Subnets': [
            {
                'SubnetIdentifier': 'string',
                'SubnetAvailabilityZone': {
                    'Name': 'string'
                }
            }
        ],
    },
}

```

Response Structure

- (*dict*) –

Represents the output of one of the following actions:

- *CreateCacheSubnetGroup*
- *ModifyCacheSubnetGroup*
- **CacheSubnetGroup** (*dict*) –

Represents the output of one of the following actions:

- * *CreateCacheSubnetGroup*
- * *ModifyCacheSubnetGroup*
- * **CacheSubnetGroupName** (*string*) –
The name of the cache subnet group.
- * **CacheSubnetGroupDescription** (*string*) –
The description of the cache subnet group.

* **VpcId** (*string*) –

The Amazon Virtual Private Cloud identifier (VPC ID) of the cache subnet group.

* **Subnets** (*list*) –

A list of subnets associated with the cache subnet group.

· (*dict*) –

Represents the subnet associated with a cache cluster. This parameter refers to subnets defined in Amazon Virtual Private Cloud (Amazon VPC) and used with ElastiCache.

· **SubnetIdentifier** (*string*) –

The unique identifier for the subnet.

· **SubnetAvailabilityZone** (*dict*) –

The Availability Zone associated with the subnet.

· **Name** (*string*) –

The name of the Availability Zone.

create_replication_group (***kwargs*)

The *CreateReplicationGroup* action creates a replication group. A replication group is a collection of cache clusters, where one of the cache clusters is a read/write primary and the others are read-only replicas. Writes to the primary are automatically propagated to the replicas.

When you create a replication group, you must specify an existing cache cluster that is in the primary role. When the replication group has been successfully created, you can add one or more read replica replicas to it, up to a total of five read replicas.

Note: This action is valid only for Redis.

Request Syntax

```
response = client.create_replication_group(
    ReplicationGroupId='string',
    ReplicationGroupDescription='string',
    PrimaryClusterId='string',
    AutomaticFailoverEnabled=True|False,
    NumCacheClusters=123,
    PreferredCacheClusterAZs=[
        'string',
    ],
    CacheNodeType='string',
    Engine='string',
    EngineVersion='string',
    CacheParameterGroupName='string',
    CacheSubnetGroupName='string',
    CacheSecurityGroupNames=[
        'string',
    ],
    SecurityGroupIds=[
        'string',
    ],
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
)
```

```

SnapshotArns=[
    'string',
],
SnapshotName='string',
PreferredMaintenanceWindow='string',
Port=123,
NotificationTopicArn='string',
AutoMinorVersionUpgrade=True|False,
SnapshotRetentionLimit=123,
SnapshotWindow='string'
)

```

Parameters

- **ReplicationGroupId** (*string*) – [REQUIRED]

The replication group identifier. This parameter is stored as a lowercase string.

Constraints:

- A name must contain from 1 to 20 alphanumeric characters or hyphens.
- The first character must be a letter.
- A name cannot end with a hyphen or contain two consecutive hyphens.

- **ReplicationGroupDescription** (*string*) – [REQUIRED]

A user-created description for the replication group.

- **PrimaryClusterId** (*string*) – The identifier of the cache cluster that will serve as the primary for this replication group. This cache cluster must already exist and have a status of *available*.

This parameter is not required if *NumCacheClusters* is specified.

- **AutomaticFailoverEnabled** (*boolean*) – Specifies whether a read-only replica will be automatically promoted to read/write primary if the existing primary fails.

If `true`, Multi-AZ is enabled for this replication group. If `false`, Multi-AZ is disabled for this replication group.

Default: `false`

Note: ElastiCache Multi-AZ replication groups is not supported on:

- Redis versions earlier than 2.8.6.
 - T1 and T2 cache node types.
-

- **NumCacheClusters** (*integer*) – The number of cache clusters this replication group will initially have.

If *Multi-AZ* is `enabled`, the value of this parameter must be at least 2.

The maximum permitted value for *NumCacheClusters* is 6 (primary plus 5 replicas). If you need to exceed this limit, please fill out the ElastiCache Limit Increase Request form at <http://aws.amazon.com/contact-us/elasticache-node-limit-request>.

- **PreferredCacheClusterAZs** (*list*) – A list of EC2 availability zones in which the replication group's cache clusters will be created. The order of the availability zones in the list is not important.

Note: If you are creating your replication group in an Amazon VPC (recommended), you can only locate cache clusters in availability zones associated with the subnets in the selected subnet group.

The number of availability zones listed must equal the value of *NumCacheClus-*

ters .

Default: system chosen availability zones.

Example: One Redis cache cluster in each of three availability zones. PreferredAvailabilityZones.member.1=us-west-2a PreferredAvailabilityZones.member.2=us-west-2c PreferredAvailabilityZones.member.3=us-west-2c
– (string) –

- **CacheNodeType** (string) – The compute and memory capacity of the nodes in the node group.

Valid node types are as follows:

- General purpose: * Current generation: cache.t2.micro , cache.t2.small , cache.t2.medium , cache.m3.medium , cache.m3.large , cache.m3.xlarge , cache.m3.2xlarge
- Previous generation: cache.t1.micro , cache.m1.small , cache.m1.medium , cache.m1.large , cache.m1.xlarge
- Compute optimized: cache.c1.xlarge
- Memory optimized * Current generation: cache.r3.large , cache.r3.xlarge , cache.r3.2xlarge , cache.r3.4xlarge , cache.r3.8xlarge
- Previous generation: cache.m2.xlarge , cache.m2.2xlarge , cache.m2.4xlarge

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached or Cache Node Type-Specific Parameters for Redis](#) .

- **Engine** (string) – The name of the cache engine to be used for the cache clusters in this replication group.

Default: redis

- **EngineVersion** (string) – The version number of the cache engine to be used for the cache clusters in this replication group. To view the supported cache engine versions, use the *DescribeCacheEngineVersions* action.
- **CacheParameterGroupName** (string) – The name of the parameter group to associate with this replication group. If this argument is omitted, the default cache parameter group for the specified engine is used.
- **CacheSubnetGroupName** (string) – The name of the cache subnet group to be used for the replication group.
- **CacheSecurityGroupNames** (list) – A list of cache security group names to associate with this replication group.
– (string) –
- **SecurityGroupIds** (list) – One or more Amazon VPC security groups associated with this replication group.

Use this parameter only when you are creating a replication group in an Amazon Virtual Private Cloud (VPC).

- (string) –
- **Tags** (list) – A list of cost allocation tags to be added to this resource. A tag is a key-value pair. A tag key must be accompanied by a tag value.
– (dict) –

A cost allocation Tag that can be added to an ElastiCache cluster or replication group. Tags are composed of a Key/Value pair. A tag with a null Value is permitted.

* **Key** (*string*) –

The key for the tag.

* **Value** (*string*) –

The tag's value. May not be null.

- **SnapshotArns** (*list*) – A single-element string list containing an Amazon Resource Name (ARN) that uniquely identifies a Redis RDB snapshot file stored in Amazon S3. The snapshot file will be used to populate the node group. The Amazon S3 object name in the ARN cannot contain any commas.

Note: This parameter is only valid if the `Engine` parameter is `redis`.

Example of an Amazon S3 ARN: `arn:aws:s3:::my_bucket/snapshot1.rdb`
– (*string*) –

- **SnapshotName** (*string*) – The name of a snapshot from which to restore data into the new node group. The snapshot status changes to `restoring` while the new node group is being created.

Note: This parameter is only valid if the `Engine` parameter is `redis`.

- **PreferredMaintenanceWindow** (*string*) – Specifies the weekly time range during which maintenance on the cache cluster is performed. It is specified as a range in the format `ddd:hh24:mi-ddd:hh24:mi` (24H Clock UTC). The minimum maintenance window is a 60 minute period. Valid values for `ddd` are:

- `sun`
- `mon`
- `tue`
- `wed`
- `thu`
- `fri`
- `sat`

Example: `sun:05:00-sun:09:00`

- **Port** (*integer*) – The port number on which each member of the replication group will accept connections.
- **NotificationTopicArn** (*string*) – The Amazon Resource Name (ARN) of the Amazon Simple Notification Service (SNS) topic to which notifications will be sent.

Note: The Amazon SNS topic owner must be the same as the cache cluster owner.

- **AutoMinorVersionUpgrade** (*boolean*) – This parameter is currently disabled.
- **SnapshotRetentionLimit** (*integer*) – The number of days for which ElastiCache will retain automatic snapshots before deleting them. For example, if you set `SnapshotRetentionLimit` to 5, then a snapshot that was taken today will be retained for 5 days before being deleted.

Note: This parameter is only valid if the `Engine` parameter is `redis`.

Default: 0 (i.e., automatic backups are disabled for this cache cluster).

- **SnapshotWindow** (*string*) – The daily time range (in UTC) during which ElastiCache will begin taking a daily snapshot of your node group.

Example: `05:00-09:00`

If you do not specify this parameter, then ElastiCache will automatically choose an appropriate time range.

Note: This parameter is only valid if the `Engine` parameter is `redis`.

Return type dict

Returns

Response Syntax

```
{
  'ReplicationGroup': {
    'ReplicationGroupId': 'string',
    'Description': 'string',
    'Status': 'string',
    'PendingModifiedValues': {
      'PrimaryClusterId': 'string',
      'AutomaticFailoverStatus': 'enabled'|'disabled'
    },
    'MemberClusters': [
      'string',
    ],
    'NodeGroups': [
      {
        'NodeGroupId': 'string',
        'Status': 'string',
        'PrimaryEndpoint': {
          'Address': 'string',
          'Port': 123
        },
        'NodeGroupMembers': [
          {
            'CacheClusterId': 'string',
            'CacheNodeId': 'string',
            'ReadEndpoint': {
              'Address': 'string',
              'Port': 123
            },
            'PreferredAvailabilityZone': 'string',
            'CurrentRole': 'string'
          },
        ],
      },
    ],
    'SnapshottingClusterId': 'string',
    'AutomaticFailover': 'enabled'|'disabled'|'enabling'|'disabling'
  }
}
```

Response Structure

- (*dict*) –

Contains all of the attributes of a specific replication group.

- **ReplicationGroup** (*dict*) –

Contains all of the attributes of a specific replication group.

- * **ReplicationGroupId** (*string*) –

The identifier for the replication group.

- * **Description** (*string*) –

The description of the replication group.

* **Status** (*string*) –

The current state of this replication group - *creating* , *available* , etc.

* **PendingModifiedValues** (*dict*) –

A group of settings to be applied to the replication group, either immediately or during the next maintenance window.

· **PrimaryClusterId** (*string*) –

The primary cluster ID which will be applied immediately (if `--apply-immediately` was specified), or during the next maintenance window.

· **AutomaticFailoverStatus** (*string*) –

Indicates the status of Multi-AZ for this replication group.

Note: ElastiCache Multi-AZ replication groups are not supported on:

- Redis versions earlier than 2.8.6.
 - T1 and T2 cache node types.
-

* **MemberClusters** (*list*) –

The names of all the cache clusters that are part of this replication group.

· (*string*) –

* **NodeGroups** (*list*) –

A single element list with information about the nodes in the replication group.

· (*dict*) –

Represents a collection of cache nodes in a replication group.

· **NodeGroupId** (*string*) –

The identifier for the node group. A replication group contains only one node group; therefore, the node group ID is 0001.

· **Status** (*string*) –

The current state of this replication group - *creating* , *available* , etc.

· **PrimaryEndpoint** (*dict*) –

Represents the information required for client programs to connect to a cache node.

· **Address** (*string*) –

The DNS hostname of the cache node.

· **Port** (*integer*) –

The port number that the cache engine is listening on.

· **NodeGroupMembers** (*list*) –

A list containing information about individual nodes within the node group.

· (*dict*) –

Represents a single node within a node group.

· **CacheClusterId** (*string*) –

The ID of the cache cluster to which the node belongs.

- **CacheNodeId** (*string*) –

The ID of the node within its cache cluster. A node ID is a numeric identifier (0001, 0002, etc.).

- **ReadEndpoint** (*dict*) –

Represents the information required for client programs to connect to a cache node.

- **Address** (*string*) –

The DNS hostname of the cache node.

- **Port** (*integer*) –

The port number that the cache engine is listening on.

- **PreferredAvailabilityZone** (*string*) –

The name of the Availability Zone in which the node is located.

- **CurrentRole** (*string*) –

The role that is currently assigned to the node - *primary* or *replica*.

- * **SnapshottingClusterId** (*string*) –

The cache cluster ID that is used as the daily snapshot source for the replication group.

- * **AutomaticFailover** (*string*) –

Indicates the status of Multi-AZ for this replication group.

Note: ElastiCache Multi-AZ replication groups are not supported on:

- Redis versions earlier than 2.8.6.
 - T1 and T2 cache node types.
-

create_snapshot (***kwargs*)

The *CreateSnapshot* action creates a copy of an entire cache cluster at a specific moment in time.

Request Syntax

```
response = client.create_snapshot(  
    CacheClusterId='string',  
    SnapshotName='string'  
)
```

Parameters

- **CacheClusterId** (*string*) – [REQUIRED]

The identifier of an existing cache cluster. The snapshot will be created from this cache cluster.

- **SnapshotName** (*string*) – [REQUIRED]

A name for the snapshot being created.

Return type dict

Returns

Response Syntax

```

{
  'Snapshot': {
    'SnapshotName': 'string',
    'CacheClusterId': 'string',
    'SnapshotStatus': 'string',
    'SnapshotSource': 'string',
    'CacheNodeType': 'string',
    'Engine': 'string',
    'EngineVersion': 'string',
    'NumCacheNodes': 123,
    'PreferredAvailabilityZone': 'string',
    'CacheClusterCreateTime': datetime(2015, 1, 1),
    'PreferredMaintenanceWindow': 'string',
    'TopicArn': 'string',
    'Port': 123,
    'CacheParameterGroupName': 'string',
    'CacheSubnetGroupName': 'string',
    'VpcId': 'string',
    'AutoMinorVersionUpgrade': True|False,
    'SnapshotRetentionLimit': 123,
    'SnapshotWindow': 'string',
    'NodeSnapshots': [
      {
        'CacheNodeId': 'string',
        'CacheSize': 'string',
        'CacheNodeCreateTime': datetime(2015, 1, 1),
        'SnapshotCreateTime': datetime(2015, 1, 1)
      },
    ]
  }
}

```

Response Structure

- (*dict*) –

Represents a copy of an entire cache cluster as of the time when the snapshot was taken.

- **Snapshot** (*dict*) –

Represents a copy of an entire cache cluster as of the time when the snapshot was taken.

- * **SnapshotName** (*string*) –

The name of a snapshot. For an automatic snapshot, the name is system-generated; for a manual snapshot, this is the user-provided name.

- * **CacheClusterId** (*string*) –

The user-supplied identifier of the source cache cluster.

- * **SnapshotStatus** (*string*) –

The status of the snapshot. Valid values: `creating`|`available`|`restoring`|`copying`|`deleting`.

- * **SnapshotSource** (*string*) –

Indicates whether the snapshot is from an automatic backup (`automated`) or was created manually (`manual`).

- * **CacheNodeType** (*string*) –

The name of the compute and memory capacity node type for the

source cache cluster.

Valid node types are as follows:

- **General purpose:** * **Current generation:** `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
- **Previous generation:** `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
- **Compute optimized:** `cache.c1.xlarge`
- **Memory optimized** * **Current generation:** `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`, `cache.r3.4xlarge`, `cache.r3.8xlarge`
- **Previous generation:** `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#).

* **Engine** (*string*) –

The name of the cache engine (*memcached* or *redis*) used by the source cache cluster.

* **EngineVersion** (*string*) –

The version of the cache engine version that is used by the source cache cluster.

* **NumCacheNodes** (*integer*) –

The number of cache nodes in the source cache cluster.

For clusters running Redis, this value must be 1. For clusters running Memcached, this value must be between 1 and 20.

* **PreferredAvailabilityZone** (*string*) –

The name of the Availability Zone in which the source cache cluster is located.

* **CacheClusterCreateTime** (*datetime*) –

The date and time when the source cache cluster was created.

* **PreferredMaintenanceWindow** (*string*) –

Specifies the weekly time range during which maintenance on the cache cluster is performed. It is specified as a range in the format `ddd:hh24:mi-ddd:hh24:mi` (24H Clock UTC). The minimum maintenance window is a 60 minute period. Valid values for `ddd` are:

- `sun`
- `mon`
- `tue`
- `wed`

- `thu`
- `fri`
- `sat`

Example: `sun:05:00-sun:09:00`

* **TopicArn** (*string*) –

The Amazon Resource Name (ARN) for the topic used by the source cache cluster for publishing notifications.

* **Port** (*integer*) –

The port number used by each cache nodes in the source cache cluster.

* **CacheParameterGroupName** (*string*) –

The cache parameter group that is associated with the source cache cluster.

* **CacheSubnetGroupName** (*string*) –

The name of the cache subnet group associated with the source cache cluster.

* **VpcId** (*string*) –

The Amazon Virtual Private Cloud identifier (VPC ID) of the cache subnet group for the source cache cluster.

* **AutoMinorVersionUpgrade** (*boolean*) –

This parameter is currently disabled.

* **SnapshotRetentionLimit** (*integer*) –

For an automatic snapshot, the number of days for which ElastiCache will retain the snapshot before deleting it.

For manual snapshots, this field reflects the *SnapshotRetentionLimit* for the source cache cluster when the snapshot was created. This field is otherwise ignored: Manual snapshots do not expire, and can only be deleted using the *DeleteSnapshot* action.

Important If the value of *SnapshotRetentionLimit* is set to zero (0), backups are turned off.

* **SnapshotWindow** (*string*) –

The daily time range during which ElastiCache takes daily snapshots of the source cache cluster.

* **NodeSnapshots** (*list*) –

A list of the cache nodes in the source cache cluster.

- (*dict*) –

Represents an individual cache node in a snapshot of a cache cluster.

- **CacheNodeId** (*string*) –

The cache node identifier for the node in the source cache cluster.

- **CacheSize** (*string*) –

The size of the cache on the source cache node.

- **CacheNodeCreateTime** (*datetime*) –

The date and time when the cache node was created in the source cache cluster.

- **SnapshotCreateTime** (*datetime*) –

The date and time when the source node's metadata and cache data set was obtained for the snapshot.

delete_cache_cluster (***kwargs*)

The *DeleteCacheCluster* action deletes a previously provisioned cache cluster. *DeleteCacheCluster* deletes all associated cache nodes, node endpoints and the cache cluster itself. When you receive a successful response from this action, Amazon ElastiCache immediately begins deleting the cache cluster; you cannot cancel or revert this action.

This API cannot be used to delete a cache cluster that is the last read replica of a replication group that has Multi-AZ mode enabled.

Request Syntax

```
response = client.delete_cache_cluster(
    CacheClusterId='string',
    FinalSnapshotIdentifier='string'
)
```

Parameters

- **CacheClusterId** (*string*) – **[REQUIRED]**

The cache cluster identifier for the cluster to be deleted. This parameter is not case sensitive.

- **FinalSnapshotIdentifier** (*string*) – The user-supplied name of a final cache cluster snapshot. This is the unique name that identifies the snapshot. ElastiCache creates the snapshot, and then deletes the cache cluster immediately afterward.

Return type dict

Returns

Response Syntax

```
{
    'CacheCluster': {
        'CacheClusterId': 'string',
        'ConfigurationEndpoint': {
            'Address': 'string',
            'Port': 123
        },
        'ClientDownloadLandingPage': 'string',
        'CacheNodeType': 'string',
        'Engine': 'string',
        'EngineVersion': 'string',
        'CacheClusterStatus': 'string',
        'NumCacheNodes': 123,
        'PreferredAvailabilityZone': 'string',
        'CacheClusterCreateTime': datetime(2015, 1, 1),
        'PreferredMaintenanceWindow': 'string',
        'PendingModifiedValues': {
            'NumCacheNodes': 123,
            'CacheNodeIdsToRemove': [
                'string',
            ],
            'EngineVersion': 'string'
        },
        'NotificationConfiguration': {
            'TopicArn': 'string',
```

```

        'TopicStatus': 'string'
    },
    'CacheSecurityGroups': [
        {
            'CacheSecurityGroupName': 'string',
            'Status': 'string'
        },
    ],
    'CacheParameterGroup': {
        'CacheParameterGroupName': 'string',
        'ParameterApplyStatus': 'string',
        'CacheNodeIdsToReboot': [
            'string',
        ]
    },
    'CacheSubnetGroupName': 'string',
    'CacheNodes': [
        {
            'CacheNodeId': 'string',
            'CacheNodeStatus': 'string',
            'CacheNodeCreateTime': datetime(2015, 1, 1),
            'Endpoint': {
                'Address': 'string',
                'Port': 123
            },
            'ParameterGroupStatus': 'string',
            'SourceCacheNodeId': 'string',
            'CustomerAvailabilityZone': 'string'
        },
    ],
    'AutoMinorVersionUpgrade': True|False,
    'SecurityGroups': [
        {
            'SecurityGroupId': 'string',
            'Status': 'string'
        },
    ],
    'ReplicationGroupId': 'string',
    'SnapshotRetentionLimit': 123,
    'SnapshotWindow': 'string'
}

```

Response Structure

- (*dict*) –

Contains all of the attributes of a specific cache cluster.

– **CacheCluster** (*dict*) –

Contains all of the attributes of a specific cache cluster.

* **CacheClusterId** (*string*) –

The user-supplied identifier of the cache cluster. This identifier is a unique key that identifies a cache cluster.

* **ConfigurationEndpoint** (*dict*) –

Represents the information required for client programs to connect to a cache node.

· **Address** (*string*) –

The DNS hostname of the cache node.

· **Port** (*integer*) –

The port number that the cache engine is listening on.

* **ClientDownloadLandingPage** (*string*) –

The URL of the web page where you can download the latest ElastiCache client library.

* **CacheNodeType** (*string*) –

The name of the compute and memory capacity node type for the cache cluster.

Valid node types are as follows:

- General purpose: * Current generation: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
- Previous generation: `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
- Compute optimized: `cache.c1.xlarge`
- Memory optimized * Current generation: `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`, `cache.r3.4xlarge`, `cache.r3.8xlarge`
- Previous generation: `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#).

* **Engine** (*string*) –

The name of the cache engine (*memcached* or *redis*) to be used for this cache cluster.

* **EngineVersion** (*string*) –

The version of the cache engine version that is used in this cache cluster.

* **CacheClusterStatus** (*string*) –

The current state of this cache cluster, one of the following values: *available*, *creating*, *deleted*, *deleting*, *incompatible-network*, *modifying*, *rebooting cache cluster nodes*, *restore-failed*, or *snapshotting*.

* **NumCacheNodes** (*integer*) –

The number of cache nodes in the cache cluster.

For clusters running Redis, this value must be 1. For clusters running Memcached, this value must be between 1 and 20.

* **PreferredAvailabilityZone** (*string*) –

The name of the Availability Zone in which the cache cluster is located or “Multiple” if the cache nodes are located in different Availability Zones.

* **CacheClusterCreateTime** (*datetime*) –

The date and time when the cache cluster was created.

* **PreferredMaintenanceWindow** (*string*) –

Specifies the weekly time range during which maintenance on the cache cluster is performed. It is specified as a range in the format ddd:hh24:mi-ddd:hh24:mi (24H Clock UTC). The minimum maintenance window is a 60 minute period. Valid values for ddd are:

- sun
- mon
- tue
- wed
- thu
- fri
- sat

Example: sun:05:00-sun:09:00

* **PendingModifiedValues** (*dict*) –

A group of settings that will be applied to the cache cluster in the future, or that are currently being applied.

· **NumCacheNodes** (*integer*) –

The new number of cache nodes for the cache cluster.

For clusters running Redis, this value must be 1. For clusters running Memcached, this value must be between 1 and 20.

· **CacheNodeIdsToRemove** (*list*) –

A list of cache node IDs that are being removed (or will be removed) from the cache cluster. A node ID is a numeric identifier (0001, 0002, etc.).

· (*string*) –

· **EngineVersion** (*string*) –

The new cache engine version that the cache cluster will run.

* **NotificationConfiguration** (*dict*) –

Describes a notification topic and its status. Notification topics are used for publishing ElastiCache events to subscribers using Amazon Simple Notification Service (SNS).

· **TopicArn** (*string*) –

The Amazon Resource Name (ARN) that identifies the topic.

· **TopicStatus** (*string*) –

The current state of the topic.

* **CacheSecurityGroups** (*list*) –

A list of cache security group elements, composed of name and status sub-elements.

· (*dict*) –

Represents a cache cluster’s status within a particular cache security group.

- **CacheSecurityGroupName** (*string*) –

The name of the cache security group.

- **Status** (*string*) –

The membership status in the cache security group. The status changes when a cache security group is modified, or when the cache security groups assigned to a cache cluster are modified.

- * **CacheParameterGroup** (*dict*) –

The status of the cache parameter group.

- **CacheParameterGroupName** (*string*) –

The name of the cache parameter group.

- **ParameterApplyStatus** (*string*) –

The status of parameter updates.

- **CacheNodeIdsToReboot** (*list*) –

A list of the cache node IDs which need to be rebooted for parameter changes to be applied. A node ID is a numeric identifier (0001, 0002, etc.).

- (*string*) –

- * **CacheSubnetGroupName** (*string*) –

The name of the cache subnet group associated with the cache cluster.

- * **CacheNodes** (*list*) –

A list of cache nodes that are members of the cache cluster.

- (*dict*) –

Represents an individual cache node within a cache cluster. Each cache node runs its own instance of the cluster's protocol-compliant caching software - either Memcached or Redis.

Valid node types are as follows:

- General purpose: * Current generation: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
- Previous generation: `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
- Compute optimized: `cache.c1.xlarge`
- Memory optimized * Current generation: `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`, `cache.r3.4xlarge`, `cache.r3.8xlarge`
- Previous generation: `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#).

- **CacheNodeId** (*string*) –

The cache node identifier. A node ID is a numeric identifier (0001, 0002, etc.). The combination of cluster ID and node ID uniquely identifies every cache node used in a customer's AWS account.

- **CacheNodeStatus** (*string*) –

The current state of this cache node.

- **CacheNodeCreateTime** (*datetime*) –

The date and time when the cache node was created.

- **Endpoint** (*dict*) –

The hostname for connecting to this cache node.

- **Address** (*string*) –

The DNS hostname of the cache node.

- **Port** (*integer*) –

The port number that the cache engine is listening on.

- **ParameterGroupStatus** (*string*) –

The status of the parameter group applied to this cache node.

- **SourceCacheNodeId** (*string*) –

The ID of the primary node to which this read replica node is synchronized. If this field is empty, then this node is not associated with a primary cache cluster.

- **CustomerAvailabilityZone** (*string*) –

The Availability Zone where this node was created and now resides.

- * **AutoMinorVersionUpgrade** (*boolean*) –

This parameter is currently disabled.

- * **SecurityGroups** (*list*) –

A list of VPC Security Groups associated with the cache cluster.

- (*dict*) –

Represents a single cache security group and its status.

- **SecurityGroupId** (*string*) –

The identifier of the cache security group.

- **Status** (*string*) –

The status of the cache security group membership. The status changes whenever a cache security group is modified, or when the cache security groups assigned to a cache cluster are modified.

- * **ReplicationGroupId** (*string*) –

The replication group to which this cache cluster belongs. If this field is empty, the cache cluster is not associated with any replication group.

* **SnapshotRetentionLimit** (*integer*) –

The number of days for which ElastiCache will retain automatic cache cluster snapshots before deleting them. For example, if you set *SnapshotRetentionLimit* to 5, then a snapshot that was taken today will be retained for 5 days before being deleted.

Important If the value of *SnapshotRetentionLimit* is set to zero (0), backups are turned off.

* **SnapshotWindow** (*string*) –

The daily time range (in UTC) during which ElastiCache will begin taking a daily snapshot of your cache cluster.

Example: 05:00-09:00

delete_cache_parameter_group (***kwargs*)

The *DeleteCacheParameterGroup* action deletes the specified cache parameter group. You cannot delete a cache parameter group if it is associated with any cache clusters.

Request Syntax

```
response = client.delete_cache_parameter_group(
    CacheParameterGroupName='string'
)
```

Parameters **CacheParameterGroupName** (*string*) – [REQUIRED]

The name of the cache parameter group to delete.

Note: The specified cache security group must not be associated with any cache clusters.

Returns None

delete_cache_security_group (***kwargs*)

The *DeleteCacheSecurityGroup* action deletes a cache security group.

Note: You cannot delete a cache security group if it is associated with any cache clusters.

Request Syntax

```
response = client.delete_cache_security_group(
    CacheSecurityGroupName='string'
)
```

Parameters **CacheSecurityGroupName** (*string*) – [REQUIRED]

The name of the cache security group to delete.

Note: You cannot delete the default security group.

Returns None

delete_cache_subnet_group (***kwargs*)

The *DeleteCacheSubnetGroup* action deletes a cache subnet group.

Note: You cannot delete a cache subnet group if it is associated with any cache clusters.

Request Syntax

```
response = client.delete_cache_subnet_group(
    CacheSubnetGroupName='string'
)
```

Parameters **CacheSubnetGroupName** (*string*) – [REQUIRED]

The name of the cache subnet group to delete.

Constraints: Must contain no more than 255 alphanumeric characters or hyphens.

Returns None

delete_replication_group (***kwargs*)

The *DeleteReplicationGroup* action deletes an existing replication group. By default, this action deletes the entire replication group, including the primary cluster and all of the read replicas. You can optionally delete only the read replicas, while retaining the primary cluster.

When you receive a successful response from this action, Amazon ElastiCache immediately begins deleting the selected resources; you cannot cancel or revert this action.

Request Syntax

```
response = client.delete_replication_group(
    ReplicationGroupId='string',
    RetainPrimaryCluster=True|False,
    FinalSnapshotIdentifier='string'
)
```

Parameters

- **ReplicationGroupId** (*string*) – [REQUIRED]

The identifier for the cluster to be deleted. This parameter is not case sensitive.

- **RetainPrimaryCluster** (*boolean*) – If set to *true*, all of the read replicas will be deleted, but the primary node will be retained.

- **FinalSnapshotIdentifier** (*string*) – The name of a final node group snapshot. ElastiCache creates the snapshot from the primary node in the cluster, rather than one of the replicas; this is to ensure that it captures the freshest data. After the final snapshot is taken, the cluster is immediately deleted.

Return type dict

Returns

Response Syntax

```
{
    'ReplicationGroup': {
        'ReplicationGroupId': 'string',
        'Description': 'string',
        'Status': 'string',
        'PendingModifiedValues': {
            'PrimaryClusterId': 'string',
            'AutomaticFailoverStatus': 'enabled'|'disabled'
        },
        'MemberClusters': [
            'string',
        ],
        'NodeGroups': [
            {
                'NodeGroupId': 'string',
                'Status': 'string',
            }
        ]
    }
}
```

```
    'PrimaryEndpoint': {
        'Address': 'string',
        'Port': 123
    },
    'NodeGroupMembers': [
        {
            'CacheClusterId': 'string',
            'CacheNodeId': 'string',
            'ReadEndpoint': {
                'Address': 'string',
                'Port': 123
            },
            'PreferredAvailabilityZone': 'string',
            'CurrentRole': 'string'
        },
        ...
    ],
    ...
],
'SnapshottingClusterId': 'string',
'AutomaticFailover': 'enabled'|'disabled'|'enabling'|'disabling'
}
```

Response Structure

- (*dict*) –

Contains all of the attributes of a specific replication group.

- **ReplicationGroup** (*dict*) –

Contains all of the attributes of a specific replication group.

- * **ReplicationGroupId** (*string*) –

The identifier for the replication group.

- * **Description** (*string*) –

The description of the replication group.

- * **Status** (*string*) –

The current state of this replication group - *creating* , *available* , etc.

- * **PendingModifiedValues** (*dict*) –

A group of settings to be applied to the replication group, either immediately or during the next maintenance window.

- **PrimaryClusterId** (*string*) –

The primary cluster ID which will be applied immediately (if `--apply-immediately` was specified), or during the next maintenance window.

- **AutomaticFailoverStatus** (*string*) –

Indicates the status of Multi-AZ for this replication group.

Note: ElastiCache Multi-AZ replication groups are not supported on:

- Redis versions earlier than 2.8.6.
 - T1 and T2 cache node types.
-

- * **MemberClusters** (*list*) –

The names of all the cache clusters that are part of this replication group.

- (*string*) –
- * **NodeGroups** (*list*) –

A single element list with information about the nodes in the replication group.

 - (*dict*) –

Represents a collection of cache nodes in a replication group.
 - **NodeGroupId** (*string*) –

The identifier for the node group. A replication group contains only one node group; therefore, the node group ID is 0001.
 - **Status** (*string*) –

The current state of this replication group - *creating* , *available* , etc.
 - **PrimaryEndpoint** (*dict*) –

Represents the information required for client programs to connect to a cache node.
 - **Address** (*string*) –

The DNS hostname of the cache node.
 - **Port** (*integer*) –

The port number that the cache engine is listening on.
 - **NodeGroupMembers** (*list*) –

A list containing information about individual nodes within the node group.
 - (*dict*) –

Represents a single node within a node group.
 - **CacheClusterId** (*string*) –

The ID of the cache cluster to which the node belongs.
 - **CacheNodeId** (*string*) –

The ID of the node within its cache cluster. A node ID is a numeric identifier (0001, 0002, etc.).
 - **ReadEndpoint** (*dict*) –

Represents the information required for client programs to connect to a cache node.
 - **Address** (*string*) –

The DNS hostname of the cache node.
 - **Port** (*integer*) –

The port number that the cache engine is listening on.
 - **PreferredAvailabilityZone** (*string*) –

The name of the Availability Zone in which the node is located.
 - **CurrentRole** (*string*) –

The role that is currently assigned to the node - *primary* or *replica* .
- * **SnapshottingClusterId** (*string*) –

The cache cluster ID that is used as the daily snapshot source for the replication group.

* **AutomaticFailover** (*string*) –

Indicates the status of Multi-AZ for this replication group.

Note: ElastiCache Multi-AZ replication groups are not supported on:

- Redis versions earlier than 2.8.6.
 - T1 and T2 cache node types.
-

delete_snapshot (**kwargs)

The *DeleteSnapshot* action deletes an existing snapshot. When you receive a successful response from this action, ElastiCache immediately begins deleting the snapshot; you cannot cancel or revert this action.

Request Syntax

```
response = client.delete_snapshot(  
    SnapshotName='string'  
)
```

Parameters **SnapshotName** (*string*) – [REQUIRED]

The name of the snapshot to be deleted.

Return type dict

Returns

Response Syntax

```
{  
    'Snapshot': {  
        'SnapshotName': 'string',  
        'CacheClusterId': 'string',  
        'SnapshotStatus': 'string',  
        'SnapshotSource': 'string',  
        'CacheNodeType': 'string',  
        'Engine': 'string',  
        'EngineVersion': 'string',  
        'NumCacheNodes': 123,  
        'PreferredAvailabilityZone': 'string',  
        'CacheClusterCreateTime': datetime(2015, 1, 1),  
        'PreferredMaintenanceWindow': 'string',  
        'TopicArn': 'string',  
        'Port': 123,  
        'CacheParameterGroupName': 'string',  
        'CacheSubnetGroupName': 'string',  
        'VpcId': 'string',  
        'AutoMinorVersionUpgrade': True|False,  
        'SnapshotRetentionLimit': 123,  
        'SnapshotWindow': 'string',  
        'NodeSnapshots': [  
            {  
                'CacheNodeId': 'string',  
                'CacheSize': 'string',  
                'CacheNodeCreateTime': datetime(2015, 1, 1),  
                'SnapshotCreateTime': datetime(2015, 1, 1)  
            },  
            ...  
        ]  
    }  
}
```



```
}
}
```

Response Structure

- (*dict*) –

Represents a copy of an entire cache cluster as of the time when the snapshot was taken.

- **Snapshot** (*dict*) –

Represents a copy of an entire cache cluster as of the time when the snapshot was taken.

- * **SnapshotName** (*string*) –

The name of a snapshot. For an automatic snapshot, the name is system-generated; for a manual snapshot, this is the user-provided name.

- * **CacheClusterId** (*string*) –

The user-supplied identifier of the source cache cluster.

- * **SnapshotStatus** (*string*) –

The status of the snapshot. Valid values: `creating` | `available` | `restoring` | `copying` | `deleting`.

- * **SnapshotSource** (*string*) –

Indicates whether the snapshot is from an automatic backup (`automated`) or was created manually (`manual`).

- * **CacheNodeType** (*string*) –

The name of the compute and memory capacity node type for the source cache cluster.

Valid node types are as follows:

- General purpose:
 - * Current generation: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
 - Previous generation: `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
- Compute optimized: `cache.c1.xlarge`
- Memory optimized
 - * Current generation: `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`, `cache.r3.4xlarge`, `cache.r3.8xlarge`
 - Previous generation: `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#).

* **Engine** (*string*) –

The name of the cache engine (*memcached* or *redis*) used by the source cache cluster.

* **EngineVersion** (*string*) –

The version of the cache engine version that is used by the source cache cluster.

* **NumCacheNodes** (*integer*) –

The number of cache nodes in the source cache cluster.

For clusters running Redis, this value must be 1. For clusters running Memcached, this value must be between 1 and 20.

* **PreferredAvailabilityZone** (*string*) –

The name of the Availability Zone in which the source cache cluster is located.

* **CacheClusterCreateTime** (*datetime*) –

The date and time when the source cache cluster was created.

* **PreferredMaintenanceWindow** (*string*) –

Specifies the weekly time range during which maintenance on the cache cluster is performed. It is specified as a range in the format ddd:hh24:mi-ddd:hh24:mi (24H Clock UTC). The minimum maintenance window is a 60 minute period. Valid values for ddd are:

- sun
- mon
- tue
- wed
- thu
- fri
- sat

Example: sun:05:00-sun:09:00

* **TopicArn** (*string*) –

The Amazon Resource Name (ARN) for the topic used by the source cache cluster for publishing notifications.

* **Port** (*integer*) –

The port number used by each cache nodes in the source cache cluster.

* **CacheParameterGroupName** (*string*) –

The cache parameter group that is associated with the source cache cluster.

* **CacheSubnetGroupName** (*string*) –

The name of the cache subnet group associated with the source cache cluster.

* **VpcId** (*string*) –

The Amazon Virtual Private Cloud identifier (VPC ID) of the cache subnet group for the source cache cluster.

* **AutoMinorVersionUpgrade** (*boolean*) –

This parameter is currently disabled.

* **SnapshotRetentionLimit** (*integer*) –

For an automatic snapshot, the number of days for which ElastiCache will retain the snapshot before deleting it.

For manual snapshots, this field reflects the *SnapshotRetentionLimit* for the source cache cluster when the snapshot was created. This field is otherwise ignored: Manual snapshots do not expire, and can only be deleted using the *DeleteSnapshot* action.

Important If the value of *SnapshotRetentionLimit* is set to zero (0), backups are turned off.

* **SnapshotWindow** (*string*) –

The daily time range during which ElastiCache takes daily snapshots of the source cache cluster.

* **NodeSnapshots** (*list*) –

A list of the cache nodes in the source cache cluster.

· (*dict*) –

Represents an individual cache node in a snapshot of a cache cluster.

· **CacheNodeId** (*string*) –

The cache node identifier for the node in the source cache cluster.

· **CacheSize** (*string*) –

The size of the cache on the source cache node.

· **CacheNodeCreateTime** (*datetime*) –

The date and time when the cache node was created in the source cache cluster.

· **SnapshotCreateTime** (*datetime*) –

The date and time when the source node's metadata and cache data set was obtained for the snapshot.

describe_cache_clusters (***kwargs*)

The *DescribeCacheClusters* action returns information about all provisioned cache clusters if no cache cluster identifier is specified, or about a specific cache cluster if a cache cluster identifier is supplied.

By default, abbreviated information about the cache clusters(s) will be returned. You can use the optional *ShowDetails* flag to retrieve detailed information about the cache nodes associated with the cache clusters. These details include the DNS address and port for the cache node endpoint.

If the cluster is in the *CREATING* state, only cluster level information will be displayed until all of the nodes are successfully provisioned.

If the cluster is in the *DELETING* state, only cluster level information will be displayed.

If cache nodes are currently being added to the cache cluster, node endpoint information and creation time for the additional nodes will not be displayed until they are completely provisioned. When the cache cluster state is *available*, the cluster is ready for use.

If cache nodes are currently being removed from the cache cluster, no endpoint information for the removed nodes is displayed.

Request Syntax

```
response = client.describe_cache_clusters(
    CacheClusterId='string',
    MaxRecords=123,
```

```
Marker='string',
ShowCacheNodeInfo=True|False
)
```

Parameters

- **CacheClusterId** (*string*) – The user-supplied cluster identifier. If this parameter is specified, only information about that specific cache cluster is returned. This parameter isn't case sensitive.
- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

- **Marker** (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this action. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.
- **ShowCacheNodeInfo** (*boolean*) – An optional flag that can be included in the `DescribeCacheCluster` request to retrieve information about the individual cache nodes.

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'CacheClusters': [
    {
      'CacheClusterId': 'string',
      'ConfigurationEndpoint': {
        'Address': 'string',
        'Port': 123
      },
      'ClientDownloadLandingPage': 'string',
      'CacheNodeType': 'string',
      'Engine': 'string',
      'EngineVersion': 'string',
      'CacheClusterStatus': 'string',
      'NumCacheNodes': 123,
      'PreferredAvailabilityZone': 'string',
      'CacheClusterCreateTime': datetime(2015, 1, 1),
      'PreferredMaintenanceWindow': 'string',
      'PendingModifiedValues': {
        'NumCacheNodes': 123,
        'CacheNodeIdsToRemove': [
          'string',
        ],
        'EngineVersion': 'string'
      },
      'NotificationConfiguration': {
        'TopicArn': 'string',
        'TopicStatus': 'string'
      },
      'CacheSecurityGroups': [
        {
```

```

        'CacheSecurityGroupName': 'string',
        'Status': 'string'
    },
],
'CacheParameterGroup': {
    'CacheParameterGroupName': 'string',
    'ParameterApplyStatus': 'string',
    'CacheNodeIdsToReboot': [
        'string',
    ]
},
'CacheSubnetGroupName': 'string',
'CacheNodes': [
    {
        'CacheNodeId': 'string',
        'CacheNodeStatus': 'string',
        'CacheNodeCreateTime': datetime(2015, 1, 1),
        'Endpoint': {
            'Address': 'string',
            'Port': 123
        },
        'ParameterGroupStatus': 'string',
        'SourceCacheNodeId': 'string',
        'CustomerAvailabilityZone': 'string'
    },
],
'AutoMinorVersionUpgrade': True|False,
'SecurityGroups': [
    {
        'SecurityGroupId': 'string',
        'Status': 'string'
    },
],
'ReplicationGroupId': 'string',
'SnapshotRetentionLimit': 123,
'SnapshotWindow': 'string'
},
]
}

```

Response Structure

- (*dict*) –

Represents the output of a *DescribeCacheClusters* action.

- **Marker** (*string*) –

Provides an identifier to allow retrieval of paginated results.

- **CacheClusters** (*list*) –

A list of cache clusters. Each item in the list contains detailed information about one cache cluster.

- * (*dict*) –

Contains all of the attributes of a specific cache cluster.

- **CacheClusterId** (*string*) –

The user-supplied identifier of the cache cluster. This identifier is a unique key that identifies a cache cluster.

- **ConfigurationEndpoint** (*dict*) –

Represents the information required for client programs to connect to a cache node.

- **Address** (*string*) –

The DNS hostname of the cache node.

- **Port** (*integer*) –

The port number that the cache engine is listening on.

- **ClientDownloadLandingPage** (*string*) –

The URL of the web page where you can download the latest ElastiCache client library.

- **CacheNodeType** (*string*) –

The name of the compute and memory capacity node type for the cache cluster.

Valid node types are as follows:

- **General purpose:** * **Current generation:** `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
- **Previous generation:** `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
- **Compute optimized:** `cache.c1.xlarge`
- **Memory optimized** * **Current generation:** `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`, `cache.r3.4xlarge`, `cache.r3.8xlarge`
- **Previous generation:** `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#).

- **Engine** (*string*) –

The name of the cache engine (*memcached* or *redis*) to be used for this cache cluster.

- **EngineVersion** (*string*) –

The version of the cache engine version that is used in this cache cluster.

- **CacheClusterStatus** (*string*) –

The current state of this cache cluster, one of the following values: *available*, *creating*, *deleted*, *deleting*, *incompatible-network*, *modifying*, *rebooting cache cluster nodes*, *restore-failed*, or *snapshotting*.

- **NumCacheNodes** (*integer*) –

The number of cache nodes in the cache cluster.

For clusters running Redis, this value must be 1. For clusters running Memcached, this value must be between 1 and 20.

- **PreferredAvailabilityZone** (*string*) –

The name of the Availability Zone in which the cache cluster is located or “Multiple” if the cache nodes are located in different Availability Zones.

- **CacheClusterCreateTime** (*datetime*) –

The date and time when the cache cluster was created.

- **PreferredMaintenanceWindow** (*string*) –

Specifies the weekly time range during which maintenance on the cache cluster is performed. It is specified as a range in the format ddd:hh24:mi-ddd:hh24:mi (24H Clock UTC). The minimum maintenance window is a 60 minute period. Valid values for ddd are:

- sun
- mon
- tue
- wed
- thu
- fri
- sat

Example: sun:05:00-sun:09:00

- **PendingModifiedValues** (*dict*) –

A group of settings that will be applied to the cache cluster in the future, or that are currently being applied.

- **NumCacheNodes** (*integer*) –

The new number of cache nodes for the cache cluster.

For clusters running Redis, this value must be 1. For clusters running Memcached, this value must be between 1 and 20.

- **CacheNodeIdsToRemove** (*list*) –

A list of cache node IDs that are being removed (or will be removed) from the cache cluster. A node ID is a numeric identifier (0001, 0002, etc.).

- (*string*) –

- **EngineVersion** (*string*) –

The new cache engine version that the cache cluster will run.

- **NotificationConfiguration** (*dict*) –

Describes a notification topic and its status. Notification topics are used for publishing ElastiCache events to subscribers using Amazon Simple Notification Service (SNS).

- **TopicArn** (*string*) –

The Amazon Resource Name (ARN) that identifies the topic.

- **TopicStatus** (*string*) –

The current state of the topic.

- **CacheSecurityGroups** (*list*) –

A list of cache security group elements, composed of name and status sub-elements.

- *(dict)* –

Represents a cache cluster's status within a particular cache security group.

- **CacheSecurityGroupName** (*string*) –

The name of the cache security group.

- **Status** (*string*) –

The membership status in the cache security group. The status changes when a cache security group is modified, or when the cache security groups assigned to a cache cluster are modified.

- **CacheParameterGroup** (*dict*) –

The status of the cache parameter group.

- **CacheParameterGroupName** (*string*) –

The name of the cache parameter group.

- **ParameterApplyStatus** (*string*) –

The status of parameter updates.

- **CacheNodeIdsToReboot** (*list*) –

A list of the cache node IDs which need to be rebooted for parameter changes to be applied. A node ID is a numeric identifier (0001, 0002, etc.).

- (*string*) –

- **CacheSubnetGroupName** (*string*) –

The name of the cache subnet group associated with the cache cluster.

- **CacheNodes** (*list*) –

A list of cache nodes that are members of the cache cluster.

- (*dict*) –

Represents an individual cache node within a cache cluster. Each cache node runs its own instance of the cluster's protocol-compliant caching software - either Memcached or Redis.

Valid node types are as follows:

- General purpose: * Current generation: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
- Previous generation: `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
- Compute optimized: `cache.c1.xlarge`
- Memory optimized * Current generation: `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`, `cache.r3.4xlarge`, `cache.r3.8xlarge`
- Previous generation: `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#).

- **CacheNodeId** (*string*) –

The cache node identifier. A node ID is a numeric identifier (0001, 0002, etc.). The combination of cluster ID and node ID uniquely identifies every cache node used in a customer's AWS account.

- **CacheNodeStatus** (*string*) –

The current state of this cache node.

- **CacheNodeCreateTime** (*datetime*) –

The date and time when the cache node was created.

- **Endpoint** (*dict*) –

The hostname for connecting to this cache node.

- **Address** (*string*) –

The DNS hostname of the cache node.

- **Port** (*integer*) –

The port number that the cache engine is listening on.

- **ParameterGroupStatus** (*string*) –

The status of the parameter group applied to this cache node.

- **SourceCacheNodeId** (*string*) –

The ID of the primary node to which this read replica node is synchronized. If this field is empty, then this node is not associated with a primary cache cluster.

- **CustomerAvailabilityZone** (*string*) –

The Availability Zone where this node was created and now resides.

- **AutoMinorVersionUpgrade** (*boolean*) –

This parameter is currently disabled.

- **SecurityGroups** (*list*) –

A list of VPC Security Groups associated with the cache cluster.

- (*dict*) –

Represents a single cache security group and its status.

- **SecurityGroupId** (*string*) –

The identifier of the cache security group.

- **Status** (*string*) –

The status of the cache security group membership. The status changes whenever a cache security group is modified, or

when the cache security groups assigned to a cache cluster are modified.

- **ReplicationGroupId** (*string*) –

The replication group to which this cache cluster belongs. If this field is empty, the cache cluster is not associated with any replication group.

- **SnapshotRetentionLimit** (*integer*) –

The number of days for which ElastiCache will retain automatic cache cluster snapshots before deleting them. For example, if you set *SnapshotRetentionLimit* to 5, then a snapshot that was taken today will be retained for 5 days before being deleted.

Important If the value of *SnapshotRetentionLimit* is set to zero (0), backups are turned off.

- **SnapshotWindow** (*string*) –

The daily time range (in UTC) during which ElastiCache will begin taking a daily snapshot of your cache cluster.

Example: 05:00–09:00

describe_cache_engine_versions (***kwargs*)

The *DescribeCacheEngineVersions* action returns a list of the available cache engines and their versions.

Request Syntax

```
response = client.describe_cache_engine_versions(
    Engine='string',
    EngineVersion='string',
    CacheParameterGroupFamily='string',
    MaxRecords=123,
    Marker='string',
    DefaultOnly=True|False
)
```

Parameters

- **Engine** (*string*) – The cache engine to return. Valid values: memcached | redis
- **EngineVersion** (*string*) – The cache engine version to return.

Example: 1.4.14

- **CacheParameterGroupFamily** (*string*) – The name of a specific cache parameter group family to return details for.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

- **Marker** (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this action. If this parameter is specified,

the response includes only records beyond the marker, up to the value specified by *MaxRecords*.

- **DefaultOnly** (*boolean*) – If *true*, specifies that only the default version of the specified engine or engine and major version combination is to be returned.

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'CacheEngineVersions': [
    {
      'Engine': 'string',
      'EngineVersion': 'string',
      'CacheParameterGroupFamily': 'string',
      'CacheEngineDescription': 'string',
      'CacheEngineVersionDescription': 'string'
    },
  ]
}
```

Response Structure

- (*dict*) –

Represents the output of a DescribeCacheEngineVersions action.

- **Marker** (*string*) –

Provides an identifier to allow retrieval of paginated results.

- **CacheEngineVersions** (*list*) –

A list of cache engine version details. Each element in the list contains detailed information about one cache engine version.

- * (*dict*) –

Provides all of the details about a particular cache engine version.

- **Engine** (*string*) –

The name of the cache engine.

- **EngineVersion** (*string*) –

The version number of the cache engine.

- **CacheParameterGroupFamily** (*string*) –

The name of the cache parameter group family associated with this cache engine.

- **CacheEngineDescription** (*string*) –

The description of the cache engine.

- **CacheEngineVersionDescription** (*string*) –

The description of the cache engine version.

describe_cache_parameter_groups (***kwargs*)

The *DescribeCacheParameterGroups* action returns a list of cache parameter group descriptions. If a cache parameter group name is specified, the list will contain only the descriptions for that group.

Request Syntax

```
response = client.describe_cache_parameter_groups(  
    CacheParameterGroupName='string',  
    MaxRecords=123,  
    Marker='string'  
)
```

Parameters

- **CacheParameterGroupName** (*string*) – The name of a specific cache parameter group to return details for.
- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

- **Marker** (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this action. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.

Return type dict

Returns

Response Syntax

```
{  
    'Marker': 'string',  
    'CacheParameterGroups': [  
        {  
            'CacheParameterGroupName': 'string',  
            'CacheParameterGroupFamily': 'string',  
            'Description': 'string'  
        },  
    ]  
}
```

Response Structure

- (*dict*) –

Represents the output of a *DescribeCacheParameterGroups* action.

- **Marker** (*string*) –

Provides an identifier to allow retrieval of paginated results.

- **CacheParameterGroups** (*list*) –

A list of cache parameter groups. Each element in the list contains detailed information about one cache parameter group.

- * (*dict*) –

Represents the output of a *CreateCacheParameterGroup* action.

- **CacheParameterGroupName** (*string*) –

The name of the cache parameter group.

- **CacheParameterGroupFamily** (*string*) –

The name of the cache parameter group family that this cache parameter group is compatible with.

- **Description** (*string*) –

The description for this cache parameter group.

describe_cache_parameters (**kwargs)

The *DescribeCacheParameters* action returns the detailed parameter list for a particular cache parameter group.

Request Syntax

```
response = client.describe_cache_parameters(
    CacheParameterGroupName='string',
    Source='string',
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **CacheParameterGroupName** (*string*) – [REQUIRED]
The name of a specific cache parameter group to return details for.
- **Source** (*string*) – The parameter types to return.
Valid values: `user` | `system` | `engine-default`
- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a marker is included in the response so that the remaining results can be retrieved.
Default: 100
Constraints: minimum 20; maximum 100.
- **Marker** (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this action. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.

Return type dict

Returns**Response Syntax**

```
{
  'Marker': 'string',
  'Parameters': [
    {
      'ParameterName': 'string',
      'ParameterValue': 'string',
      'Description': 'string',
      'Source': 'string',
      'DataType': 'string',
      'AllowedValues': 'string',
      'IsModifiable': True|False,
      'MinimumEngineVersion': 'string'
    },
  ],
  'CacheNodeTypeSpecificParameters': [
    {
      'ParameterName': 'string',
      'Description': 'string',
      'Source': 'string',
      'DataType': 'string',
      'AllowedValues': 'string',
      'IsModifiable': True|False,
      'MinimumEngineVersion': 'string',
```

```
        'CacheNodeTypeSpecificValues': [
            {
                'CacheNodeType': 'string',
                'Value': 'string'
            },
        ],
    ],
}
```

Response Structure

- (*dict*) –

Represents the output of a *DescribeCacheParameters* action.

- **Marker** (*string*) –

Provides an identifier to allow retrieval of paginated results.

- **Parameters** (*list*) –

A list of Parameter instances.

- * (*dict*) –

Describes an individual setting that controls some aspect of ElastiCache behavior.

- **ParameterName** (*string*) –

The name of the parameter.

- **ParameterValue** (*string*) –

The value of the parameter.

- **Description** (*string*) –

A description of the parameter.

- **Source** (*string*) –

The source of the parameter.

- **DataType** (*string*) –

The valid data type for the parameter.

- **AllowedValues** (*string*) –

The valid range of values for the parameter.

- **IsModifiable** (*boolean*) –

Indicates whether (*true*) or not (*false*) the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.

- **MinimumEngineVersion** (*string*) –

The earliest cache engine version to which the parameter can apply.

- **CacheNodeTypeSpecificParameters** (*list*) –

A list of parameters specific to a particular cache node type. Each element in the list contains detailed information about one parameter.

- * (*dict*) –

A parameter that has a different value for each cache node type it is applied to. For example, in a Redis cache cluster, a *cache.m1.large* cache node type would have a larger *maxmemory* value than a *cache.m1.small* type.

- **ParameterName** (*string*) –
The name of the parameter.
- **Description** (*string*) –
A description of the parameter.
- **Source** (*string*) –
The source of the parameter value.
- **DataType** (*string*) –
The valid data type for the parameter.
- **AllowedValues** (*string*) –
The valid range of values for the parameter.
- **IsModifiable** (*boolean*) –
Indicates whether (`true`) or not (`false`) the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.
- **MinimumEngineVersion** (*string*) –
The earliest cache engine version to which the parameter can apply.
- **CacheNodeTypeSpecificValues** (*list*) –
A list of cache node types and their corresponding values for this parameter.
- (*dict*) –
A value that applies only to a certain cache node type.
- **CacheNodeType** (*string*) –
The cache node type for which this value applies.
- **Value** (*string*) –
The value for the cache node type.

describe_cache_security_groups (***kwargs*)

The *DescribeCacheSecurityGroups* action returns a list of cache security group descriptions. If a cache security group name is specified, the list will contain only the description of that group.

Request Syntax

```
response = client.describe_cache_security_groups(
    CacheSecurityGroupName='string',
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **CacheSecurityGroupName** (*string*) – The name of the cache security group to return details for.
- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

- **Marker** (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this action. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'CacheSecurityGroups': [
    {
      'OwnerId': 'string',
      'CacheSecurityGroupName': 'string',
      'Description': 'string',
      'EC2SecurityGroups': [
        {
          'Status': 'string',
          'EC2SecurityGroupName': 'string',
          'EC2SecurityGroupOwnerId': 'string'
        },
        ...
      ]
    },
    ...
  ]
}
```

Response Structure

- (*dict*) –

Represents the output of a *DescribeCacheSecurityGroups* action.

- **Marker** (*string*) –

Provides an identifier to allow retrieval of paginated results.

- **CacheSecurityGroups** (*list*) –

A list of cache security groups. Each element in the list contains detailed information about one group.

- * (*dict*) –

Represents the output of one of the following actions:

- *AuthorizeCacheSecurityGroupIngress*
- *CreateCacheSecurityGroup*
- *RevokeCacheSecurityGroupIngress*

- **OwnerId** (*string*) –

The AWS account ID of the cache security group owner.

- **CacheSecurityGroupName** (*string*) –

The name of the cache security group.

- **Description** (*string*) –

The description of the cache security group.

- **EC2SecurityGroups** (*list*) –

A list of Amazon EC2 security groups that are associated with this cache security group.

- (*dict*) –

Provides ownership and status information for an Amazon EC2 security group.

- **Status** (*string*) –

The status of the Amazon EC2 security group.

- **EC2SecurityGroupName** (*string*) –

The name of the Amazon EC2 security group.

- **EC2SecurityGroupOwnerId** (*string*) –

The AWS account ID of the Amazon EC2 security group owner.

describe_cache_subnet_groups (***kwargs*)

The *DescribeCacheSubnetGroups* action returns a list of cache subnet group descriptions. If a subnet group name is specified, the list will contain only the description of that group.

Request Syntax

```
response = client.describe_cache_subnet_groups(
    CacheSubnetGroupName='string',
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **CacheSubnetGroupName** (*string*) – The name of the cache subnet group to return details for.
- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

- **Marker** (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this action. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'CacheSubnetGroups': [
    {
      'CacheSubnetGroupName': 'string',
      'CacheSubnetGroupDescription': 'string',
      'VpcId': 'string',
      'Subnets': [
        {
          'SubnetIdentifier': 'string',
          'SubnetAvailabilityZone': {
            'Name': 'string'
          }
        }
      ],
    },
  ]
}
```

```
    },  
  ]  
}
```

Response Structure

- (*dict*) –

Represents the output of a *DescribeCacheSubnetGroups* action.

- **Marker** (*string*) –

Provides an identifier to allow retrieval of paginated results.

- **CacheSubnetGroups** (*list*) –

A list of cache subnet groups. Each element in the list contains detailed information about one group.

- * (*dict*) –

Represents the output of one of the following actions:

- *CreateCacheSubnetGroup*
- *ModifyCacheSubnetGroup*

- **CacheSubnetGroupName** (*string*) –

The name of the cache subnet group.

- **CacheSubnetGroupDescription** (*string*) –

The description of the cache subnet group.

- **VpcId** (*string*) –

The Amazon Virtual Private Cloud identifier (VPC ID) of the cache subnet group.

- **Subnets** (*list*) –

A list of subnets associated with the cache subnet group.

- (*dict*) –

Represents the subnet associated with a cache cluster. This parameter refers to subnets defined in Amazon Virtual Private Cloud (Amazon VPC) and used with ElastiCache.

- **SubnetIdentifier** (*string*) –

The unique identifier for the subnet.

- **SubnetAvailabilityZone** (*dict*) –

The Availability Zone associated with the subnet.

- **Name** (*string*) –

The name of the Availability Zone.

describe_engine_default_parameters (**kwargs)

The *DescribeEngineDefaultParameters* action returns the default engine and system parameter information for the specified cache engine.

Request Syntax

```
response = client.describe_engine_default_parameters(  
    CacheParameterGroupFamily='string',  
    MaxRecords=123,  
    Marker='string'  
)
```

Parameters

- **CacheParameterGroupFamily** (*string*) – [REQUIRED]

The name of the cache parameter group family. Valid values are: memcached1.4|redis2.6|redis2.8

- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

- **Marker** (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this action. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.

Return type dict

Returns

Response Syntax

```
{
  'EngineDefaults': {
    'CacheParameterGroupFamily': 'string',
    'Marker': 'string',
    'Parameters': [
      {
        'ParameterName': 'string',
        'ParameterValue': 'string',
        'Description': 'string',
        'Source': 'string',
        'DataType': 'string',
        'AllowedValues': 'string',
        'IsModifiable': True|False,
        'MinimumEngineVersion': 'string'
      },
    ],
    'CacheNodeTypeSpecificParameters': [
      {
        'ParameterName': 'string',
        'Description': 'string',
        'Source': 'string',
        'DataType': 'string',
        'AllowedValues': 'string',
        'IsModifiable': True|False,
        'MinimumEngineVersion': 'string',
        'CacheNodeTypeSpecificValues': [
          {
            'CacheNodeType': 'string',
            'Value': 'string'
          },
        ],
      },
    ],
  },
}
```

Response Structure

- (*dict*) –

Represents the output of a *DescribeEngineDefaultParameters* action.

– **EngineDefaults** (*dict*) –

Represents the output of a *DescribeEngineDefaultParameters* action.

* **CacheParameterGroupFamily** (*string*) –

Specifies the name of the cache parameter group family to which the engine default parameters apply.

* **Marker** (*string*) –

Provides an identifier to allow retrieval of paginated results.

* **Parameters** (*list*) –

Contains a list of engine default parameters.

· (*dict*) –

Describes an individual setting that controls some aspect of ElastiCache behavior.

· **ParameterName** (*string*) –

The name of the parameter.

· **ParameterValue** (*string*) –

The value of the parameter.

· **Description** (*string*) –

A description of the parameter.

· **Source** (*string*) –

The source of the parameter.

· **DataType** (*string*) –

The valid data type for the parameter.

· **AllowedValues** (*string*) –

The valid range of values for the parameter.

· **IsModifiable** (*boolean*) –

Indicates whether (*true*) or not (*false*) the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.

· **MinimumEngineVersion** (*string*) –

The earliest cache engine version to which the parameter can apply.

* **CacheNodeTypeSpecificParameters** (*list*) –

A list of parameters specific to a particular cache node type. Each element in the list contains detailed information about one parameter.

· (*dict*) –

A parameter that has a different value for each cache node type it is applied to. For example, in a Redis cache cluster, a *cache.m1.large* cache node type would have a larger *maxmemory* value than a *cache.m1.small* type.

· **ParameterName** (*string*) –

The name of the parameter.

· **Description** (*string*) –

A description of the parameter.

- **Source** (*string*) –
The source of the parameter value.
- **DataType** (*string*) –
The valid data type for the parameter.
- **AllowedValues** (*string*) –
The valid range of values for the parameter.
- **IsModifiable** (*boolean*) –
Indicates whether (`true`) or not (`false`) the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.
- **MinimumEngineVersion** (*string*) –
The earliest cache engine version to which the parameter can apply.
- **CacheNodeTypeSpecificValues** (*list*) –
A list of cache node types and their corresponding values for this parameter.
- (*dict*) –
A value that applies only to a certain cache node type.
- **CacheNodeType** (*string*) –
The cache node type for which this value applies.
- **Value** (*string*) –
The value for the cache node type.

describe_events (***kwargs*)

The *DescribeEvents* action returns events related to cache clusters, cache security groups, and cache parameter groups. You can obtain events specific to a particular cache cluster, cache security group, or cache parameter group by providing the name as a parameter.

By default, only the events occurring within the last hour are returned; however, you can retrieve up to 14 days' worth of events if necessary.

Request Syntax

```
response = client.describe_events(
    SourceIdentifier='string',
    SourceType='cache-cluster'|'cache-parameter-group'|'cache-security-group'|'cache-subnet',
    StartTime=datetime(2015, 1, 1),
    EndTime=datetime(2015, 1, 1),
    Duration=123,
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **SourceIdentifier** (*string*) – The identifier of the event source for which events will be returned. If not specified, then all sources are included in the response.
- **SourceType** (*string*) – The event source to retrieve events for. If no value is specified, all events are returned.

Valid values are: `cache-cluster` | `cache-parameter-group` | `cache-security-group` | `cache-subnet-group`

- **StartTime** (*datetime*) – The beginning of the time interval to retrieve events for, specified in ISO 8601 format.
- **EndTime** (*datetime*) – The end of the time interval for which to retrieve events, specified in ISO 8601 format.
- **Duration** (*integer*) – The number of minutes' worth of events to retrieve.
- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

- **Marker** (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this action. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords* .

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'Events': [
    {
      'SourceIdentifier': 'string',
      'SourceType': 'cache-cluster'|'cache-parameter-group'|'cache-security-
      'Message': 'string',
      'Date': datetime(2015, 1, 1)
    },
  ],
}
```

Response Structure

- (*dict*) –

Represents the output of a *DescribeEvents* action.

- **Marker** (*string*) –

Provides an identifier to allow retrieval of paginated results.

- **Events** (*list*) –

A list of events. Each element in the list contains detailed information about one event.

- * (*dict*) –

Represents a single occurrence of something interesting within the system. Some examples of events are creating a cache cluster, adding or removing a cache node, or rebooting a node.

- **SourceIdentifier** (*string*) –

The identifier for the source of the event. For example, if the event occurred at the cache cluster level, the identifier would be the name of the cache cluster.

- **SourceType** (*string*) –

Specifies the origin of this event - a cache cluster, a parameter group, a security group, etc.

- **Message** (*string*) –
The text of the event.
- **Date** (*datetime*) –
The date and time when the event occurred.

describe_replication_groups (**kwargs)

The *DescribeReplicationGroups* action returns information about a particular replication group. If no identifier is specified, *DescribeReplicationGroups* returns information about all replication groups.

Request Syntax

```
response = client.describe_replication_groups(
    ReplicationGroupId='string',
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **ReplicationGroupId** (*string*) – The identifier for the replication group to be described. This parameter is not case sensitive.

If you do not specify this parameter, information about all replication groups is returned.

- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

- **Marker** (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this action. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'ReplicationGroups': [
    {
      'ReplicationGroupId': 'string',
      'Description': 'string',
      'Status': 'string',
      'PendingModifiedValues': {
        'PrimaryClusterId': 'string',
        'AutomaticFailoverStatus': 'enabled'|'disabled'
      },
      'MemberClusters': [
        'string',
      ],
      'NodeGroups': [
        {
          'NodeGroupId': 'string',
          'Status': 'string',
          'PrimaryEndpoint': {
```

```
        'Address': 'string',
        'Port': 123
    },
    'NodeGroupMembers': [
        {
            'CacheClusterId': 'string',
            'CacheNodeId': 'string',
            'ReadEndpoint': {
                'Address': 'string',
                'Port': 123
            },
            'PreferredAvailabilityZone': 'string',
            'CurrentRole': 'string'
        },
    ],
    ],
    'SnapshottingClusterId': 'string',
    'AutomaticFailover': 'enabled'|'disabled'|'enabling'|'disabling'
},
]
```

Response Structure

- (*dict*) –

Represents the output of a *DescribeReplicationGroups* action.

- **Marker** (*string*) –

Provides an identifier to allow retrieval of paginated results.

- **ReplicationGroups** (*list*) –

A list of replication groups. Each item in the list contains detailed information about one replication group.

- * (*dict*) –

Contains all of the attributes of a specific replication group.

- **ReplicationGroupId** (*string*) –

The identifier for the replication group.

- **Description** (*string*) –

The description of the replication group.

- **Status** (*string*) –

The current state of this replication group - *creating* , *available* , etc.

- **PendingModifiedValues** (*dict*) –

A group of settings to be applied to the replication group, either immediately or during the next maintenance window.

- **PrimaryClusterId** (*string*) –

The primary cluster ID which will be applied immediately (if `--apply-immediately` was specified), or during the next maintenance window.

- **AutomaticFailoverStatus** (*string*) –

Indicates the status of Multi-AZ for this replication group.

Note: ElastiCache Multi-AZ replication groups are not supported on:

- Redis versions earlier than 2.8.6.
- T1 and T2 cache node types.

- **MemberClusters** (*list*) –

The names of all the cache clusters that are part of this replication group.

- (*string*) –

- **NodeGroups** (*list*) –

A single element list with information about the nodes in the replication group.

- (*dict*) –

Represents a collection of cache nodes in a replication group.

- **NodeGroupId** (*string*) –

The identifier for the node group. A replication group contains only one node group; therefore, the node group ID is 0001.

- **Status** (*string*) –

The current state of this replication group - *creating* , *available* , etc.

- **PrimaryEndpoint** (*dict*) –

Represents the information required for client programs to connect to a cache node.

- **Address** (*string*) –

The DNS hostname of the cache node.

- **Port** (*integer*) –

The port number that the cache engine is listening on.

- **NodeGroupMembers** (*list*) –

A list containing information about individual nodes within the node group.

- (*dict*) –

Represents a single node within a node group.

- **CacheClusterId** (*string*) –

The ID of the cache cluster to which the node belongs.

- **CacheNodeId** (*string*) –

The ID of the node within its cache cluster. A node ID is a numeric identifier (0001, 0002, etc.).

- **ReadEndpoint** (*dict*) –

Represents the information required for client programs to connect to a cache node.

- **Address** (*string*) –

The DNS hostname of the cache node.

- **Port** (*integer*) –

The port number that the cache engine is listening on.

- **PreferredAvailabilityZone** (*string*) –

The name of the Availability Zone in which the node is located.

- **CurrentRole** (*string*) –

The role that is currently assigned to the node - *primary* or *replica*.

- **SnapshottingClusterId** (*string*) –

The cache cluster ID that is used as the daily snapshot source for the replication group.

- **AutomaticFailover** (*string*) –

Indicates the status of Multi-AZ for this replication group.

Note: ElastiCache Multi-AZ replication groups are not supported on:

- Redis versions earlier than 2.8.6.
 - T1 and T2 cache node types.
-

describe_reserved_cache_nodes (**kwargs)

The *DescribeReservedCacheNodes* action returns information about reserved cache nodes for this account, or about a specified reserved cache node.

Request Syntax

```
response = client.describe_reserved_cache_nodes(
    ReservedCacheNodeId='string',
    ReservedCacheNodesOfferingId='string',
    CacheNodeType='string',
    Duration='string',
    ProductDescription='string',
    OfferingType='string',
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **ReservedCacheNodeId** (*string*) – The reserved cache node identifier filter value. Use this parameter to show only the reservation that matches the specified reservation ID.
- **ReservedCacheNodesOfferingId** (*string*) – The offering identifier filter value. Use this parameter to show only purchased reservations matching the specified offering identifier.
- **CacheNodeType** (*string*) – The cache node type filter value. Use this parameter to show only those reservations matching the specified cache node type.

Valid node types are as follows:

- General purpose: * Current generation: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
- Previous generation: `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
- Compute optimized: `cache.c1.xlarge`
- Memory optimized * Current generation: `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`, `cache.r3.4xlarge`, `cache.r3.8xlarge`
- Previous generation: `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#).

- **Duration** (*string*) – The duration filter value, specified in years or seconds. Use this parameter to show only reservations for this duration.

Valid Values: 1 | 3 | 31536000 | 94608000

- **ProductDescription** (*string*) – The product description filter value. Use this parameter to show only those reservations matching the specified product description.
- **OfferingType** (*string*) – The offering type filter value. Use this parameter to show only the available offerings matching the specified offering type.

Valid values: "Light Utilization"|"Medium Utilization"|"Heavy Utilization"

- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

- **Marker** (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this action. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'ReservedCacheNodes': [
    {
      'ReservedCacheNodeId': 'string',
      'ReservedCacheNodesOfferingId': 'string',
      'CacheNodeType': 'string',
      'StartTime': datetime(2015, 1, 1),
      'Duration': 123,
      'FixedPrice': 123.0,
      'UsagePrice': 123.0,
      'CacheNodeCount': 123,
      'ProductDescription': 'string',
      'OfferingType': 'string',
      'State': 'string',
      'RecurringCharges': [
        {
          'RecurringChargeAmount': 123.0,
          'RecurringChargeFrequency': 'string'
        }
      ],
    }
  ],
}
```

```
]
}
```

Response Structure

- *(dict)* –

Represents the output of a *DescribeReservedCacheNodes* action.

- **Marker** (*string*) –

Provides an identifier to allow retrieval of paginated results.

- **ReservedCacheNodes** (*list*) –

A list of reserved cache nodes. Each element in the list contains detailed information about one node.

- * *(dict)* –

Represents the output of a *PurchaseReservedCacheNodesOffering* action.

- **ReservedCacheNodeId** (*string*) –

The unique identifier for the reservation.

- **ReservedCacheNodesOfferingId** (*string*) –

The offering identifier.

- **CacheNodeType** (*string*) –

The cache node type for the reserved cache nodes.

Valid node types are as follows:

- **General purpose:** * **Current generation:** `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
- **Previous generation:** `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
- **Compute optimized:** `cache.c1.xlarge`
- **Memory optimized** * **Current generation:** `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`, `cache.r3.4xlarge`, `cache.r3.8xlarge`
- **Previous generation:** `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#).

- **StartTime** (*datetime*) –

The time the reservation started.

- **Duration** (*integer*) –

The duration of the reservation in seconds.

- **FixedPrice** (*float*) –
The fixed price charged for this reserved cache node.
- **UsagePrice** (*float*) –
The hourly price charged for this reserved cache node.
- **CacheNodeCount** (*integer*) –
The number of cache nodes that have been reserved.
- **ProductDescription** (*string*) –
The description of the reserved cache node.
- **OfferingType** (*string*) –
The offering type of this reserved cache node.
- **State** (*string*) –
The state of the reserved cache node.
- **RecurringCharges** (*list*) –
The recurring price charged to run this reserved cache node.
(*dict*) –
Contains the specific price and frequency of a recurring charges for a reserved cache node, or for a reserved cache node offering.
- **RecurringChargeAmount** (*float*) –
The monetary amount of the recurring charge.
- **RecurringChargeFrequency** (*string*) –
The frequency of the recurring charge.

describe_reserved_cache_nodes_offerings (***kwargs*)

The *DescribeReservedCacheNodesOfferings* action lists available reserved cache node offerings.

Request Syntax

```
response = client.describe_reserved_cache_nodes_offerings(
    ReservedCacheNodesOfferingId='string',
    CacheNodeType='string',
    Duration='string',
    ProductDescription='string',
    OfferingType='string',
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **ReservedCacheNodesOfferingId** (*string*) – The offering identifier filter value. Use this parameter to show only the available offering that matches the specified reservation identifier.

Example: 438012d3-4052-4cc7-b2e3-8d3372e0e706

- **CacheNodeType** (*string*) – The cache node type filter value. Use this parameter to show only the available offerings matching the specified cache node type.

Valid node types are as follows:

- General purpose: * Current generation: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`

- Previous generation: `cache.t1.micro` , `cache.m1.small` , `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
- Compute optimized: `cache.c1.xlarge`
- Memory optimized * Current generation: `cache.r3.large` , `cache.r3.xlarge` , `cache.r3.2xlarge` , `cache.r3.4xlarge`, `cache.r3.8xlarge`
- Previous generation: `cache.m2.xlarge` , `cache.m2.2xlarge` , `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#) .

- **Duration** (*string*) – Duration filter value, specified in years or seconds. Use this parameter to show only reservations for a given duration.

Valid Values: 1 | 3 | 31536000 | 94608000

- **ProductDescription** (*string*) – The product description filter value. Use this parameter to show only the available offerings matching the specified product description.
- **OfferingType** (*string*) – The offering type filter value. Use this parameter to show only the available offerings matching the specified offering type.

Valid Values: "Light Utilization"|"Medium Utilization"|"Heavy Utilization"

- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

- **Marker** (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this action. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords* .

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'ReservedCacheNodesOfferings': [
    {
      'ReservedCacheNodesOfferingId': 'string',
      'CacheNodeType': 'string',
      'Duration': 123,
      'FixedPrice': 123.0,
      'UsagePrice': 123.0,
      'ProductDescription': 'string',
      'OfferingType': 'string',
      'RecurringCharges': [
        {
```

```

        'RecurringChargeAmount': 123.0,
        'RecurringChargeFrequency': 'string'
    },
]
},
]
}

```

Response Structure

- (*dict*) –

Represents the output of a *DescribeReservedCacheNodesOfferings* action.

- **Marker** (*string*) –

Provides an identifier to allow retrieval of paginated results.

- **ReservedCacheNodesOfferings** (*list*) –

A list of reserved cache node offerings. Each element in the list contains detailed information about one offering.

- * (*dict*) –

Describes all of the attributes of a reserved cache node offering.

- **ReservedCacheNodesOfferingId** (*string*) –

A unique identifier for the reserved cache node offering.

- **CacheNodeType** (*string*) –

The cache node type for the reserved cache node.

Valid node types are as follows:

- **General purpose:** * **Current generation:** `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
- **Previous generation:** `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
- **Compute optimized:** `cache.c1.xlarge`
- **Memory optimized** * **Current generation:** `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`, `cache.r3.4xlarge`, `cache.r3.8xlarge`
- **Previous generation:** `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#).

- **Duration** (*integer*) –

The duration of the offering. in seconds.

- **FixedPrice** (*float*) –

- The fixed price charged for this offering.
- **UsagePrice** (*float*) –
The hourly price charged for this offering.
- **ProductDescription** (*string*) –
The cache engine used by the offering.
- **OfferingType** (*string*) –
The offering type.
- **RecurringCharges** (*list*) –
The recurring price charged to run this reserved cache node.
(*dict*) –
Contains the specific price and frequency of a recurring charges for a reserved cache node, or for a reserved cache node offering.
- **RecurringChargeAmount** (*float*) –
The monetary amount of the recurring charge.
- **RecurringChargeFrequency** (*string*) –
The frequency of the recurring charge.

describe_snapshots (***kwargs*)

The *DescribeSnapshots* action returns information about cache cluster snapshots. By default, *DescribeSnapshots* lists all of your snapshots; it can optionally describe a single snapshot, or just the snapshots associated with a particular cache cluster.

Request Syntax

```
response = client.describe_snapshots(
    CacheClusterId='string',
    SnapshotName='string',
    SnapshotSource='string',
    Marker='string',
    MaxRecords=123
)
```

Parameters

- **CacheClusterId** (*string*) – A user-supplied cluster identifier. If this parameter is specified, only snapshots associated with that specific cache cluster will be described.
- **SnapshotName** (*string*) – A user-supplied name of the snapshot. If this parameter is specified, only this snapshot will be described.
- **SnapshotSource** (*string*) – If set to `system`, the output shows snapshots that were automatically created by ElastiCache. If set to `user` the output shows snapshots that were manually created. If omitted, the output shows both automatically and manually created snapshots.
- **Marker** (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this action. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.
- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a marker is included in the response so that the remaining results can be retrieved.

Default: 50

Constraints: minimum 20; maximum 50.

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'Snapshots': [
    {
      'SnapshotName': 'string',
      'CacheClusterId': 'string',
      'SnapshotStatus': 'string',
      'SnapshotSource': 'string',
      'CacheNodeType': 'string',
      'Engine': 'string',
      'EngineVersion': 'string',
      'NumCacheNodes': 123,
      'PreferredAvailabilityZone': 'string',
      'CacheClusterCreateTime': datetime(2015, 1, 1),
      'PreferredMaintenanceWindow': 'string',
      'TopicArn': 'string',
      'Port': 123,
      'CacheParameterGroupName': 'string',
      'CacheSubnetGroupName': 'string',
      'VpcId': 'string',
      'AutoMinorVersionUpgrade': True|False,
      'SnapshotRetentionLimit': 123,
      'SnapshotWindow': 'string',
      'NodeSnapshots': [
        {
          'CacheNodeId': 'string',
          'CacheSize': 'string',
          'CacheNodeCreateTime': datetime(2015, 1, 1),
          'SnapshotCreateTime': datetime(2015, 1, 1)
        },
        ...
      ],
    },
    ...
  ],
}
```

Response Structure

- (*dict*) –

Represents the output of a *DescribeSnapshots* action.

- **Marker** (*string*) –

An optional marker returned from a prior request. Use this marker for pagination of results from this action. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.

- **Snapshots** (*list*) –

A list of snapshots. Each item in the list contains detailed information about one snapshot.

- * (*dict*) –

Represents a copy of an entire cache cluster as of the time when the snapshot was taken.

- **SnapshotName** (*string*) –
The name of a snapshot. For an automatic snapshot, the name is system-generated; for a manual snapshot, this is the user-provided name.
- **CacheClusterId** (*string*) –
The user-supplied identifier of the source cache cluster.
- **SnapshotStatus** (*string*) –
The status of the snapshot. Valid values: `creating` | `available` | `restoring` | `copying` | `deleting`.
- **SnapshotSource** (*string*) –
Indicates whether the snapshot is from an automatic backup (`automated`) or was created manually (`manual`).
- **CacheNodeType** (*string*) –
The name of the compute and memory capacity node type for the source cache cluster.

Valid node types are as follows:
 - General purpose: * Current generation: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
 - Previous generation: `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
 - Compute optimized: `cache.c1.xlarge`
 - Memory optimized * Current generation: `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`, `cache.r3.4xlarge`, `cache.r3.8xlarge`
 - Previous generation: `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`
- **Notes:**
 - All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
 - Redis backup/restore is not supported for t2 instances.
 - Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#).
- **Engine** (*string*) –
The name of the cache engine (`memcached` or `redis`) used by the source cache cluster.
- **EngineVersion** (*string*) –
The version of the cache engine version that is used by the source cache cluster.
- **NumCacheNodes** (*integer*) –
The number of cache nodes in the source cache cluster.

For clusters running Redis, this value must be 1. For clusters running Memcached, this value must be between 1 and 20.

- **PreferredAvailabilityZone** (*string*) –

The name of the Availability Zone in which the source cache cluster is located.

- **CacheClusterCreateTime** (*datetime*) –

The date and time when the source cache cluster was created.

- **PreferredMaintenanceWindow** (*string*) –

Specifies the weekly time range during which maintenance on the cache cluster is performed. It is specified as a range in the format ddd:hh24:mi-ddd:hh24:mi (24H Clock UTC). The minimum maintenance window is a 60 minute period. Valid values for ddd are:

- sun
- mon
- tue
- wed
- thu
- fri
- sat

Example: sun:05:00-sun:09:00

- **TopicArn** (*string*) –

The Amazon Resource Name (ARN) for the topic used by the source cache cluster for publishing notifications.

- **Port** (*integer*) –

The port number used by each cache nodes in the source cache cluster.

- **CacheParameterGroupName** (*string*) –

The cache parameter group that is associated with the source cache cluster.

- **CacheSubnetGroupName** (*string*) –

The name of the cache subnet group associated with the source cache cluster.

- **VpcId** (*string*) –

The Amazon Virtual Private Cloud identifier (VPC ID) of the cache subnet group for the source cache cluster.

- **AutoMinorVersionUpgrade** (*boolean*) –

This parameter is currently disabled.

- **SnapshotRetentionLimit** (*integer*) –

For an automatic snapshot, the number of days for which ElastiCache will retain the snapshot before deleting it.

For manual snapshots, this field reflects the *SnapshotRetentionLimit* for the source cache cluster when the snapshot was created. This field is otherwise ignored: Manual snapshots do not expire, and can only be deleted using the *DeleteSnapshot* action.

Important If the value of *SnapshotRetentionLimit* is set to zero (0), backups are turned off.

- **SnapshotWindow** (*string*) –
The daily time range during which ElastiCache takes daily snapshots of the source cache cluster.
- **NodeSnapshots** (*list*) –
A list of the cache nodes in the source cache cluster.
- (*dict*) –
Represents an individual cache node in a snapshot of a cache cluster.
- **CacheNodeId** (*string*) –
The cache node identifier for the node in the source cache cluster.
- **CacheSize** (*string*) –
The size of the cache on the source cache node.
- **CacheNodeCreateTime** (*datetime*) –
The date and time when the cache node was created in the source cache cluster.
- **SnapshotCreateTime** (*datetime*) –
The date and time when the source node's metadata and cache data set was obtained for the snapshot.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_waiter (*waiter_name*)

list_tags_for_resource (***kwargs*)

The *ListTagsForResource* action lists all cost allocation tags currently on the named resource. A *cost allocation tag* is a key-value pair where the key is case-sensitive and the value is optional. Cost allocation tags can be used to categorize and track your AWS costs.

You can have a maximum of 10 cost allocation tags on an ElastiCache resource. For more information, see [Using Cost Allocation Tags in Amazon ElastiCache](#).

Request Syntax

```
response = client.list_tags_for_resource(
    ResourceName='string'
)
```

Parameters **ResourceName** (*string*) – [REQUIRED]

The name of the resource for which you want the list of tags, for example `arn:aws:elasticache:us-west-2:0123456789:cluster:myCluster`.

Return type dict

Returns

Response Syntax

```
{
    'TagList': [
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
}
```

Response Structure

- (*dict*) –

Represents the output from the *AddTagsToResource* , *ListTagsOnResource* , and *RemoveTagsFromResource* actions.

- **TagList** (*list*) –

A list of cost allocation tags as key-value pairs.

- * (*dict*) –

A cost allocation Tag that can be added to an ElastiCache cluster or replication group. Tags are composed of a Key/Value pair. A tag with a null Value is permitted.

- **Key** (*string*) –

The key for the tag.

- **Value** (*string*) –

The tag's value. May not be null.

modify_cache_cluster (***kwargs*)

The *ModifyCacheCluster* action modifies the settings for a cache cluster. You can use this action to change one or more cluster configuration parameters by specifying the parameters and the new values.

Request Syntax

```
response = client.modify_cache_cluster(
    CacheClusterId='string',
    NumCacheNodes=123,
    CacheNodeIdsToRemove=[
        'string',
    ],
    AZMode='single-az'|'cross-az',
    NewAvailabilityZones=[
        'string',
    ],
)
```

```
,
CacheSecurityGroupNames=[
    'string',
],
SecurityGroupIds=[
    'string',
],
PreferredMaintenanceWindow='string',
NotificationTopicArn='string',
CacheParameterGroupName='string',
NotificationTopicStatus='string',
ApplyImmediately=True|False,
EngineVersion='string',
AutoMinorVersionUpgrade=True|False,
SnapshotRetentionLimit=123,
SnapshotWindow='string'
)
```

Parameters

- **CacheClusterId** (*string*) – [REQUIRED]

The cache cluster identifier. This value is stored as a lowercase string.

- **NumCacheNodes** (*integer*) – The number of cache nodes that the cache cluster should have. If the value for `NumCacheNodes` is greater than the sum of the number of current cache nodes and the number of cache nodes pending creation (which may be zero), then more nodes will be added. If the value is less than the number of existing cache nodes, then nodes will be removed. If the value is equal to the number of current cache nodes, then any pending add or remove requests are canceled.

If you are removing cache nodes, you must use the `CacheNodeIdsToRemove` parameter to provide the IDs of the specific cache nodes to remove.

For clusters running Redis, this value must be 1. For clusters running Memcached, this value must be between 1 and 20.

Note: Adding or removing Memcached cache nodes can be applied immediately or as a pending action. See `ApplyImmediately`. A pending action to modify the number of cache nodes in a cluster during its maintenance window, whether by adding or removing nodes in accordance with the scale out architecture, is not queued. The customer's latest request to add or remove nodes to the cluster overrides any previous pending actions to modify the number of cache nodes in the cluster. For example, a request to remove 2 nodes would override a previous pending action to remove 3 nodes. Similarly, a request to add 2 nodes would override a previous pending action to remove 3 nodes and vice versa. As Memcached cache nodes may now be provisioned in different Availability Zones with flexible cache node placement, a request to add nodes does not automatically override a previous pending action to add nodes. The customer can modify the previous pending action to add more nodes or explicitly cancel the pending request and retry the new request. To cancel pending actions to modify the number of cache nodes in a cluster, use the `ModifyCacheCluster` request and set `NumCacheNodes` equal to the number of cache nodes currently in the cache cluster.

- **CacheNodeIdsToRemove** (*list*) – A list of cache node IDs to be removed. A node ID is a numeric identifier (0001, 0002, etc.). This parameter is only valid when `NumCacheNodes` is less than the existing number of cache nodes. The

number of cache node IDs supplied in this parameter must match the difference between the existing number of cache nodes in the cluster or pending cache nodes, whichever is greater, and the value of *NumCacheNodes* in the request.

For example: If you have 3 active cache nodes, 7 pending cache nodes, and the number of cache nodes in this `ModifyCacheCluster` call is 5, you must list 2 (7 - 5) cache node IDs to remove.

– (*string*) –

- **AZMode** (*string*) – Specifies whether the new nodes in this Memcached cache cluster are all created in a single Availability Zone or created across multiple Availability Zones.

Valid values: `single-az` | `cross-az`.

This option is only supported for Memcached cache clusters.

Note: You cannot specify `single-az` if the Memcached cache cluster already has cache nodes in different Availability Zones. If `cross-az` is specified, existing Memcached nodes remain in their current Availability Zone.

Only newly created nodes will be located in different Availability Zones. For instructions on how to move existing Memcached nodes to different Availability Zones, see the **Availability Zone Considerations** section of [Cache Node Considerations for Memcached](#).

-
- **NewAvailabilityZones** (*list*) – The list of Availability Zones where the new Memcached cache nodes will be created.

This parameter is only valid when *NumCacheNodes* in the request is greater than the sum of the number of active cache nodes and the number of cache nodes pending creation (which may be zero). The number of Availability Zones supplied in this list must match the cache nodes being added in this request.

This option is only supported on Memcached clusters.

Scenarios:

- **Scenario 1:** You have 3 active nodes and wish to add 2 nodes. Specify `NumCacheNodes=5` (3 + 2) and optionally specify two Availability Zones for the two new nodes.
- **Scenario 2:** You have 3 active nodes and 2 nodes pending creation (from the scenario 1 call) and want to add 1 more node. Specify `NumCacheNodes=6` ((3 + 2) + 1)

and optionally specify an Availability Zone for the new node. * **Scenario 3:** You want to cancel all pending actions. Specify `NumCacheNodes=3` to cancel all pending actions.

The Availability Zone placement of nodes pending creation cannot be modified. If you wish to cancel any nodes pending creation, add 0 nodes by setting `NumCacheNodes` to the number of current nodes.

If `cross-az` is specified, existing Memcached nodes remain in their current Availability Zone. Only newly created nodes can be located in different Availability Zones. For guidance on how to move existing Memcached nodes to different Availability Zones, see the **Availability Zone Considerations** section of [Cache Node Considerations for Memcached](#).

Impact of new add/remove requests upon pending requests

Scenarios Pending action New Request Results Scenario-1 Delete
Delete The new delete, pending or immediate, replaces the pend-

ing delete. Scenario-2 Delete Create The new create, pending or immediate, replaces the pending delete. Scenario-3 Create Delete The new delete, pending or immediate, replaces the pending create. Scenario-4 Create Create The new create is added to the pending create. **Important:** If the new create request is **Apply Immediately - Yes**, all creates are performed immediately. If the new create request is **Apply Immediately - No**, all creates are pending.

Example: `NewAvailabilityZones.member.1=us-west-2aNewAvailabilityZones.me`

– (*string*) –

- **CacheSecurityGroupNames** (*list*) – A list of cache security group names to authorize on this cache cluster. This change is asynchronously applied as soon as possible.

This parameter can be used only with clusters that are created outside of an Amazon Virtual Private Cloud (VPC).

Constraints: Must contain no more than 255 alphanumeric characters. Must not be “Default”.

– (*string*) –

- **SecurityGroupIds** (*list*) – Specifies the VPC Security Groups associated with the cache cluster.

This parameter can be used only with clusters that are created in an Amazon Virtual Private Cloud (VPC).

– (*string*) –

- **PreferredMaintenanceWindow** (*string*) – Specifies the weekly time range during which maintenance on the cache cluster is performed. It is specified as a range in the format `ddd:hh24:mi-ddd:hh24:mi` (24H Clock UTC). The minimum maintenance window is a 60 minute period. Valid values for `ddd` are:

– sun
– mon
– tue
– wed
– thu
– fri
– sat

Example: `sun:05:00-sun:09:00`

- **NotificationTopicArn** (*string*) – The Amazon Resource Name (ARN) of the Amazon SNS topic to which notifications will be sent.

Note: The Amazon SNS topic owner must be same as the cache cluster owner.

- **CacheParameterGroupName** (*string*) – The name of the cache parameter group to apply to this cache cluster. This change is asynchronously applied as soon as possible for parameters when the *ApplyImmediately* parameter is specified as *true* for this request.
- **NotificationTopicStatus** (*string*) – The status of the Amazon SNS notification topic. Notifications are sent only if the status is *active*.

Valid values: `active|inactive`

- **ApplyImmediately** (*boolean*) – If `true`, this parameter causes the modifications in this request and any pending modifications to be applied, asynchronously and as soon as possible, regardless of the *PreferredMaintenanceWindow* setting for the cache cluster.

If `false`, then changes to the cache cluster are applied on the next maintenance reboot, or the next failure reboot, whichever occurs first.

Warning: If you perform a `ModifyCacheCluster` before a pending modification is applied, the pending modification is replaced by the newer modification.

Valid values: `true` | `false`

Default: `false`

- **EngineVersion** (*string*) – The upgraded version of the cache engine to be run on the cache nodes.
- **AutoMinorVersionUpgrade** (*boolean*) – This parameter is currently disabled.
- **SnapshotRetentionLimit** (*integer*) – The number of days for which ElastiCache will retain automatic cache cluster snapshots before deleting them. For example, if you set *SnapshotRetentionLimit* to 5, then a snapshot that was taken today will be retained for 5 days before being deleted.

Important If the value of *SnapshotRetentionLimit* is set to zero (0), backups are turned off.

- **SnapshotWindow** (*string*) – The daily time range (in UTC) during which ElastiCache will begin taking a daily snapshot of your cache cluster.

Return type dict

Returns

Response Syntax

```
{
  'CacheCluster': {
    'CacheClusterId': 'string',
    'ConfigurationEndpoint': {
      'Address': 'string',
      'Port': 123
    },
    'ClientDownloadLandingPage': 'string',
    'CacheNodeType': 'string',
    'Engine': 'string',
    'EngineVersion': 'string',
    'CacheClusterStatus': 'string',
    'NumCacheNodes': 123,
    'PreferredAvailabilityZone': 'string',
    'CacheClusterCreateTime': datetime(2015, 1, 1),
    'PreferredMaintenanceWindow': 'string',
    'PendingModifiedValues': {
      'NumCacheNodes': 123,
      'CacheNodeIdsToRemove': [
        'string',
      ],
      'EngineVersion': 'string'
    },
    'NotificationConfiguration': {
      'TopicArn': 'string',
      'TopicStatus': 'string'
    },
    'CacheSecurityGroups': [
      {
        'CacheSecurityGroupName': 'string',
        'Status': 'string'
      },
    ],
  },
}
```

```
'CacheParameterGroup': {
    'CacheParameterGroupName': 'string',
    'ParameterApplyStatus': 'string',
    'CacheNodeIdsToReboot': [
        'string',
    ]
},
'CacheSubnetGroupName': 'string',
'CacheNodes': [
    {
        'CacheNodeId': 'string',
        'CacheNodeStatus': 'string',
        'CacheNodeCreateTime': datetime(2015, 1, 1),
        'Endpoint': {
            'Address': 'string',
            'Port': 123
        },
        'ParameterGroupStatus': 'string',
        'SourceCacheNodeId': 'string',
        'CustomerAvailabilityZone': 'string'
    },
],
'AutoMinorVersionUpgrade': True|False,
'SecurityGroups': [
    {
        'SecurityGroupId': 'string',
        'Status': 'string'
    },
],
'ReplicationGroupId': 'string',
'SnapshotRetentionLimit': 123,
'SnapshotWindow': 'string'
}
```

Response Structure

- (*dict*) –

Contains all of the attributes of a specific cache cluster.

- **CacheCluster** (*dict*) –

Contains all of the attributes of a specific cache cluster.

- * **CacheClusterId** (*string*) –

The user-supplied identifier of the cache cluster. This identifier is a unique key that identifies a cache cluster.

- * **ConfigurationEndpoint** (*dict*) –

Represents the information required for client programs to connect to a cache node.

- **Address** (*string*) –

The DNS hostname of the cache node.

- **Port** (*integer*) –

The port number that the cache engine is listening on.

- * **ClientDownloadLandingPage** (*string*) –

The URL of the web page where you can download the latest ElastiCache client library.

* **CacheNodeType** (*string*) –

The name of the compute and memory capacity node type for the cache cluster.

Valid node types are as follows:

- General purpose: * Current generation: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
- Previous generation: `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
- Compute optimized: `cache.c1.xlarge`
- Memory optimized * Current generation: `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`, `cache.r3.4xlarge`, `cache.r3.8xlarge`
- Previous generation: `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#).

* **Engine** (*string*) –

The name of the cache engine (*memcached* or *redis*) to be used for this cache cluster.

* **EngineVersion** (*string*) –

The version of the cache engine version that is used in this cache cluster.

* **CacheClusterStatus** (*string*) –

The current state of this cache cluster, one of the following values: *available*, *creating*, *deleted*, *deleting*, *incompatible-network*, *modifying*, *rebooting cache cluster nodes*, *restore-failed*, or *snapshotting*.

* **NumCacheNodes** (*integer*) –

The number of cache nodes in the cache cluster.

For clusters running Redis, this value must be 1. For clusters running Memcached, this value must be between 1 and 20.

* **PreferredAvailabilityZone** (*string*) –

The name of the Availability Zone in which the cache cluster is located or “Multiple” if the cache nodes are located in different Availability Zones.

* **CacheClusterCreateTime** (*datetime*) –

The date and time when the cache cluster was created.

* **PreferredMaintenanceWindow** (*string*) –

Specifies the weekly time range during which maintenance on the cache cluster is performed. It is specified as a range in the format ddd:hh24:mi-ddd:hh24:mi (24H Clock UTC). The minimum maintenance window is a 60 minute period. Valid values for ddd are:

- sun
- mon
- tue
- wed
- thu
- fri
- sat

Example: sun:05:00-sun:09:00

* **PendingModifiedValues** (*dict*) –

A group of settings that will be applied to the cache cluster in the future, or that are currently being applied.

- **NumCacheNodes** (*integer*) –

The new number of cache nodes for the cache cluster.

For clusters running Redis, this value must be 1. For clusters running Memcached, this value must be between 1 and 20.

- **CacheNodeIdsToRemove** (*list*) –

A list of cache node IDs that are being removed (or will be removed) from the cache cluster. A node ID is a numeric identifier (0001, 0002, etc.).

- (*string*) –
- **EngineVersion** (*string*) –

The new cache engine version that the cache cluster will run.

* **NotificationConfiguration** (*dict*) –

Describes a notification topic and its status. Notification topics are used for publishing ElastiCache events to subscribers using Amazon Simple Notification Service (SNS).

- **TopicArn** (*string*) –

The Amazon Resource Name (ARN) that identifies the topic.

- **TopicStatus** (*string*) –

The current state of the topic.

* **CacheSecurityGroups** (*list*) –

A list of cache security group elements, composed of name and status sub-elements.

- (*dict*) –

Represents a cache cluster's status within a particular cache security group.

- **CacheSecurityGroupName** (*string*) –

The name of the cache security group.

- **Status** (*string*) –

The membership status in the cache security group. The status changes when a cache security group is modified, or when

the cache security groups assigned to a cache cluster are modified.

* **CacheParameterGroup** (*dict*) –

The status of the cache parameter group.

· **CacheParameterGroupName** (*string*) –

The name of the cache parameter group.

· **ParameterApplyStatus** (*string*) –

The status of parameter updates.

· **CacheNodeIdsToReboot** (*list*) –

A list of the cache node IDs which need to be rebooted for parameter changes to be applied. A node ID is a numeric identifier (0001, 0002, etc.).

· (*string*) –

* **CacheSubnetGroupName** (*string*) –

The name of the cache subnet group associated with the cache cluster.

* **CacheNodes** (*list*) –

A list of cache nodes that are members of the cache cluster.

· (*dict*) –

Represents an individual cache node within a cache cluster. Each cache node runs its own instance of the cluster's protocol-compliant caching software - either Memcached or Redis.

Valid node types are as follows:

- General purpose: * Current generation: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
- Previous generation: `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
- Compute optimized: `cache.c1.xlarge`
- Memory optimized * Current generation: `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`, `cache.r3.4xlarge`, `cache.r3.8xlarge`
- Previous generation: `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#).

· **CacheNodeId** (*string*) –

The cache node identifier. A node ID is a numeric identifier (0001, 0002, etc.). The combination of cluster ID and node ID uniquely identifies every cache node used in a customer's AWS account.

- **CacheNodeStatus** (*string*) –

The current state of this cache node.

- **CacheNodeCreateTime** (*datetime*) –

The date and time when the cache node was created.

- **Endpoint** (*dict*) –

The hostname for connecting to this cache node.

- **Address** (*string*) –

The DNS hostname of the cache node.

- **Port** (*integer*) –

The port number that the cache engine is listening on.

- **ParameterGroupStatus** (*string*) –

The status of the parameter group applied to this cache node.

- **SourceCacheNodeId** (*string*) –

The ID of the primary node to which this read replica node is synchronized. If this field is empty, then this node is not associated with a primary cache cluster.

- **CustomerAvailabilityZone** (*string*) –

The Availability Zone where this node was created and now resides.

- * **AutoMinorVersionUpgrade** (*boolean*) –

This parameter is currently disabled.

- * **SecurityGroups** (*list*) –

A list of VPC Security Groups associated with the cache cluster.

- (*dict*) –

Represents a single cache security group and its status.

- **SecurityGroupId** (*string*) –

The identifier of the cache security group.

- **Status** (*string*) –

The status of the cache security group membership. The status changes whenever a cache security group is modified, or when the cache security groups assigned to a cache cluster are modified.

- * **ReplicationGroupId** (*string*) –

The replication group to which this cache cluster belongs. If this field is empty, the cache cluster is not associated with any replication group.

- * **SnapshotRetentionLimit** (*integer*) –

The number of days for which ElastiCache will retain automatic cache cluster snapshots before deleting them. For example, if you set *SnapshotRetentionLimit* to 5, then a snapshot that was taken today will be retained for 5 days before being deleted.

Important If the value of `SnapshotRetentionLimit` is set to zero (0), backups are turned off.

* **SnapshotWindow** (*string*) –

The daily time range (in UTC) during which ElastiCache will begin taking a daily snapshot of your cache cluster.

Example: 05:00-09:00

modify_cache_parameter_group (***kwargs*)

The *ModifyCacheParameterGroup* action modifies the parameters of a cache parameter group. You can modify up to 20 parameters in a single request by submitting a list parameter name and value pairs.

Request Syntax

```
response = client.modify_cache_parameter_group(
    CacheParameterGroupName='string',
    ParameterNameValues=[
        {
            'ParameterName': 'string',
            'ParameterValue': 'string'
        },
    ]
)
```

Parameters

- **CacheParameterGroupName** (*string*) – **[REQUIRED]**

The name of the cache parameter group to modify.

- **ParameterNameValues** (*list*) – **[REQUIRED]**

An array of parameter names and values for the parameter update. You must supply at least one parameter name and value; subsequent arguments are optional. A maximum of 20 parameters may be modified per request.

– (*dict*) –

Describes a name-value pair that is used to update the value of a parameter.

* **ParameterName** (*string*) –

The name of the parameter.

* **ParameterValue** (*string*) –

The value of the parameter.

Return type dict

Returns

Response Syntax

```
{
    'CacheParameterGroupName': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of one of the following actions:

- *ModifyCacheParameterGroup*
- *ResetCacheParameterGroup*
- **CacheParameterGroupName** (*string*) –
The name of the cache parameter group.

modify_cache_subnet_group (***kwargs*)

The *ModifyCacheSubnetGroup* action modifies an existing cache subnet group.

Request Syntax

```
response = client.modify_cache_subnet_group(
    CacheSubnetGroupName='string',
    CacheSubnetGroupDescription='string',
    SubnetIds=[
        'string',
    ]
)
```

Parameters

- **CacheSubnetGroupName** (*string*) – **[REQUIRED]**
The name for the cache subnet group. This value is stored as a lowercase string.
Constraints: Must contain no more than 255 alphanumeric characters or hyphens.
Example: mysubnetgroup
- **CacheSubnetGroupDescription** (*string*) – A description for the cache subnet group.
- **SubnetIds** (*list*) – The EC2 subnet IDs for the cache subnet group.
 - (*string*) –

Return type dict

Returns**Response Syntax**

```
{
  'CacheSubnetGroup': {
    'CacheSubnetGroupName': 'string',
    'CacheSubnetGroupDescription': 'string',
    'VpcId': 'string',
    'Subnets': [
      {
        'SubnetIdentifier': 'string',
        'SubnetAvailabilityZone': {
          'Name': 'string'
        }
      },
    ]
  }
}
```

Response Structure

- (*dict*) –
Represents the output of one of the following actions:
 - *CreateCacheSubnetGroup*
 - *ModifyCacheSubnetGroup*
 - **CacheSubnetGroup** (*dict*) –
Represents the output of one of the following actions:
 - * *CreateCacheSubnetGroup*
 - * *ModifyCacheSubnetGroup*

- * **CacheSubnetGroupName** (*string*) –
The name of the cache subnet group.
- * **CacheSubnetGroupDescription** (*string*) –
The description of the cache subnet group.
- * **VpcId** (*string*) –
The Amazon Virtual Private Cloud identifier (VPC ID) of the cache subnet group.
- * **Subnets** (*list*) –
A list of subnets associated with the cache subnet group.
 - (*dict*) –
Represents the subnet associated with a cache cluster. This parameter refers to subnets defined in Amazon Virtual Private Cloud (Amazon VPC) and used with ElastiCache.
 - **SubnetIdentifier** (*string*) –
The unique identifier for the subnet.
 - **SubnetAvailabilityZone** (*dict*) –
The Availability Zone associated with the subnet.
 - **Name** (*string*) –
The name of the Availability Zone.

modify_replication_group (***kwargs*)

The *ModifyReplicationGroup* action modifies the settings for a replication group.

Request Syntax

```
response = client.modify_replication_group(
    ReplicationGroupId='string',
    ReplicationGroupDescription='string',
    PrimaryClusterId='string',
    SnapshottingClusterId='string',
    AutomaticFailoverEnabled=True|False,
    CacheSecurityGroupNames=[
        'string',
    ],
    SecurityGroupIds=[
        'string',
    ],
    PreferredMaintenanceWindow='string',
    NotificationTopicArn='string',
    CacheParameterGroupName='string',
    NotificationTopicStatus='string',
    ApplyImmediately=True|False,
    EngineVersion='string',
    AutoMinorVersionUpgrade=True|False,
    SnapshotRetentionLimit=123,
    SnapshotWindow='string'
)
```

Parameters

- **ReplicationGroupId** (*string*) – [REQUIRED]
The identifier of the replication group to modify.

- **ReplicationGroupDescription** (*string*) – A description for the replication group. Maximum length is 255 characters.
- **PrimaryClusterId** (*string*) – If this parameter is specified, ElastiCache will promote each of the cache clusters in the specified replication group to the primary role. The nodes of all other cache clusters in the replication group will be read replicas.
- **SnapshottingClusterId** (*string*) – The cache cluster ID that will be used as the daily snapshot source for the replication group.
- **AutomaticFailoverEnabled** (*boolean*) – Whether a read replica will be automatically promoted to read/write primary if the existing primary encounters a failure.

Valid values: `true` | `false`

Note: ElastiCache Multi-AZ replication groups are not supported on:

- Redis versions earlier than 2.8.6.
- T1 and T2 cache node types.

- **CacheSecurityGroupNames** (*list*) – A list of cache security group names to authorize for the clusters in this replication group. This change is asynchronously applied as soon as possible.

This parameter can be used only with replication group containing cache clusters running outside of an Amazon Virtual Private Cloud (VPC).

Constraints: Must contain no more than 255 alphanumeric characters. Must not be “Default”.

– (*string*) –

- **SecurityGroupIds** (*list*) – Specifies the VPC Security Groups associated with the cache clusters in the replication group.

This parameter can be used only with replication group containing cache clusters running in an Amazon Virtual Private Cloud (VPC).

– (*string*) –

- **PreferredMaintenanceWindow** (*string*) – Specifies the weekly time range during which maintenance on the cache cluster is performed. It is specified as a range in the format `ddd:hh24:mi-ddd:hh24:mi` (24H Clock UTC). The minimum maintenance window is a 60 minute period. Valid values for `ddd` are:

- `sun`
- `mon`
- `tue`
- `wed`
- `thu`
- `fri`
- `sat`

Example: `sun:05:00-sun:09:00`

- **NotificationTopicArn** (*string*) – The Amazon Resource Name (ARN) of the Amazon SNS topic to which notifications will be sent.

Note: The Amazon SNS topic owner must be same as the replication group owner.

- **CacheParameterGroupName** (*string*) – The name of the cache parameter group to apply to all of the clusters in this replication group. This change is asynchronously applied as soon as possible for parameters when the *ApplyImmediately* parameter is specified as *true* for this request.
- **NotificationTopicStatus** (*string*) – The status of the Amazon SNS notification

topic for the replication group. Notifications are sent only if the status is *active* .

Valid values: `active` | `inactive`

- **ApplyImmediately** (*boolean*) – If `true` , this parameter causes the modifications in this request and any pending modifications to be applied, asynchronously and as soon as possible, regardless of the *PreferredMaintenanceWindow* setting for the replication group.

If `false` , then changes to the nodes in the replication group are applied on the next maintenance reboot, or the next failure reboot, whichever occurs first.

Valid values: `true` | `false`

Default: `false`

- **EngineVersion** (*string*) – The upgraded version of the cache engine to be run on the cache clusters in the replication group.
- **AutoMinorVersionUpgrade** (*boolean*) – This parameter is currently disabled.
- **SnapshotRetentionLimit** (*integer*) – The number of days for which ElastiCache will retain automatic node group snapshots before deleting them. For example, if you set *SnapshotRetentionLimit* to 5, then a snapshot that was taken today will be retained for 5 days before being deleted.

Important If the value of *SnapshotRetentionLimit* is set to zero (0), backups are turned off.

- **SnapshotWindow** (*string*) – The daily time range (in UTC) during which ElastiCache will begin taking a daily snapshot of the node group specified by *SnapshottingClusterId* .

Example: `05:00–09:00`

If you do not specify this parameter, then ElastiCache will automatically choose an appropriate time range.

Return type `dict`

Returns

Response Syntax

```
{
  'ReplicationGroup': {
    'ReplicationGroupId': 'string',
    'Description': 'string',
    'Status': 'string',
    'PendingModifiedValues': {
      'PrimaryClusterId': 'string',
      'AutomaticFailoverStatus': 'enabled'|'disabled'
    },
    'MemberClusters': [
      'string',
    ],
    'NodeGroups': [
      {
        'NodeGroupId': 'string',
        'Status': 'string',
        'PrimaryEndpoint': {
          'Address': 'string',
          'Port': 123
        },
        'NodeGroupMembers': [
          {
```

```
        'CacheClusterId': 'string',
        'CacheNodeId': 'string',
        'ReadEndpoint': {
            'Address': 'string',
            'Port': 123
        },
        'PreferredAvailabilityZone': 'string',
        'CurrentRole': 'string'
    },
]
},
],
'SnapshottingClusterId': 'string',
'AutomaticFailover': 'enabled'|'disabled'|'enabling'|'disabling'
}
```

Response Structure

- (*dict*) –

Contains all of the attributes of a specific replication group.

- **ReplicationGroup** (*dict*) –

Contains all of the attributes of a specific replication group.

- * **ReplicationGroupId** (*string*) –

The identifier for the replication group.

- * **Description** (*string*) –

The description of the replication group.

- * **Status** (*string*) –

The current state of this replication group - *creating* , *available* , etc.

- * **PendingModifiedValues** (*dict*) –

A group of settings to be applied to the replication group, either immediately or during the next maintenance window.

- **PrimaryClusterId** (*string*) –

The primary cluster ID which will be applied immediately (if `--apply-immediately` was specified), or during the next maintenance window.

- **AutomaticFailoverStatus** (*string*) –

Indicates the status of Multi-AZ for this replication group.

Note: ElastiCache Multi-AZ replication groups are not supported on:

- Redis versions earlier than 2.8.6.
 - T1 and T2 cache node types.
-

- * **MemberClusters** (*list*) –

The names of all the cache clusters that are part of this replication group.

- (*string*) –

- * **NodeGroups** (*list*) –

A single element list with information about the nodes in the replication group.

- (*dict*) –

Represents a collection of cache nodes in a replication group.

- **NodeGroupId** (*string*) –

The identifier for the node group. A replication group contains only one node group; therefore, the node group ID is 0001.

- **Status** (*string*) –

The current state of this replication group - *creating* , *available* , etc.

- **PrimaryEndpoint** (*dict*) –

Represents the information required for client programs to connect to a cache node.

- **Address** (*string*) –

The DNS hostname of the cache node.

- **Port** (*integer*) –

The port number that the cache engine is listening on.

- **NodeGroupMembers** (*list*) –

A list containing information about individual nodes within the node group.

- (*dict*) –

Represents a single node within a node group.

- **CacheClusterId** (*string*) –

The ID of the cache cluster to which the node belongs.

- **CacheNodeId** (*string*) –

The ID of the node within its cache cluster. A node ID is a numeric identifier (0001, 0002, etc.).

- **ReadEndpoint** (*dict*) –

Represents the information required for client programs to connect to a cache node.

- **Address** (*string*) –

The DNS hostname of the cache node.

- **Port** (*integer*) –

The port number that the cache engine is listening on.

- **PreferredAvailabilityZone** (*string*) –

The name of the Availability Zone in which the node is located.

- **CurrentRole** (*string*) –

The role that is currently assigned to the node - *primary* or *replica* .

- * **SnapshottingClusterId** (*string*) –

The cache cluster ID that is used as the daily snapshot source for the replication group.

- * **AutomaticFailover** (*string*) –

Indicates the status of Multi-AZ for this replication group.

Note: ElastiCache Multi-AZ replication groups are not supported on:

- Redis versions earlier than 2.8.6.
 - T1 and T2 cache node types.
-

purchase_reserved_cache_nodes_offering (**kwargs)

The *PurchaseReservedCacheNodesOffering* action allows you to purchase a reserved cache node offering.

Request Syntax

```
response = client.purchase_reserved_cache_nodes_offering(  
    ReservedCacheNodesOfferingId='string',  
    ReservedCacheNodeId='string',  
    CacheNodeCount=123  
)
```

Parameters

- **ReservedCacheNodesOfferingId** (*string*) – [REQUIRED]

The ID of the reserved cache node offering to purchase.

Example: 438012d3-4052-4cc7-b2e3-8d3372e0e706

- **ReservedCacheNodeId** (*string*) – A customer-specified identifier to track this reservation.

Example: myreservationID

- **CacheNodeCount** (*integer*) – The number of cache node instances to reserve.

Default: 1

Return type dict

Returns

Response Syntax

```
{  
    'ReservedCacheNode': {  
        'ReservedCacheNodeId': 'string',  
        'ReservedCacheNodesOfferingId': 'string',  
        'CacheNodeType': 'string',  
        'StartTime': datetime(2015, 1, 1),  
        'Duration': 123,  
        'FixedPrice': 123.0,  
        'UsagePrice': 123.0,  
        'CacheNodeCount': 123,  
        'ProductDescription': 'string',  
        'OfferingType': 'string',  
        'State': 'string',  
        'RecurringCharges': [  
            {  
                'RecurringChargeAmount': 123.0,  
                'RecurringChargeFrequency': 'string'  
            },  
        ],  
    }  
}
```

Response Structure

- (*dict*) –

Represents the output of a *PurchaseReservedCacheNodesOffering* action.

- **ReservedCacheNode** (*dict*) –

Represents the output of a *PurchaseReservedCacheNodesOffering* action.

* **ReservedCacheNodeId** (*string*) –

The unique identifier for the reservation.

* **ReservedCacheNodesOfferingId** (*string*) –

The offering identifier.

* **CacheNodeType** (*string*) –

The cache node type for the reserved cache nodes.

Valid node types are as follows:

- General purpose: * Current generation: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
- Previous generation: `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
- Compute optimized: `cache.c1.xlarge`
- Memory optimized * Current generation: `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`, `cache.r3.4xlarge`, `cache.r3.8xlarge`
- Previous generation: `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#).

* **StartTime** (*datetime*) –

The time the reservation started.

* **Duration** (*integer*) –

The duration of the reservation in seconds.

* **FixedPrice** (*float*) –

The fixed price charged for this reserved cache node.

* **UsagePrice** (*float*) –

The hourly price charged for this reserved cache node.

* **CacheNodeCount** (*integer*) –

The number of cache nodes that have been reserved.

* **ProductDescription** (*string*) –

The description of the reserved cache node.

* **OfferingType** (*string*) –

The offering type of this reserved cache node.

* **State** (*string*) –

The state of the reserved cache node.

- * **RecurringCharges** (*list*) –

The recurring price charged to run this reserved cache node.

- (*dict*) –

Contains the specific price and frequency of a recurring charges for a reserved cache node, or for a reserved cache node offering.

- **RecurringChargeAmount** (*float*) –

The monetary amount of the recurring charge.

- **RecurringChargeFrequency** (*string*) –

The frequency of the recurring charge.

reboot_cache_cluster (***kwargs*)

The *RebootCacheCluster* action reboots some, or all, of the cache nodes within a provisioned cache cluster. This API will apply any modified cache parameter groups to the cache cluster. The reboot action takes place as soon as possible, and results in a momentary outage to the cache cluster. During the reboot, the cache cluster status is set to REBOOTING.

The reboot causes the contents of the cache (for each cache node being rebooted) to be lost.

When the reboot is complete, a cache cluster event is created.

Request Syntax

```
response = client.reboot_cache_cluster(
    CacheClusterId='string',
    CacheNodeIdsToReboot=[
        'string',
    ]
)
```

Parameters

- **CacheClusterId** (*string*) – [REQUIRED]

The cache cluster identifier. This parameter is stored as a lowercase string.

- **CacheNodeIdsToReboot** (*list*) – [REQUIRED]

A list of cache node IDs to reboot. A node ID is a numeric identifier (0001, 0002, etc.). To reboot an entire cache cluster, specify all of the cache node IDs.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'CacheCluster': {
    'CacheClusterId': 'string',
    'ConfigurationEndpoint': {
      'Address': 'string',
      'Port': 123
    },
  },
  'ClientDownloadLandingPage': 'string',
  'CacheNodeType': 'string',
  'Engine': 'string',
  'EngineVersion': 'string',
  'CacheClusterStatus': 'string',
  'NumCacheNodes': 123,
```



```

'PreferredAvailabilityZone': 'string',
'CacheClusterCreateTime': datetime(2015, 1, 1),
'PreferredMaintenanceWindow': 'string',
'PendingModifiedValues': {
    'NumCacheNodes': 123,
    'CacheNodeIdsToRemove': [
        'string',
    ],
    'EngineVersion': 'string'
},
'NotificationConfiguration': {
    'TopicArn': 'string',
    'TopicStatus': 'string'
},
'CacheSecurityGroups': [
    {
        'CacheSecurityGroupName': 'string',
        'Status': 'string'
    },
],
'CacheParameterGroup': {
    'CacheParameterGroupName': 'string',
    'ParameterApplyStatus': 'string',
    'CacheNodeIdsToReboot': [
        'string',
    ]
},
'CacheSubnetGroupName': 'string',
'CacheNodes': [
    {
        'CacheNodeId': 'string',
        'CacheNodeStatus': 'string',
        'CacheNodeCreateTime': datetime(2015, 1, 1),
        'Endpoint': {
            'Address': 'string',
            'Port': 123
        },
        'ParameterGroupStatus': 'string',
        'SourceCacheNodeId': 'string',
        'CustomerAvailabilityZone': 'string'
    },
],
'AutoMinorVersionUpgrade': True|False,
'SecurityGroups': [
    {
        'SecurityGroupId': 'string',
        'Status': 'string'
    },
],
'ReplicationGroupId': 'string',
'SnapshotRetentionLimit': 123,
'SnapshotWindow': 'string'
}
}

```

Response Structure

- (*dict*) –

Contains all of the attributes of a specific cache cluster.

– **CacheCluster** (*dict*) –

Contains all of the attributes of a specific cache cluster.

* **CacheClusterId** (*string*) –

The user-supplied identifier of the cache cluster. This identifier is a unique key that identifies a cache cluster.

* **ConfigurationEndpoint** (*dict*) –

Represents the information required for client programs to connect to a cache node.

· **Address** (*string*) –

The DNS hostname of the cache node.

· **Port** (*integer*) –

The port number that the cache engine is listening on.

* **ClientDownloadLandingPage** (*string*) –

The URL of the web page where you can download the latest ElastiCache client library.

* **CacheNodeType** (*string*) –

The name of the compute and memory capacity node type for the cache cluster.

Valid node types are as follows:

- General purpose: * Current generation: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
- Previous generation: `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
- Compute optimized: `cache.c1.xlarge`
- Memory optimized * Current generation: `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`, `cache.r3.4xlarge`, `cache.r3.8xlarge`
- Previous generation: `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#).

* **Engine** (*string*) –

The name of the cache engine (*memcached* or *redis*) to be used for this cache cluster.

* **EngineVersion** (*string*) –

The version of the cache engine version that is used in this cache cluster.

* **CacheClusterStatus** (*string*) –

The current state of this cache cluster, one of the following values: *available* , *creating* , *deleted* , *deleting* , *incompatible-network* , *modifying* , *rebooting cache cluster nodes* , *restore-failed* , or *snapshotting* .

* **NumCacheNodes** (*integer*) –

The number of cache nodes in the cache cluster.

For clusters running Redis, this value must be 1. For clusters running Memcached, this value must be between 1 and 20.

* **PreferredAvailabilityZone** (*string*) –

The name of the Availability Zone in which the cache cluster is located or “Multiple” if the cache nodes are located in different Availability Zones.

* **CacheClusterCreateTime** (*datetime*) –

The date and time when the cache cluster was created.

* **PreferredMaintenanceWindow** (*string*) –

Specifies the weekly time range during which maintenance on the cache cluster is performed. It is specified as a range in the format ddd:hh24:mi-ddd:hh24:mi (24H Clock UTC). The minimum maintenance window is a 60 minute period. Valid values for ddd are:

- sun
- mon
- tue
- wed
- thu
- fri
- sat

Example: sun:05:00-sun:09:00

* **PendingModifiedValues** (*dict*) –

A group of settings that will be applied to the cache cluster in the future, or that are currently being applied.

- **NumCacheNodes** (*integer*) –

The new number of cache nodes for the cache cluster.

For clusters running Redis, this value must be 1. For clusters running Memcached, this value must be between 1 and 20.

- **CacheNodeIdsToRemove** (*list*) –

A list of cache node IDs that are being removed (or will be removed) from the cache cluster. A node ID is a numeric identifier (0001, 0002, etc.).

- (*string*) –

- **EngineVersion** (*string*) –

The new cache engine version that the cache cluster will run.

* **NotificationConfiguration** (*dict*) –

Describes a notification topic and its status. Notification topics are used for publishing ElastiCache events to subscribers using Amazon Simple Notification Service (SNS).

- **TopicArn** (*string*) –

The Amazon Resource Name (ARN) that identifies the topic.

- **TopicStatus** (*string*) –

The current state of the topic.

- * **CacheSecurityGroups** (*list*) –

A list of cache security group elements, composed of name and status sub-elements.

- (*dict*) –

Represents a cache cluster's status within a particular cache security group.

- **CacheSecurityGroupName** (*string*) –

The name of the cache security group.

- **Status** (*string*) –

The membership status in the cache security group. The status changes when a cache security group is modified, or when the cache security groups assigned to a cache cluster are modified.

- * **CacheParameterGroup** (*dict*) –

The status of the cache parameter group.

- **CacheParameterGroupName** (*string*) –

The name of the cache parameter group.

- **ParameterApplyStatus** (*string*) –

The status of parameter updates.

- **CacheNodeIdsToReboot** (*list*) –

A list of the cache node IDs which need to be rebooted for parameter changes to be applied. A node ID is a numeric identifier (0001, 0002, etc.).

- (*string*) –

- * **CacheSubnetGroupName** (*string*) –

The name of the cache subnet group associated with the cache cluster.

- * **CacheNodes** (*list*) –

A list of cache nodes that are members of the cache cluster.

- (*dict*) –

Represents an individual cache node within a cache cluster. Each cache node runs its own instance of the cluster's protocol-compliant caching software - either Memcached or Redis.

Valid node types are as follows:

- General purpose: * Current generation: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
- Previous generation: `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
- Compute optimized: `cache.c1.xlarge`

- **Memory optimized** * **Current generation:**
`cache.r3.large` , `cache.r3.xlarge` ,
`cache.r3.2xlarge` , `cache.r3.4xlarge` ,
`cache.r3.8xlarge`
- **Previous generation:** `cache.m2.xlarge` ,
`cache.m2.2xlarge` , `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#) .

- **CacheNodeId** (*string*) –

The cache node identifier. A node ID is a numeric identifier (0001, 0002, etc.). The combination of cluster ID and node ID uniquely identifies every cache node used in a customer's AWS account.

- **CacheNodeStatus** (*string*) –

The current state of this cache node.

- **CacheNodeCreateTime** (*datetime*) –

The date and time when the cache node was created.

- **Endpoint** (*dict*) –

The hostname for connecting to this cache node.

- **Address** (*string*) –

The DNS hostname of the cache node.

- **Port** (*integer*) –

The port number that the cache engine is listening on.

- **ParameterGroupStatus** (*string*) –

The status of the parameter group applied to this cache node.

- **SourceCacheNodeId** (*string*) –

The ID of the primary node to which this read replica node is synchronized. If this field is empty, then this node is not associated with a primary cache cluster.

- **CustomerAvailabilityZone** (*string*) –

The Availability Zone where this node was created and now resides.

- * **AutoMinorVersionUpgrade** (*boolean*) –

This parameter is currently disabled.

- * **SecurityGroups** (*list*) –

A list of VPC Security Groups associated with the cache cluster.

- (*dict*) –

Represents a single cache security group and its status.

- **SecurityGroupId** (*string*) –

The identifier of the cache security group.

- **Status** (*string*) –

The status of the cache security group membership. The status changes whenever a cache security group is modified, or when the cache security groups assigned to a cache cluster are modified.

- * **ReplicationGroupId** (*string*) –

The replication group to which this cache cluster belongs. If this field is empty, the cache cluster is not associated with any replication group.

- * **SnapshotRetentionLimit** (*integer*) –

The number of days for which ElastiCache will retain automatic cache cluster snapshots before deleting them. For example, if you set *SnapshotRetentionLimit* to 5, then a snapshot that was taken today will be retained for 5 days before being deleted.

Important If the value of *SnapshotRetentionLimit* is set to zero (0), backups are turned off.

- * **SnapshotWindow** (*string*) –

The daily time range (in UTC) during which ElastiCache will begin taking a daily snapshot of your cache cluster.

Example: 05:00-09:00

remove_tags_from_resource (**kwargs)

The *RemoveTagsFromResource* action removes the tags identified by the *TagKeys* list from the named resource.

Request Syntax

```
response = client.remove_tags_from_resource(
    ResourceName='string',
    TagKeys=[
        'string',
    ]
)
```

Parameters

- **ResourceName** (*string*) – [REQUIRED]

The name of the ElastiCache resource from which you want the listed tags removed, for example `arn:aws:elasticache:us-west-2:0123456789:cluster:myCluster`.

- **TagKeys** (*list*) – [REQUIRED]

A list of *TagKeys* identifying the tags you want removed from the named resource. For example, `TagKeys.member.1=Region` removes the cost allocation tag with the key name *Region* from the resource named by the *ResourceName* parameter.

– (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'TagList': [
    {
      'Key': 'string',
      'Value': 'string'
    },
  ]
}
```

Response Structure

- (dict) –

Represents the output from the *AddTagsToResource* , *ListTagsOnResource* , and *RemoveTagsFromResource* actions.

- **TagList** (list) –

A list of cost allocation tags as key-value pairs.

- * (dict) –

A cost allocation Tag that can be added to an ElastiCache cluster or replication group. Tags are composed of a Key/Value pair. A tag with a null Value is permitted.

- **Key** (string) –

The key for the tag.

- **Value** (string) –

The tag's value. May not be null.

reset_cache_parameter_group (**kwargs)

The *ResetCacheParameterGroup* action modifies the parameters of a cache parameter group to the engine or system default value. You can reset specific parameters by submitting a list of parameter names. To reset the entire cache parameter group, specify the *ResetAllParameters* and *CacheParameterGroupName* parameters.

Request Syntax

```
response = client.reset_cache_parameter_group(
    CacheParameterGroupName='string',
    ResetAllParameters=True|False,
    ParameterNameValues=[
        {
            'ParameterName': 'string',
            'ParameterValue': 'string'
        },
    ]
)
```

Parameters

- **CacheParameterGroupName** (string) – [REQUIRED]

The name of the cache parameter group to reset.

- **ResetAllParameters** (boolean) – If *true* , all parameters in the cache parameter group will be reset to default values. If *false* , no such action occurs.

Valid values: `true` | `false`

- **ParameterNameValues** (list) – [REQUIRED]

An array of parameter names to be reset. If you are not resetting the entire cache parameter group, you must specify at least one parameter name.

– (dict) –

Describes a name-value pair that is used to update the value of a parameter.

* **ParameterName** (*string*) –

The name of the parameter.

* **ParameterValue** (*string*) –

The value of the parameter.

Return type dict

Returns

Response Syntax

```
{
    'CacheParameterGroupName': 'string'
}
```

Response Structure

• (dict) –

Represents the output of one of the following actions:

– *ModifyCacheParameterGroup*

– *ResetCacheParameterGroup*

– **CacheParameterGroupName** (*string*) –

The name of the cache parameter group.

revoke_cache_security_group_ingress (**kwargs)

The *RevokeCacheSecurityGroupIngress* action revokes ingress from a cache security group. Use this action to disallow access from an Amazon EC2 security group that had been previously authorized.

Request Syntax

```
response = client.revoke_cache_security_group_ingress(
    CacheSecurityGroupName='string',
    EC2SecurityGroupName='string',
    EC2SecurityGroupOwnerId='string'
)
```

Parameters

• **CacheSecurityGroupName** (*string*) – [REQUIRED]

The name of the cache security group to revoke ingress from.

• **EC2SecurityGroupName** (*string*) – [REQUIRED]

The name of the Amazon EC2 security group to revoke access from.

• **EC2SecurityGroupOwnerId** (*string*) – [REQUIRED]

The AWS account number of the Amazon EC2 security group owner. Note that this is not the same thing as an AWS access key ID - you must provide a valid AWS account number for this parameter.

Return type dict

Returns

Response Syntax

```
{
    'CacheSecurityGroup': {
        'OwnerId': 'string',

```



```

'CacheSecurityGroupName': 'string',
'Description': 'string',
'EC2SecurityGroups': [
    {
        'Status': 'string',
        'EC2SecurityGroupName': 'string',
        'EC2SecurityGroupOwnerId': 'string'
    },
]
}
}

```

Response Structure

- (dict) –

Represents the output of one of the following actions:

- *AuthorizeCacheSecurityGroupIngress*
- *CreateCacheSecurityGroup*
- *RevokeCacheSecurityGroupIngress*
- **CacheSecurityGroup** (dict) –

Represents the output of one of the following actions:

- * *AuthorizeCacheSecurityGroupIngress*
- * *CreateCacheSecurityGroup*
- * *RevokeCacheSecurityGroupIngress*

- * **OwnerId** (string) –

The AWS account ID of the cache security group owner.

- * **CacheSecurityGroupName** (string) –

The name of the cache security group.

- * **Description** (string) –

The description of the cache security group.

- * **EC2SecurityGroups** (list) –

A list of Amazon EC2 security groups that are associated with this cache security group.

- (dict) –

Provides ownership and status information for an Amazon EC2 security group.

- **Status** (string) –

The status of the Amazon EC2 security group.

- **EC2SecurityGroupName** (string) –

The name of the Amazon EC2 security group.

- **EC2SecurityGroupOwnerId** (string) –

The AWS account ID of the Amazon EC2 security group owner.

Paginators

The available paginators are:

- `ElastiCache.Paginator.DescribeCacheClusters`

- `ElastiCache.Paginator.DescribeCacheEngineVersions`
- `ElastiCache.Paginator.DescribeCacheParameterGroups`
- `ElastiCache.Paginator.DescribeCacheParameters`
- `ElastiCache.Paginator.DescribeCacheSecurityGroups`
- `ElastiCache.Paginator.DescribeCacheSubnetGroups`
- `ElastiCache.Paginator.DescribeEngineDefaultParameters`
- `ElastiCache.Paginator.DescribeEvents`
- `ElastiCache.Paginator.DescribeReplicationGroups`
- `ElastiCache.Paginator.DescribeReservedCacheNodes`
- `ElastiCache.Paginator.DescribeReservedCacheNodesOfferings`
- `ElastiCache.Paginator.DescribeSnapshots`

class `ElastiCache.Paginator.DescribeCacheClusters`

```
paginator = client.get_paginator('describe_cache_clusters')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `ElastiCache.Client.describe_cache_clusters()`.

Request Syntax

```
response_iterator = paginator.paginate(
    CacheClusterId='string',
    ShowCacheNodeInfo=True|False,
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **CacheClusterId** (*string*) – The user-supplied cluster identifier. If this parameter is specified, only information about that specific cache cluster is returned. This parameter isn't case sensitive.
- **ShowCacheNodeInfo** (*boolean*) – An optional flag that can be included in the `DescribeCacheCluster` request to retrieve information about the individual cache nodes.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.

– **StartingToken** (*string*) –

A token to specify where to start paginating. This is the NextToken from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'CacheClusters': [
    {
      'CacheClusterId': 'string',
      'ConfigurationEndpoint': {
        'Address': 'string',
        'Port': 123
      },
      'ClientDownloadLandingPage': 'string',
      'CacheNodeType': 'string',
      'Engine': 'string',
      'EngineVersion': 'string',
      'CacheClusterStatus': 'string',
      'NumCacheNodes': 123,
      'PreferredAvailabilityZone': 'string',
      'CacheClusterCreateTime': datetime(2015, 1, 1),
      'PreferredMaintenanceWindow': 'string',
      'PendingModifiedValues': {
        'NumCacheNodes': 123,
        'CacheNodeIdsToRemove': [
          'string',
        ],
        'EngineVersion': 'string'
      },
      'NotificationConfiguration': {
        'TopicArn': 'string',
        'TopicStatus': 'string'
      },
      'CacheSecurityGroups': [
        {
          'CacheSecurityGroupName': 'string',
          'Status': 'string'
        },
      ],
      'CacheParameterGroup': {
        'CacheParameterGroupName': 'string',
        'ParameterApplyStatus': 'string',
        'CacheNodeIdsToReboot': [
          'string',
        ]
      },
      'CacheSubnetGroupName': 'string',
      'CacheNodes': [
        {
          'CacheNodeId': 'string',
          'CacheNodeStatus': 'string',
          'CacheNodeCreateTime': datetime(2015, 1, 1),
          'Endpoint': {
            'Address': 'string',
            'Port': 123
          }
        }
      ]
    }
  ]
}
```

```
        },
        'ParameterGroupStatus': 'string',
        'SourceCacheNodeId': 'string',
        'CustomerAvailabilityZone': 'string'
    },
],
'AutoMinorVersionUpgrade': True|False,
'SecurityGroups': [
    {
        'SecurityGroupId': 'string',
        'Status': 'string'
    },
],
'ReplicationGroupId': 'string',
'SnapshotRetentionLimit': 123,
'SnapshotWindow': 'string'
},
],
'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a *DescribeCacheClusters* action.

- **CacheClusters** (*list*) –

A list of cache clusters. Each item in the list contains detailed information about one cache cluster.

- * (*dict*) –

Contains all of the attributes of a specific cache cluster.

- **CacheClusterId** (*string*) –

The user-supplied identifier of the cache cluster. This identifier is a unique key that identifies a cache cluster.

- **ConfigurationEndpoint** (*dict*) –

Represents the information required for client programs to connect to a cache node.

- **Address** (*string*) –

The DNS hostname of the cache node.

- **Port** (*integer*) –

The port number that the cache engine is listening on.

- **ClientDownloadLandingPage** (*string*) –

The URL of the web page where you can download the latest ElastiCache client library.

- **CacheNodeType** (*string*) –

The name of the compute and memory capacity node type for the cache cluster.

Valid node types are as follows:

- General purpose: * Current generation: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`

- Previous generation: `cache.t1.micro` ,
`cache.m1.small` , `cache.m1.medium` ,
`cache.m1.large` , `cache.m1.xlarge`
- Compute optimized: `cache.c1.xlarge`
- Memory optimized * Current generation:
`cache.r3.large` , `cache.r3.xlarge` ,
`cache.r3.2xlarge` , `cache.r3.4xlarge` ,
`cache.r3.8xlarge`
- Previous generation: `cache.m2.xlarge` ,
`cache.m2.2xlarge` , `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#) .

- **Engine** (*string*) –

The name of the cache engine (*memcached* or *redis*) to be used for this cache cluster.

- **EngineVersion** (*string*) –

The version of the cache engine version that is used in this cache cluster.

- **CacheClusterStatus** (*string*) –

The current state of this cache cluster, one of the following values: *available* , *creating* , *deleted* , *deleting* , *incompatible-network* , *modifying* , *rebooting cache cluster nodes* , *restore-failed* , or *snapshotting* .

- **NumCacheNodes** (*integer*) –

The number of cache nodes in the cache cluster.

For clusters running Redis, this value must be 1. For clusters running Memcached, this value must be between 1 and 20.

- **PreferredAvailabilityZone** (*string*) –

The name of the Availability Zone in which the cache cluster is located or “Multiple” if the cache nodes are located in different Availability Zones.

- **CacheClusterCreateTime** (*datetime*) –

The date and time when the cache cluster was created.

- **PreferredMaintenanceWindow** (*string*) –

Specifies the weekly time range during which maintenance on the cache cluster is performed. It is specified as a range in the format `ddd:hh24:mi-ddd:hh24:mi` (24H Clock UTC). The minimum maintenance window is a 60 minute period. Valid values for `ddd` are:

- `sun`
- `mon`
- `tue`

- wed
- thu
- fri
- sat

Example: sun:05:00-sun:09:00

- **PendingModifiedValues** (*dict*) –

A group of settings that will be applied to the cache cluster in the future, or that are currently being applied.

- **NumCacheNodes** (*integer*) –

The new number of cache nodes for the cache cluster.

For clusters running Redis, this value must be 1. For clusters running Memcached, this value must be between 1 and 20.

- **CacheNodeIdsToRemove** (*list*) –

A list of cache node IDs that are being removed (or will be removed) from the cache cluster. A node ID is a numeric identifier (0001, 0002, etc.).

- (*string*) –

- **EngineVersion** (*string*) –

The new cache engine version that the cache cluster will run.

- **NotificationConfiguration** (*dict*) –

Describes a notification topic and its status. Notification topics are used for publishing ElastiCache events to subscribers using Amazon Simple Notification Service (SNS).

- **TopicArn** (*string*) –

The Amazon Resource Name (ARN) that identifies the topic.

- **TopicStatus** (*string*) –

The current state of the topic.

- **CacheSecurityGroups** (*list*) –

A list of cache security group elements, composed of name and status sub-elements.

- (*dict*) –

Represents a cache cluster's status within a particular cache security group.

- **CacheSecurityGroupName** (*string*) –

The name of the cache security group.

- **Status** (*string*) –

The membership status in the cache security group. The status changes when a cache security group is modified, or when the cache security groups assigned to a cache cluster are modified.

- **CacheParameterGroup** (*dict*) –

The status of the cache parameter group.

- **CacheParameterGroupName** (*string*) –

The name of the cache parameter group.

- **ParameterApplyStatus** (*string*) –

The status of parameter updates.

- **CacheNodeIdsToReboot** (*list*) –

A list of the cache node IDs which need to be rebooted for parameter changes to be applied. A node ID is a numeric identifier (0001, 0002, etc.).

- (*string*) –

- **CacheSubnetGroupName** (*string*) –

The name of the cache subnet group associated with the cache cluster.

- **CacheNodes** (*list*) –

A list of cache nodes that are members of the cache cluster.

- (*dict*) –

Represents an individual cache node within a cache cluster. Each cache node runs its own instance of the cluster's protocol-compliant caching software - either Memcached or Redis.

Valid node types are as follows:

- General purpose: * Current generation: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
- Previous generation: `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
- Compute optimized: `cache.c1.xlarge`
- Memory optimized * Current generation: `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`, `cache.r3.4xlarge`, `cache.r3.8xlarge`
- Previous generation: `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#).

- **CacheNodeId** (*string*) –

The cache node identifier. A node ID is a numeric identifier (0001, 0002, etc.). The combination of cluster ID and node ID uniquely identifies every cache node used in a customer's AWS account.

- **CacheNodeStatus** (*string*) –

The current state of this cache node.

- **CacheNodeCreateTime** (*datetime*) –

The date and time when the cache node was created.

- **Endpoint** (*dict*) –

The hostname for connecting to this cache node.

- **Address** (*string*) –

The DNS hostname of the cache node.

- **Port** (*integer*) –

The port number that the cache engine is listening on.

- **ParameterGroupStatus** (*string*) –

The status of the parameter group applied to this cache node.

- **SourceCacheNodeId** (*string*) –

The ID of the primary node to which this read replica node is synchronized. If this field is empty, then this node is not associated with a primary cache cluster.

- **CustomerAvailabilityZone** (*string*) –

The Availability Zone where this node was created and now resides.

- **AutoMinorVersionUpgrade** (*boolean*) –

This parameter is currently disabled.

- **SecurityGroups** (*list*) –

A list of VPC Security Groups associated with the cache cluster.

- (*dict*) –

Represents a single cache security group and its status.

- **SecurityGroupId** (*string*) –

The identifier of the cache security group.

- **Status** (*string*) –

The status of the cache security group membership. The status changes whenever a cache security group is modified, or when the cache security groups assigned to a cache cluster are modified.

- **ReplicationGroupId** (*string*) –

The replication group to which this cache cluster belongs. If this field is empty, the cache cluster is not associated with any replication group.

- **SnapshotRetentionLimit** (*integer*) –

The number of days for which ElastiCache will retain automatic cache cluster snapshots before deleting them. For example, if you set *SnapshotRetentionLimit* to 5, then a snapshot that was taken today will be retained for 5 days before being deleted.

Important If the value of *SnapshotRetentionLimit* is set to zero (0), backups are turned off.

- **SnapshotWindow** (*string*) –

The daily time range (in UTC) during which ElastiCache will begin taking a daily snapshot of your cache cluster.

Example: 05:00–09:00

- **NextToken** (*string*) –

A token to resume pagination.

`class ElasticCache.Paginator.DescribeCacheEngineVersions`

```
paginator = client.get_paginator('describe_cache_engine_versions')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `ElasticCache.Client.describe_cache_engine_versions()`.

Request Syntax

```
response_iterator = paginator.paginate(
    Engine='string',
    EngineVersion='string',
    CacheParameterGroupFamily='string',
    DefaultOnly=True|False,
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **Engine** (*string*) – The cache engine to return. Valid values: memcached | redis
- **EngineVersion** (*string*) – The cache engine version to return.

Example: 1.4.14

- **CacheParameterGroupFamily** (*string*) – The name of a specific cache parameter group family to return details for.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens
- **DefaultOnly** (*boolean*) – If *true*, specifies that only the default version of the specified engine or engine and major version combination is to be returned.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'CacheEngineVersions': [
    {
      'Engine': 'string',
      'EngineVersion': 'string',
      'CacheParameterGroupFamily': 'string',
      'CacheEngineDescription': 'string',
      'CacheEngineVersionDescription': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- *(dict)* –

Represents the output of a DescribeCacheEngineVersions action.

- **CacheEngineVersions** (*list*) –

A list of cache engine version details. Each element in the list contains detailed information about one cache engine version.

- * *(dict)* –

Provides all of the details about a particular cache engine version.

- **Engine** (*string*) –

The name of the cache engine.

- **EngineVersion** (*string*) –

The version number of the cache engine.

- **CacheParameterGroupFamily** (*string*) –

The name of the cache parameter group family associated with this cache engine.

- **CacheEngineDescription** (*string*) –

The description of the cache engine.

- **CacheEngineVersionDescription** (*string*) –

The description of the cache engine version.

- **NextToken** (*string*) –

A token to resume pagination.

class `ElastiCache.Paginator.DescribeCacheParameterGroups`

```
paginator = client.get_paginator('describe_cache_parameter_groups')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `ElastiCache.Client.describe_cache_parameter_groups()`.

Request Syntax

```
response_iterator = paginator.paginate(
    CacheParameterGroupName='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
```

```

        'StartingToken': 'string'
    }
)

```

Parameters

- **CacheParameterGroupName** (*string*) – The name of a specific cache parameter group to return details for.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```

{
    'CacheParameterGroups': [
        {
            'CacheParameterGroupName': 'string',
            'CacheParameterGroupFamily': 'string',
            'Description': 'string'
        },
    ],
    'NextToken': 'string'
}

```

Response Structure

- (*dict*) –
Represents the output of a *DescribeCacheParameterGroups* action.
 - **CacheParameterGroups** (*list*) –
A list of cache parameter groups. Each element in the list contains detailed information about one cache parameter group.
 - * (*dict*) –
Represents the output of a *CreateCacheParameterGroup* action.
 - **CacheParameterGroupName** (*string*) –
The name of the cache parameter group.
 - **CacheParameterGroupFamily** (*string*) –
The name of the cache parameter group family that this cache parameter group is compatible with.
 - **Description** (*string*) –
The description for this cache parameter group.

– **NextToken** (*string*) –

A token to resume pagination.

class `ElastiCache.Paginator.DescribeCacheParameters`

```
paginator = client.get_paginator('describe_cache_parameters')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `ElastiCache.Client.describe_cache_parameters()`.

Request Syntax

```
response_iterator = paginator.paginate(
    CacheParameterGroupName='string',
    Source='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **CacheParameterGroupName** (*string*) – **[REQUIRED]**

The name of a specific cache parameter group to return details for.

- **Source** (*string*) – The parameter types to return.

Valid values: `user` | `system` | `engine-default`

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
    'Parameters': [
        {
            'ParameterName': 'string',
            'ParameterValue': 'string',
            'Description': 'string',
            'Source': 'string',
            'DataType': 'string',
            'AllowedValues': 'string',
```

```

        'IsModifiable': True|False,
        'MinimumEngineVersion': 'string'
    },
],
'CacheNodeTypeSpecificParameters': [
    {
        'ParameterName': 'string',
        'Description': 'string',
        'Source': 'string',
        'DataType': 'string',
        'AllowedValues': 'string',
        'IsModifiable': True|False,
        'MinimumEngineVersion': 'string',
        'CacheNodeTypeSpecificValues': [
            {
                'CacheNodeType': 'string',
                'Value': 'string'
            },
        ]
    },
]
],
'NextToken': 'string'
}

```

Response Structure

- (dict) –

Represents the output of a *DescribeCacheParameters* action.

- **Parameters** (list) –

A list of Parameter instances.

- * (dict) –

Describes an individual setting that controls some aspect of ElastiCache behavior.

- **ParameterName** (string) –

The name of the parameter.

- **ParameterValue** (string) –

The value of the parameter.

- **Description** (string) –

A description of the parameter.

- **Source** (string) –

The source of the parameter.

- **DataType** (string) –

The valid data type for the parameter.

- **AllowedValues** (string) –

The valid range of values for the parameter.

- **IsModifiable** (boolean) –

Indicates whether (true) or not (false) the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.

- **MinimumEngineVersion** (string) –

The earliest cache engine version to which the parameter can apply.

– **CacheNodeTypeSpecificParameters** (*list*) –

A list of parameters specific to a particular cache node type. Each element in the list contains detailed information about one parameter.

* (*dict*) –

A parameter that has a different value for each cache node type it is applied to. For example, in a Redis cache cluster, a *cache.m1.large* cache node type would have a larger *maxmemory* value than a *cache.m1.small* type.

· **ParameterName** (*string*) –

The name of the parameter.

· **Description** (*string*) –

A description of the parameter.

· **Source** (*string*) –

The source of the parameter value.

· **DataType** (*string*) –

The valid data type for the parameter.

· **AllowedValues** (*string*) –

The valid range of values for the parameter.

· **IsModifiable** (*boolean*) –

Indicates whether (*true*) or not (*false*) the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.

· **MinimumEngineVersion** (*string*) –

The earliest cache engine version to which the parameter can apply.

· **CacheNodeTypeSpecificValues** (*list*) –

A list of cache node types and their corresponding values for this parameter.

· (*dict*) –

A value that applies only to a certain cache node type.

· **CacheNodeType** (*string*) –

The cache node type for which this value applies.

· **Value** (*string*) –

The value for the cache node type.

– **NextToken** (*string*) –

A token to resume pagination.

class `ElastiCache.Paginator.DescribeCacheSecurityGroups`

```
paginator = client.get_paginator('describe_cache_security_groups')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `ElastiCache.Client.describe_cache_security_groups()`.

Request Syntax

```
response_iterator = paginator.paginate(
    CacheSecurityGroupName='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **CacheSecurityGroupName** (*string*) – The name of the cache security group to return details for.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
    'CacheSecurityGroups': [
        {
            'OwnerId': 'string',
            'CacheSecurityGroupName': 'string',
            'Description': 'string',
            'EC2SecurityGroups': [
                {
                    'Status': 'string',
                    'EC2SecurityGroupName': 'string',
                    'EC2SecurityGroupOwnerId': 'string'
                },
            ],
        },
    ],
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
Represents the output of a `DescribeCacheSecurityGroups` action.
 - **CacheSecurityGroups** (*list*) –
A list of cache security groups. Each element in the list contains detailed information about one group.

* (*dict*) –

Represents the output of one of the following actions:

- *AuthorizeCacheSecurityGroupIngress*
- *CreateCacheSecurityGroup*
- *RevokeCacheSecurityGroupIngress*
- **OwnerId** (*string*) –
The AWS account ID of the cache security group owner.
- **CacheSecurityGroupName** (*string*) –
The name of the cache security group.
- **Description** (*string*) –
The description of the cache security group.
- **EC2SecurityGroups** (*list*) –
A list of Amazon EC2 security groups that are associated with this cache security group.
- (*dict*) –
Provides ownership and status information for an Amazon EC2 security group.
- **Status** (*string*) –
The status of the Amazon EC2 security group.
- **EC2SecurityGroupName** (*string*) –
The name of the Amazon EC2 security group.
- **EC2SecurityGroupOwnerId** (*string*) –
The AWS account ID of the Amazon EC2 security group owner.

– **NextToken** (*string*) –

A token to resume pagination.

class `ElastiCache.Paginator.DescribeCacheSubnetGroups`

```
paginator = client.get_paginator('describe_cache_subnet_groups')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `ElastiCache.Client.describe_cache_subnet_groups()`.

Request Syntax

```
response_iterator = paginator.paginate(
    CacheSubnetGroupName='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **CacheSubnetGroupName** (*string*) – The name of the cache subnet group to return details for.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
  'CacheSubnetGroups': [
    {
      'CacheSubnetGroupName': 'string',
      'CacheSubnetGroupDescription': 'string',
      'VpcId': 'string',
      'Subnets': [
        {
          'SubnetIdentifier': 'string',
          'SubnetAvailabilityZone': {
            'Name': 'string'
          }
        }
      ],
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
Represents the output of a `DescribeCacheSubnetGroups` action.
 - **CacheSubnetGroups** (*list*) –
A list of cache subnet groups. Each element in the list contains detailed information about one group.
 - * (*dict*) –
Represents the output of one of the following actions:
 - `CreateCacheSubnetGroup`
 - `ModifyCacheSubnetGroup`
 - **CacheSubnetGroupName** (*string*) –
The name of the cache subnet group.
 - **CacheSubnetGroupDescription** (*string*) –
The description of the cache subnet group.
 - **VpcId** (*string*) –

The Amazon Virtual Private Cloud identifier (VPC ID) of the cache subnet group.

- **Subnets** (*list*) –

A list of subnets associated with the cache subnet group.

- (*dict*) –

Represents the subnet associated with a cache cluster. This parameter refers to subnets defined in Amazon Virtual Private Cloud (Amazon VPC) and used with ElastiCache.

- **SubnetIdentifier** (*string*) –

The unique identifier for the subnet.

- **SubnetAvailabilityZone** (*dict*) –

The Availability Zone associated with the subnet.

- **Name** (*string*) –

The name of the Availability Zone.

- **NextToken** (*string*) –

A token to resume pagination.

class ElastiCache.Paginator.**DescribeEngineDefaultParameters**

```
paginator = client.get_paginator('describe_engine_default_parameters')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `ElastiCache.Client.describe_engine_default_parameters()`.

Request Syntax

```
response_iterator = paginator.paginate(
    CacheParameterGroupFamily='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **CacheParameterGroupFamily** (*string*) – **[REQUIRED]**

The name of the cache parameter group family. Valid values are: `memcached1.4` | `redis2.6` | `redis2.8`

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

– **StartingToken** (*string*) –

A token to specify where to start paginating. This is the NextToken from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'EngineDefaults': {
    'CacheParameterGroupFamily': 'string',
    'Marker': 'string',
    'Parameters': [
      {
        'ParameterName': 'string',
        'ParameterValue': 'string',
        'Description': 'string',
        'Source': 'string',
        'DataType': 'string',
        'AllowedValues': 'string',
        'IsModifiable': True|False,
        'MinimumEngineVersion': 'string'
      },
    ],
    'CacheNodeTypeSpecificParameters': [
      {
        'ParameterName': 'string',
        'Description': 'string',
        'Source': 'string',
        'DataType': 'string',
        'AllowedValues': 'string',
        'IsModifiable': True|False,
        'MinimumEngineVersion': 'string',
        'CacheNodeTypeSpecificValues': [
          {
            'CacheNodeType': 'string',
            'Value': 'string'
          },
        ]
      },
    ]
  },
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a *DescribeEngineDefaultParameters* action.

– **EngineDefaults** (*dict*) –

Represents the output of a *DescribeEngineDefaultParameters* action.

- * **CacheParameterGroupFamily** (*string*) –

Specifies the name of the cache parameter group family to which the engine default parameters apply.

- * **Marker** (*string*) –

Provides an identifier to allow retrieval of paginated results.

* **Parameters** (*list*) –

Contains a list of engine default parameters.

· (*dict*) –

Describes an individual setting that controls some aspect of ElastiCache behavior.

· **ParameterName** (*string*) –

The name of the parameter.

· **ParameterValue** (*string*) –

The value of the parameter.

· **Description** (*string*) –

A description of the parameter.

· **Source** (*string*) –

The source of the parameter.

· **DataType** (*string*) –

The valid data type for the parameter.

· **AllowedValues** (*string*) –

The valid range of values for the parameter.

· **IsModifiable** (*boolean*) –

Indicates whether (`true`) or not (`false`) the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.

· **MinimumEngineVersion** (*string*) –

The earliest cache engine version to which the parameter can apply.

* **CacheNodeTypeSpecificParameters** (*list*) –

A list of parameters specific to a particular cache node type. Each element in the list contains detailed information about one parameter.

· (*dict*) –

A parameter that has a different value for each cache node type it is applied to. For example, in a Redis cache cluster, a `cache.m1.large` cache node type would have a larger `maxmemory` value than a `cache.m1.small` type.

· **ParameterName** (*string*) –

The name of the parameter.

· **Description** (*string*) –

A description of the parameter.

· **Source** (*string*) –

The source of the parameter value.

· **DataType** (*string*) –

The valid data type for the parameter.

· **AllowedValues** (*string*) –

The valid range of values for the parameter.

· **IsModifiable** (*boolean*) –

Indicates whether (`true`) or not (`false`) the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.

- **MinimumEngineVersion** (*string*) –

The earliest cache engine version to which the parameter can apply.

- **CacheNodeTypeSpecificValues** (*list*) –

A list of cache node types and their corresponding values for this parameter.

- (*dict*) –

A value that applies only to a certain cache node type.

- **CacheNodeType** (*string*) –

The cache node type for which this value applies.

- **Value** (*string*) –

The value for the cache node type.

- **NextToken** (*string*) –

A token to resume pagination.

class `ElastiCache.Paginator.DescribeEvents`

```
paginator = client.get_paginator('describe_events')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `ElastiCache.Client.describe_events()`.

Request Syntax

```
response_iterator = paginator.paginate(
    SourceIdentifier='string',
    SourceType='cache-cluster'|'cache-parameter-group'|'cache-security-group'|'cache-subnet',
    StartTime=datetime(2015, 1, 1),
    EndTime=datetime(2015, 1, 1),
    Duration=123,
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **SourceIdentifier** (*string*) – The identifier of the event source for which events will be returned. If not specified, then all sources are included in the response.
- **SourceType** (*string*) – The event source to retrieve events for. If no value is specified, all events are returned.

Valid values are: `cache-cluster` | `cache-parameter-group` | `cache-security-group` | `cache-subnet-group`

- **StartTime** (*datetime*) – The beginning of the time interval to retrieve events for, specified in ISO 8601 format.
- **EndTime** (*datetime*) – The end of the time interval for which to retrieve events, specified in ISO 8601 format.

- **Duration** (*integer*) – The number of minutes’ worth of events to retrieve.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Events': [
    {
      'SourceIdentifier': 'string',
      'SourceType': 'cache-cluster'|'cache-parameter-group'|'cache-security-
      'Message': 'string',
      'Date': datetime(2015, 1, 1)
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a *DescribeEvents* action.

- **Events** (*list*) –

A list of events. Each element in the list contains detailed information about one event.

- * (*dict*) –

Represents a single occurrence of something interesting within the system. Some examples of events are creating a cache cluster, adding or removing a cache node, or rebooting a node.

- **SourceIdentifier** (*string*) –

The identifier for the source of the event. For example, if the event occurred at the cache cluster level, the identifier would be the name of the cache cluster.

- **SourceType** (*string*) –

Specifies the origin of this event - a cache cluster, a parameter group, a security group, etc.

- **Message** (*string*) –

The text of the event.

- **Date** (*datetime*) –

The date and time when the event occurred.

– **NextToken** (*string*) –

A token to resume pagination.

class `ElastiCache.Paginator.DescribeReplicationGroups`

```
paginator = client.get_paginator('describe_replication_groups')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `ElastiCache.Client.describe_replication_groups()`.

Request Syntax

```
response_iterator = paginator.paginate(
    ReplicationGroupId='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **ReplicationGroupId** (*string*) – The identifier for the replication group to be described. This parameter is not case sensitive.

If you do not specify this parameter, information about all replication groups is returned.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

– **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

– **PageSize** (*integer*) –

The size of each page.

– **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
  'ReplicationGroups': [
    {
      'ReplicationGroupId': 'string',
      'Description': 'string',
      'Status': 'string',
      'PendingModifiedValues': {
        'PrimaryClusterId': 'string',
        'AutomaticFailoverStatus': 'enabled'|'disabled'
      },
    },
  ],
}
```

```
    'MemberClusters': [
        'string',
    ],
    'NodeGroups': [
        {
            'NodeGroupId': 'string',
            'Status': 'string',
            'PrimaryEndpoint': {
                'Address': 'string',
                'Port': 123
            },
            'NodeGroupMembers': [
                {
                    'CacheClusterId': 'string',
                    'CacheNodeId': 'string',
                    'ReadEndpoint': {
                        'Address': 'string',
                        'Port': 123
                    },
                    'PreferredAvailabilityZone': 'string',
                    'CurrentRole': 'string'
                },
            ],
        },
    ],
    'SnapshottingClusterId': 'string',
    'AutomaticFailover': 'enabled'|'disabled'|'enabling'|'disabling'
},
'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a *DescribeReplicationGroups* action.

- **ReplicationGroups** (*list*) –

A list of replication groups. Each item in the list contains detailed information about one replication group.

- * (*dict*) –

Contains all of the attributes of a specific replication group.

- **ReplicationGroupId** (*string*) –

The identifier for the replication group.

- **Description** (*string*) –

The description of the replication group.

- **Status** (*string*) –

The current state of this replication group - *creating* , *available* , etc.

- **PendingModifiedValues** (*dict*) –

A group of settings to be applied to the replication group, either immediately or during the next maintenance window.

- **PrimaryClusterId** (*string*) –

The primary cluster ID which will be applied immediately

(if `--apply-immediately` was specified), or during the next maintenance window.

- **AutomaticFailoverStatus** (*string*) –

Indicates the status of Multi-AZ for this replication group.

Note: ElastiCache Multi-AZ replication groups are not supported on:

- Redis versions earlier than 2.8.6.
 - T1 and T2 cache node types.
-

- **MemberClusters** (*list*) –

The names of all the cache clusters that are part of this replication group.

- (*string*) –

- **NodeGroups** (*list*) –

A single element list with information about the nodes in the replication group.

- (*dict*) –

Represents a collection of cache nodes in a replication group.

- **NodeGroupId** (*string*) –

The identifier for the node group. A replication group contains only one node group; therefore, the node group ID is 0001.

- **Status** (*string*) –

The current state of this replication group - *creating*, *available*, etc.

- **PrimaryEndpoint** (*dict*) –

Represents the information required for client programs to connect to a cache node.

- **Address** (*string*) –

The DNS hostname of the cache node.

- **Port** (*integer*) –

The port number that the cache engine is listening on.

- **NodeGroupMembers** (*list*) –

A list containing information about individual nodes within the node group.

- (*dict*) –

Represents a single node within a node group.

- **CacheClusterId** (*string*) –

The ID of the cache cluster to which the node belongs.

- **CacheNodeId** (*string*) –

The ID of the node within its cache cluster. A node ID is a numeric identifier (0001, 0002, etc.).

- **ReadEndpoint** (*dict*) –

Represents the information required for client programs to connect to a cache node.

- **Address** (*string*) –

The DNS hostname of the cache node.

- **Port** (*integer*) –

The port number that the cache engine is listening on.

- **PreferredAvailabilityZone** (*string*) –

The name of the Availability Zone in which the node is located.

- **CurrentRole** (*string*) –

The role that is currently assigned to the node - *primary* or *replica*.

- **SnapshottingClusterId** (*string*) –

The cache cluster ID that is used as the daily snapshot source for the replication group.

- **AutomaticFailover** (*string*) –

Indicates the status of Multi-AZ for this replication group.

Note: ElastiCache Multi-AZ replication groups are not supported on:

- Redis versions earlier than 2.8.6.
 - T1 and T2 cache node types.
-

- **NextToken** (*string*) –

A token to resume pagination.

class `ElastiCache.Paginator.DescribeReservedCacheNodes`

```
paginator = client.get_paginator('describe_reserved_cache_nodes')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `ElastiCache.Client.describe_reserved_cache_nodes()`.

Request Syntax

```
response_iterator = paginator.paginate(
    ReservedCacheNodeId='string',
    ReservedCacheNodesOfferingId='string',
    CacheNodeType='string',
    Duration='string',
    ProductDescription='string',
    OfferingType='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **ReservedCacheNodeId** (*string*) – The reserved cache node identifier filter value. Use this parameter to show only the reservation that matches the specified reservation ID.

- **ReservedCacheNodesOfferingId** (*string*) – The offering identifier filter value. Use this parameter to show only purchased reservations matching the specified offering identifier.
- **CacheNodeType** (*string*) – The cache node type filter value. Use this parameter to show only those reservations matching the specified cache node type.

Valid node types are as follows:

- General purpose: * Current generation: `cache.t2.micro` , `cache.t2.small` , `cache.t2.medium` , `cache.m3.medium` , `cache.m3.large` , `cache.m3.xlarge` , `cache.m3.2xlarge`
- Previous generation: `cache.t1.micro` , `cache.m1.small` , `cache.m1.medium` , `cache.m1.large` , `cache.m1.xlarge`
- Compute optimized: `cache.c1.xlarge`
- Memory optimized * Current generation: `cache.r3.large` , `cache.r3.xlarge` , `cache.r3.2xlarge` , `cache.r3.4xlarge` , `cache.r3.8xlarge`
- Previous generation: `cache.m2.xlarge` , `cache.m2.2xlarge` , `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#) .

- **Duration** (*string*) – The duration filter value, specified in years or seconds. Use this parameter to show only reservations for this duration.

Valid Values: 1 | 3 | 31536000 | 94608000

- **ProductDescription** (*string*) – The product description filter value. Use this parameter to show only those reservations matching the specified product description.
- **OfferingType** (*string*) – The offering type filter value. Use this parameter to show only the available offerings matching the specified offering type.

Valid values: "Light Utilization"|"Medium Utilization"|"Heavy Utilization"

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'ReservedCacheNodes': [
    {
      'ReservedCacheNodeId': 'string',
      'ReservedCacheNodesOfferingId': 'string',
      'CacheNodeType': 'string',
      'StartTime': datetime(2015, 1, 1),
      'Duration': 123,
      'FixedPrice': 123.0,
      'UsagePrice': 123.0,
      'CacheNodeCount': 123,
      'ProductDescription': 'string',
      'OfferingType': 'string',
      'State': 'string',
      'RecurringCharges': [
        {
          'RecurringChargeAmount': 123.0,
          'RecurringChargeFrequency': 'string'
        },
      ],
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a *DescribeReservedCacheNodes* action.

- **ReservedCacheNodes** (*list*) –

A list of reserved cache nodes. Each element in the list contains detailed information about one node.

- * (*dict*) –

Represents the output of a *PurchaseReservedCacheNodesOffering* action.

- **ReservedCacheNodeId** (*string*) –

The unique identifier for the reservation.

- **ReservedCacheNodesOfferingId** (*string*) –

The offering identifier.

- **CacheNodeType** (*string*) –

The cache node type for the reserved cache nodes.

Valid node types are as follows:

- General purpose: * Current generation: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
- Previous generation: `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
- Compute optimized: `cache.c1.xlarge`
- Memory optimized * Current generation: `cache.r3.large`, `cache.r3.xlarge`,

```
cache.r3.2xlarge , cache.r3.4xlarge ,
cache.r3.8xlarge
· Previous generation: cache.m2.xlarge ,
cache.m2.2xlarge , cache.m2.4xlarge
```

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#).

- **StartTime** (*datetime*) –

The time the reservation started.

- **Duration** (*integer*) –

The duration of the reservation in seconds.

- **FixedPrice** (*float*) –

The fixed price charged for this reserved cache node.

- **UsagePrice** (*float*) –

The hourly price charged for this reserved cache node.

- **CacheNodeCount** (*integer*) –

The number of cache nodes that have been reserved.

- **ProductDescription** (*string*) –

The description of the reserved cache node.

- **OfferingType** (*string*) –

The offering type of this reserved cache node.

- **State** (*string*) –

The state of the reserved cache node.

- **RecurringCharges** (*list*) –

The recurring price charged to run this reserved cache node.

- (*dict*) –

Contains the specific price and frequency of a recurring charges for a reserved cache node, or for a reserved cache node offering.

- **RecurringChargeAmount** (*float*) –

The monetary amount of the recurring charge.

- **RecurringChargeFrequency** (*string*) –

The frequency of the recurring charge.

- **NextToken** (*string*) –

A token to resume pagination.

```
class ElastiCache.Paginator.DescribeReservedCacheNodesOfferings
```

```
paginator = client.get_paginator('describe_reserved_cache_nodes_offerings')
```

paginate (**kwargs)

Creates an iterator that will paginate through responses from `ElastiCache.Client.describe_reserved_cache_nodes_offerings()`.

Request Syntax

```
response_iterator = paginator.paginate(
    ReservedCacheNodesOfferingId='string',
    CacheNodeType='string',
    Duration='string',
    ProductDescription='string',
    OfferingType='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **ReservedCacheNodesOfferingId** (*string*) – The offering identifier filter value. Use this parameter to show only the available offering that matches the specified reservation identifier.

Example: 438012d3-4052-4cc7-b2e3-8d3372e0e706

- **CacheNodeType** (*string*) – The cache node type filter value. Use this parameter to show only the available offerings matching the specified cache node type.

Valid node types are as follows:

- General purpose: * Current generation: `cache.t2.micro` , `cache.t2.small` , `cache.t2.medium` , `cache.m3.medium` , `cache.m3.large` , `cache.m3.xlarge` , `cache.m3.2xlarge`
- Previous generation: `cache.t1.micro` , `cache.m1.small` , `cache.m1.medium` , `cache.m1.large` , `cache.m1.xlarge`
- Compute optimized: `cache.c1.xlarge`
- Memory optimized * Current generation: `cache.r3.large` , `cache.r3.xlarge` , `cache.r3.2xlarge` , `cache.r3.4xlarge` , `cache.r3.8xlarge`
- Previous generation: `cache.m2.xlarge` , `cache.m2.2xlarge` , `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#) .

- **Duration** (*string*) – Duration filter value, specified in years or seconds. Use this parameter to show only reservations for a given duration.

Valid Values: 1 | 3 | 31536000 | 94608000

- **ProductDescription** (*string*) – The product description filter value. Use this parameter to show only the available offerings matching the specified product description.
- **OfferingType** (*string*) – The offering type filter value. Use this parameter to show only the available offerings matching the specified offering type.

Valid Values: "Light Utilization"|"Medium Utilization"|"Heavy Utilization"

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'ReservedCacheNodesOfferings': [
    {
      'ReservedCacheNodesOfferingId': 'string',
      'CacheNodeType': 'string',
      'Duration': 123,
      'FixedPrice': 123.0,
      'UsagePrice': 123.0,
      'ProductDescription': 'string',
      'OfferingType': 'string',
      'RecurringCharges': [
        {
          'RecurringChargeAmount': 123.0,
          'RecurringChargeFrequency': 'string'
        }
      ],
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a `DescribeReservedCacheNodesOfferings` action.

- **ReservedCacheNodesOfferings** (*list*) –

A list of reserved cache node offerings. Each element in the list contains detailed information about one offering.

- * (*dict*) –

Describes all of the attributes of a reserved cache node offering.

- **ReservedCacheNodesOfferingId** (*string*) –

A unique identifier for the reserved cache node offering.

- **CacheNodeType** (*string*) –

The cache node type for the reserved cache node.

Valid node types are as follows:

- General purpose: * Current generation: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
- Previous generation: `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
- Compute optimized: `cache.c1.xlarge`
- Memory optimized * Current generation: `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`, `cache.r3.4xlarge`, `cache.r3.8xlarge`
- Previous generation: `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#).

- **Duration** (*integer*) –

The duration of the offering, in seconds.

- **FixedPrice** (*float*) –

The fixed price charged for this offering.

- **UsagePrice** (*float*) –

The hourly price charged for this offering.

- **ProductDescription** (*string*) –

The cache engine used by the offering.

- **OfferingType** (*string*) –

The offering type.

- **RecurringCharges** (*list*) –

The recurring price charged to run this reserved cache node.

- (*dict*) –

Contains the specific price and frequency of a recurring charges for a reserved cache node, or for a reserved cache node offering.

- **RecurringChargeAmount** (*float*) –

The monetary amount of the recurring charge.

- **RecurringChargeFrequency** (*string*) –

The frequency of the recurring charge.

- **NextToken** (*string*) –

A token to resume pagination.

`class ElastiCache.Paginator.DescribeSnapshots`


```
paginator = client.get_paginator('describe_snapshots')
```

paginate (**kwargs)

Creates an iterator that will paginate through responses from `ElastiCache.Client.describe_snapshots()`.

Request Syntax

```
response_iterator = paginator.paginate(
    CacheClusterId='string',
    SnapshotName='string',
    SnapshotSource='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **CacheClusterId** (*string*) – A user-supplied cluster identifier. If this parameter is specified, only snapshots associated with that specific cache cluster will be described.
- **SnapshotName** (*string*) – A user-supplied name of the snapshot. If this parameter is specified, only this snapshot will be described.
- **SnapshotSource** (*string*) – If set to `system`, the output shows snapshots that were automatically created by ElastiCache. If set to `user` the output shows snapshots that were manually created. If omitted, the output shows both automatically and manually created snapshots.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Snapshots': [
    {
      'SnapshotName': 'string',
      'CacheClusterId': 'string',
      'SnapshotStatus': 'string',
      'SnapshotSource': 'string',
      'CacheNodeType': 'string',
      'Engine': 'string',
```

```
'EngineVersion': 'string',
'NumCacheNodes': 123,
'PreferredAvailabilityZone': 'string',
'CacheClusterCreateTime': datetime(2015, 1, 1),
'PreferredMaintenanceWindow': 'string',
'TopicArn': 'string',
'Port': 123,
'CacheParameterGroupName': 'string',
'CacheSubnetGroupName': 'string',
'VpcId': 'string',
'AutoMinorVersionUpgrade': True|False,
'SnapshotRetentionLimit': 123,
'SnapshotWindow': 'string',
'NodeSnapshots': [
    {
        'CacheNodeId': 'string',
        'CacheSize': 'string',
        'CacheNodeCreateTime': datetime(2015, 1, 1),
        'SnapshotCreateTime': datetime(2015, 1, 1)
    },
],
},
],
'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a *DescribeSnapshots* action.

- **Snapshots** (*list*) –

A list of snapshots. Each item in the list contains detailed information about one snapshot.

- * (*dict*) –

Represents a copy of an entire cache cluster as of the time when the snapshot was taken.

- **SnapshotName** (*string*) –

The name of a snapshot. For an automatic snapshot, the name is system-generated; for a manual snapshot, this is the user-provided name.

- **CacheClusterId** (*string*) –

The user-supplied identifier of the source cache cluster.

- **SnapshotStatus** (*string*) –

The status of the snapshot. Valid values: `creating` | `available` | `restoring` | `copying` | `deleting`.

- **SnapshotSource** (*string*) –

Indicates whether the snapshot is from an automatic backup (`automated`) or was created manually (`manual`).

- **CacheNodeType** (*string*) –

The name of the compute and memory capacity node type for the source cache cluster.

Valid node types are as follows:

- General purpose: * Current generation: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`, `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`, `cache.m3.2xlarge`
- Previous generation: `cache.t1.micro`, `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`, `cache.m1.xlarge`
- Compute optimized: `cache.c1.xlarge`
- Memory optimized * Current generation: `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`, `cache.r3.4xlarge`, `cache.r3.8xlarge`
- Previous generation: `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`

Notes:

- All t2 instances are created in an Amazon Virtual Private Cloud (VPC).
- Redis backup/restore is not supported for t2 instances.
- Redis Append-only files (AOF) functionality is not supported for t1 or t2 instances.

For a complete listing of cache node types and specifications, see [Amazon ElastiCache Product Features and Details](#) and [Cache Node Type-Specific Parameters for Memcached](#) or [Cache Node Type-Specific Parameters for Redis](#).

- **Engine** (*string*) –

The name of the cache engine (*memcached* or *redis*) used by the source cache cluster.

- **EngineVersion** (*string*) –

The version of the cache engine version that is used by the source cache cluster.

- **NumCacheNodes** (*integer*) –

The number of cache nodes in the source cache cluster.

For clusters running Redis, this value must be 1. For clusters running Memcached, this value must be between 1 and 20.

- **PreferredAvailabilityZone** (*string*) –

The name of the Availability Zone in which the source cache cluster is located.

- **CacheClusterCreateTime** (*datetime*) –

The date and time when the source cache cluster was created.

- **PreferredMaintenanceWindow** (*string*) –

Specifies the weekly time range during which maintenance on the cache cluster is performed. It is specified as a range in the format `ddd:hh24:mi-ddd:hh24:mi` (24H Clock UTC). The minimum maintenance window is a 60 minute period. Valid values for `ddd` are:

- `sun`
- `mon`
- `tue`
- `wed`
- `thu`

- `fri`
- `sat`
- Example: `sun:05:00-sun:09:00`
- **TopicArn** (*string*) –
The Amazon Resource Name (ARN) for the topic used by the source cache cluster for publishing notifications.
- **Port** (*integer*) –
The port number used by each cache nodes in the source cache cluster.
- **CacheParameterGroupName** (*string*) –
The cache parameter group that is associated with the source cache cluster.
- **CacheSubnetGroupName** (*string*) –
The name of the cache subnet group associated with the source cache cluster.
- **VpcId** (*string*) –
The Amazon Virtual Private Cloud identifier (VPC ID) of the cache subnet group for the source cache cluster.
- **AutoMinorVersionUpgrade** (*boolean*) –
This parameter is currently disabled.
- **SnapshotRetentionLimit** (*integer*) –
For an automatic snapshot, the number of days for which ElastiCache will retain the snapshot before deleting it.

For manual snapshots, this field reflects the *SnapshotRetentionLimit* for the source cache cluster when the snapshot was created. This field is otherwise ignored: Manual snapshots do not expire, and can only be deleted using the *DeleteSnapshot* action.
- Important** If the value of *SnapshotRetentionLimit* is set to zero (0), backups are turned off.
- **SnapshotWindow** (*string*) –
The daily time range during which ElastiCache takes daily snapshots of the source cache cluster.
- **NodeSnapshots** (*list*) –
A list of the cache nodes in the source cache cluster.
- (*dict*) –
Represents an individual cache node in a snapshot of a cache cluster.
- **CacheNodeId** (*string*) –
The cache node identifier for the node in the source cache cluster.
- **CacheSize** (*string*) –
The size of the cache on the source cache node.
- **CacheNodeCreateTime** (*datetime*) –
The date and time when the cache node was created in the source cache cluster.

- **SnapshotCreateTime** (*datetime*) –
The date and time when the source node’s metadata and cache data set was obtained for the snapshot.
- **NextToken** (*string*) –
A token to resume pagination.

Waiters

The available waiters are:

- `ElastiCache.Waiter.CacheClusterAvailable`
- `ElastiCache.Waiter.CacheClusterDeleted`
- `ElastiCache.Waiter.ReplicationGroupAvailable`
- `ElastiCache.Waiter.ReplicationGroupDeleted`

class `ElastiCache.Waiter.CacheClusterAvailable`

```
waiter = client.get_waiter('cache_cluster_available')
```

wait (***kwargs*)

Polls `ElastiCache.Client.describe_cache_clusters()` every 30 seconds until a successful state is reached. An error is returned after 60 failed checks.

Request Syntax

```
waiter.wait(
    CacheClusterId='string',
    MaxRecords=123,
    Marker='string',
    ShowCacheNodeInfo=True|False
)
```

Parameters

- **CacheClusterId** (*string*) – The user-supplied cluster identifier. If this parameter is specified, only information about that specific cache cluster is returned. This parameter isn’t case sensitive.
- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

- **Marker** (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this action. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.
- **ShowCacheNodeInfo** (*boolean*) – An optional flag that can be included in the `DescribeCacheCluster` request to retrieve information about the individual cache nodes.

Returns None

class `ElastiCache.Waiter.CacheClusterDeleted`

```
waiter = client.get_waiter('cache_cluster_deleted')
```

wait (***kwargs*)

Polls `ElastiCache.Client.describe_cache_clusters()` every 30 seconds until a successful state is reached. An error is returned after 60 failed checks.

Request Syntax

```
waiter.wait(
    CacheClusterId='string',
    MaxRecords=123,
    Marker='string',
    ShowCacheNodeInfo=True|False
)
```

Parameters

- **CacheClusterId** (*string*) – The user-supplied cluster identifier. If this parameter is specified, only information about that specific cache cluster is returned. This parameter isn't case sensitive.
- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

- **Marker** (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this action. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.
- **ShowCacheNodeInfo** (*boolean*) – An optional flag that can be included in the `DescribeCacheCluster` request to retrieve information about the individual cache nodes.

Returns None

class `ElastiCache.Waiter.ReplicationGroupAvailable`

```
waiter = client.get_waiter('replication_group_available')
```

wait (***kwargs*)

Polls `ElastiCache.Client.describe_replication_groups()` every 30 seconds until a successful state is reached. An error is returned after 60 failed checks.

Request Syntax

```
waiter.wait(
    ReplicationGroupId='string',
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **ReplicationGroupId** (*string*) – The identifier for the replication group to be described. This parameter is not case sensitive.

If you do not specify this parameter, information about all replication groups is returned.

- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

- **Marker** (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this action. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Returns None

`class ElasticCache.Waiter.ReplicationGroupDeleted`

```
waiter = client.get_waiter('replication_group_deleted')
```

wait (***kwargs*)

Polls `ElasticCache.Client.describe_replication_groups()` every 30 seconds until a successful state is reached. An error is returned after 60 failed checks.

Request Syntax

```
waiter.wait(
    ReplicationGroupId='string',
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **ReplicationGroupId** (*string*) – The identifier for the replication group to be described. This parameter is not case sensitive.

If you do not specify this parameter, information about all replication groups is returned.

- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

- **Marker** (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this action. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Returns None

ElasticBeanstalk

Table of Contents

- ElasticBeanstalk
 - Client
 - Paginators

Client**class ElasticBeanstalk.Client**

A low-level client representing AWS Elastic Beanstalk:

```
import boto3

client = boto3.client('elasticbeanstalk')
```

These are the available methods:

- `abort_environment_update()`
- `can_paginate()`
- `check_dns_availability()`
- `create_application()`
- `create_application_version()`
- `create_configuration_template()`
- `create_environment()`
- `create_storage_location()`
- `delete_application()`
- `delete_application_version()`
- `delete_configuration_template()`
- `delete_environment_configuration()`
- `describe_application_versions()`
- `describe_applications()`
- `describe_configuration_options()`
- `describe_configuration_settings()`
- `describe_environment_health()`
- `describe_environment_resources()`
- `describe_environments()`
- `describe_events()`
- `describe_instances_health()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_waiter()`
- `list_available_solution_stacks()`
- `rebuild_environment()`
- `request_environment_info()`
- `restart_app_server()`
- `retrieve_environment_info()`
- `swap_environment_cnames()`
- `terminate_environment()`
- `update_application()`
- `update_application_version()`
- `update_configuration_template()`
- `update_environment()`
- `validate_configuration_settings()`

abort_environment_update (**kwargs)

Cancels in-progress environment configuration update or application version deployment.

Request Syntax

```
response = client.abort_environment_update(
    EnvironmentId='string',
    EnvironmentName='string'
)
```

Parameters

- **EnvironmentId** (*string*) – This specifies the ID of the environment with the in-progress update that you want to cancel.
- **EnvironmentName** (*string*) – This specifies the name of the environment with the in-progress update that you want to cancel.

Returns None

can_paginate (operation_name)

Check if an operation can be paginated.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

check_dns_availability (**kwargs)

Checks if the specified CNAME is available.

Request Syntax

```
response = client.check_dns_availability(
    CNAMEPrefix='string'
)
```

Parameters **CNAMEPrefix** (*string*) – [REQUIRED]

The prefix used when this CNAME is reserved.

Return type dict

Returns

Response Syntax

```
{
    'Available': True|False,
    'FullyQualifiedCNAME': 'string'
}
```

Response Structure

- (*dict*) –

Indicates if the specified CNAME is available.

- **Available** (*boolean*) –

Indicates if the specified CNAME is available:

 true: The CNAME is available.

 true: The CNAME is not available.

* true: The CNAME is available.

- * `false`: The CNAME is not available.
- **FullyQualifiedCNAME** (*string*) –

The fully qualified CNAME to reserve when `CreateEnvironment` is called with the provided prefix.

create_application (***kwargs*)

Creates an application that has one configuration template named `default` and no application versions.

Request Syntax

```
response = client.create_application(
    ApplicationName='string',
    Description='string'
)
```

Parameters

- **ApplicationName** (*string*) – [REQUIRED]

The name of the application.

Constraint: This name must be unique within your account. If the specified name already exists, the action returns an `InvalidParameterValue` error.

- **Description** (*string*) – Describes the application.

Return type dict

Returns

Response Syntax

```
{
  'Application': {
    'ApplicationName': 'string',
    'Description': 'string',
    'DateCreated': datetime(2015, 1, 1),
    'DateUpdated': datetime(2015, 1, 1),
    'Versions': [
      'string',
    ],
    'ConfigurationTemplates': [
      'string',
    ]
  }
}
```

Response Structure

- (*dict*) –

Result message containing a single description of an application.

- **Application** (*dict*) –

The `ApplicationDescription` of the application.

- * **ApplicationName** (*string*) –

The name of the application.

- * **Description** (*string*) –

User-defined description of the application.

- * **DateCreated** (*datetime*) –

The date when the application was created.

- * **DateUpdated** (*datetime*) –
The date when the application was last modified.
- * **Versions** (*list*) –
The names of the versions for this application.
· (*string*) –
- * **ConfigurationTemplates** (*list*) –
The names of the configuration templates associated with this application.
· (*string*) –

create_application_version (***kwargs*)

Creates an application version for the specified application.

Note: Once you create an application version with a specified Amazon S3 bucket and key location, you cannot change that Amazon S3 location. If you change the Amazon S3 location, you receive an exception when you attempt to launch an environment from the application version.

Request Syntax

```
response = client.create_application_version(
    ApplicationName='string',
    VersionLabel='string',
    Description='string',
    SourceBundle={
        'S3Bucket': 'string',
        'S3Key': 'string'
    },
    AutoCreateApplication=True|False
)
```

Parameters

- **ApplicationName** (*string*) – **[REQUIRED]**

The name of the application. If no application is found with this name, and `AutoCreateApplication` is `false`, returns an `InvalidParameterValue` error.

- **VersionLabel** (*string*) – **[REQUIRED]**

A label identifying this version.

Constraint: Must be unique per application. If an application version already exists with this label for the specified application, AWS Elastic Beanstalk returns an `InvalidParameterValue` error.

- **Description** (*string*) – Describes this version.
- **SourceBundle** (*dict*) – The Amazon S3 bucket and key that identify the location of the source bundle for this version.

If data found at the Amazon S3 location exceeds the maximum allowed source bundle size, AWS Elastic Beanstalk returns an `InvalidParameterValue` error. The maximum size allowed is 512 MB.

Default: If not specified, AWS Elastic Beanstalk uses a sample application. If only partially specified (for example, a bucket is provided but not the key) or if no data is found at the Amazon S3 location, AWS Elastic Beanstalk returns an `InvalidParameterCombination` error.

- **S3Bucket** (*string*) –
The Amazon S3 bucket where the data is located.
- **S3Key** (*string*) –
The Amazon S3 key where the data is located.
- **AutoCreateApplication** (*boolean*) – Determines how the system behaves if the specified application for this version does not already exist:
 - `true` : Automatically creates the specified application for this version if it does not already exist.
 - `false` : Returns an `InvalidParameterValue` if the specified application for this version does not already exist.
- `true` : Automatically creates the specified application for this release if it does not already exist.
- `false` : Throws an `InvalidParameterValue` if the specified application for this release does not already exist.

Default: `false`

Valid Values: `true` | `false`

Return type `dict`

Returns

Response Syntax

```
{
  'ApplicationVersion': {
    'ApplicationName': 'string',
    'Description': 'string',
    'VersionLabel': 'string',
    'SourceBundle': {
      'S3Bucket': 'string',
      'S3Key': 'string'
    },
    'DateCreated': datetime(2015, 1, 1),
    'DateUpdated': datetime(2015, 1, 1)
  }
}
```

Response Structure

- (*dict*) –

Result message wrapping a single description of an application version.

- **ApplicationVersion** (*dict*) –

The `ApplicationVersionDescription` of the application version.

- * **ApplicationName** (*string*) –

The name of the application associated with this release.

- * **Description** (*string*) –

The description of this application version.

- * **VersionLabel** (*string*) –

A label uniquely identifying the version for the associated application.

- * **SourceBundle** (*dict*) –

The location where the source bundle is located for this version.

- **S3Bucket** (*string*) –
The Amazon S3 bucket where the data is located.
- **S3Key** (*string*) –
The Amazon S3 key where the data is located.
- * **DateCreated** (*datetime*) –
The creation date of the application version.
- * **DateUpdated** (*datetime*) –
The last modified date of the application version.

create_configuration_template (***kwargs*)

Creates a configuration template. Templates are associated with a specific application and are used to deploy different versions of the application with the same configuration settings.

Related Topics

- DescribeConfigurationOptions
- DescribeConfigurationSettings
- ListAvailableSolutionStacks

Request Syntax

```
response = client.create_configuration_template(
    ApplicationName='string',
    TemplateName='string',
    SolutionStackName='string',
    SourceConfiguration={
        'ApplicationName': 'string',
        'TemplateName': 'string'
    },
    EnvironmentId='string',
    Description='string',
    OptionSettings=[
        {
            'ResourceName': 'string',
            'Namespace': 'string',
            'OptionName': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **ApplicationName** (*string*) – **[REQUIRED]**
The name of the application to associate with this configuration template. If no application is found with this name, AWS Elastic Beanstalk returns an `InvalidParameterValue` error.
- **TemplateName** (*string*) – **[REQUIRED]**
The name of the configuration template.

Constraint: This name must be unique per application.

Default: If a configuration template already exists with this name, AWS Elastic Beanstalk returns an `InvalidParameterValue` error.
- **SolutionStackName** (*string*) – The name of the solution stack used by this configuration. The solution stack specifies the operating system, architecture, and application server for a configuration template. It determines the set of configuration options as well as the possible and default values.

Use `ListAvailableSolutionStacks` to obtain a list of available solution stacks.

A solution stack name or a source configuration parameter must be specified, otherwise AWS Elastic Beanstalk returns an `InvalidParameterValue` error.

If a solution stack name is not specified and the source configuration parameter is specified, AWS Elastic Beanstalk uses the same solution stack as the source configuration template.

- **SourceConfiguration** (*dict*) – If specified, AWS Elastic Beanstalk uses the configuration values from the specified configuration template to create a new configuration.

Values specified in the `OptionSettings` parameter of this call overrides any values obtained from the `SourceConfiguration`.

If no configuration template is found, returns an `InvalidParameterValue` error.

Constraint: If both the solution stack name parameter and the source configuration parameters are specified, the solution stack of the source configuration template must match the specified solution stack name or else AWS Elastic Beanstalk returns an `InvalidParameterCombination` error.

- **ApplicationName** (*string*) –

The name of the application associated with the configuration.

- **TemplateName** (*string*) –

The name of the configuration template.

- **EnvironmentId** (*string*) – The ID of the environment used with this configuration template.
- **Description** (*string*) – Describes this configuration.
- **OptionSettings** (*list*) – If specified, AWS Elastic Beanstalk sets the specified configuration option to the requested value. The new value overrides the value obtained from the solution stack or the source configuration template.
 - (*dict*) –

A specification identifying an individual configuration option along with its current value. For a list of possible option values, go to [Option Values](#) in the *AWS Elastic Beanstalk Developer Guide*.

- * **ResourceName** (*string*) –

A unique resource name for a time-based scaling configuration option.

- * **Namespace** (*string*) –

A unique namespace identifying the option's associated AWS resource.

- * **OptionName** (*string*) –

The name of the configuration option.

- * **Value** (*string*) –

The current value for the configuration option.

Return type dict

Returns

Response Syntax

```

{
    'SolutionStackName': 'string',
    'ApplicationName': 'string',
    'TemplateName': 'string',
    'Description': 'string',
    'EnvironmentName': 'string',
    'DeploymentStatus': 'deployed'|'pending'|'failed',
    'DateCreated': datetime(2015, 1, 1),
    'DateUpdated': datetime(2015, 1, 1),
    'OptionSettings': [
        {
            'ResourceName': 'string',
            'Namespace': 'string',
            'OptionName': 'string',
            'Value': 'string'
        },
    ]
}

```

Response Structure

- (*dict*) –

Describes the settings for a configuration set.

- **SolutionStackName** (*string*) –

The name of the solution stack this configuration set uses.

- **ApplicationName** (*string*) –

The name of the application associated with this configuration set.

- **TemplateName** (*string*) –

If not `null`, the name of the configuration template for this configuration set.

- **Description** (*string*) –

Describes this configuration set.

- **EnvironmentName** (*string*) –

If not `null`, the name of the environment for this configuration set.

- **DeploymentStatus** (*string*) –

If this configuration set is associated with an environment, the `DeploymentStatus` parameter indicates the deployment status of this configuration set:

`null` : This configuration is not associated with a running environment.

`pending` : This is a draft configuration that is not deployed to the associated environment but is in the process of deploying.

`deployed` : This is the configuration that is currently deployed to the associated running environment.

`failed` : This is a draft configuration, that failed to successfully deploy.

* `null` : This configuration is not associated with a running environment.

- * `pending`: This is a draft configuration that is not deployed to the associated environment but is in the process of deploying.
- * `deployed`: This is the configuration that is currently deployed to the associated running environment.
- * `failed`: This is a draft configuration that failed to successfully deploy.

– **DateCreated** (*datetime*) –

The date (in UTC time) when this configuration set was created.

– **DateUpdated** (*datetime*) –

The date (in UTC time) when this configuration set was last modified.

– **OptionSettings** (*list*) –

A list of the configuration options and their values in this configuration set.

- * (*dict*) –

A specification identifying an individual configuration option along with its current value. For a list of possible option values, go to [Option Values](#) in the *AWS Elastic Beanstalk Developer Guide*.

- **ResourceName** (*string*) –

A unique resource name for a time-based scaling configuration option.

- **Namespace** (*string*) –

A unique namespace identifying the option's associated AWS resource.

- **OptionName** (*string*) –

The name of the configuration option.

- **Value** (*string*) –

The current value for the configuration option.

create_environment (***kwargs*)

Launches an environment for the specified application using the specified configuration.

Request Syntax

```
response = client.create_environment(
    ApplicationName='string',
    EnvironmentName='string',
    Description='string',
    CNAMEPrefix='string',
    Tier={
        'Name': 'string',
        'Type': 'string',
        'Version': 'string'
    },
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
    VersionLabel='string',
    TemplateName='string',
    SolutionStackName='string',
```



```

OptionSettings=[
    {
        'ResourceName': 'string',
        'Namespace': 'string',
        'OptionName': 'string',
        'Value': 'string'
    },
],
OptionsToRemove=[
    {
        'ResourceName': 'string',
        'Namespace': 'string',
        'OptionName': 'string'
    },
]
)

```

Parameters

- **ApplicationName** (*string*) – [REQUIRED]

The name of the application that contains the version to be deployed.

If no application is found with this name, `CreateEnvironment` returns an `InvalidParameterValue` error.

- **EnvironmentName** (*string*) – [REQUIRED]

A unique name for the deployment environment. Used in the application URL.

Constraint: Must be from 4 to 23 characters in length. The name can contain only letters, numbers, and hyphens. It cannot start or end with a hyphen. This name must be unique in your account. If the specified name already exists, AWS Elastic Beanstalk returns an `InvalidParameterValue` error.

Default: If the `CNAME` parameter is not specified, the environment name becomes part of the `CNAME`, and therefore part of the visible URL for your application.

- **Description** (*string*) – Describes this environment.

- **CNAMEPrefix** (*string*) – If specified, the environment attempts to use this value as the prefix for the `CNAME`. If not specified, the `CNAME` is generated automatically by appending a random alphanumeric string to the environment name.

- **Tier** (*dict*) – This specifies the tier to use for creating this environment.

- **Name** (*string*) –

The name of this environment tier.

- **Type** (*string*) –

The type of this environment tier.

- **Version** (*string*) –

The version of this environment tier.

- **Tags** (*list*) – This specifies the tags applied to resources in the environment.

- (*dict*) –

Describes a tag applied to a resource in an environment.

- * **Key** (*string*) –

The key of the tag.

- * **Value** (*string*) –

The value of the tag.

- **VersionLabel** (*string*) – The name of the application version to deploy.

If the specified application has no associated application versions, AWS Elastic Beanstalk `UpdateEnvironment` returns an `InvalidParameterValue` error.

Default: If not specified, AWS Elastic Beanstalk attempts to launch the sample application in the container.

- **TemplateName** (*string*) – The name of the configuration template to use in deployment. If no configuration template is found with this name, AWS Elastic Beanstalk returns an `InvalidParameterValue` error.

Condition: You must specify either this parameter or a `SolutionStackName`, but not both. If you specify both, AWS Elastic Beanstalk returns an `InvalidParameterCombination` error. If you do not specify either, AWS Elastic Beanstalk returns a `MissingRequiredParameter` error.

- **SolutionStackName** (*string*) – This is an alternative to specifying a configuration name. If specified, AWS Elastic Beanstalk sets the configuration values to the default values associated with the specified solution stack.

Condition: You must specify either this or a `TemplateName`, but not both. If you specify both, AWS Elastic Beanstalk returns an `InvalidParameterCombination` error. If you do not specify either, AWS Elastic Beanstalk returns a `MissingRequiredParameter` error.

- **OptionSettings** (*list*) – If specified, AWS Elastic Beanstalk sets the specified configuration options to the requested value in the configuration set for the new environment. These override the values obtained from the solution stack or the configuration template.

– (*dict*) –

A specification identifying an individual configuration option along with its current value. For a list of possible option values, go to [Option Values](#) in the *AWS Elastic Beanstalk Developer Guide*.

- * **ResourceName** (*string*) –

A unique resource name for a time-based scaling configuration option.

- * **Namespace** (*string*) –

A unique namespace identifying the option's associated AWS resource.

- * **OptionName** (*string*) –

The name of the configuration option.

- * **Value** (*string*) –

The current value for the configuration option.

- **OptionsToRemove** (*list*) – A list of custom user-defined configuration options to remove from the configuration set for this new environment.

– (*dict*) –

A specification identifying an individual configuration option.

- * **ResourceName** (*string*) –

A unique resource name for a time-based scaling configuration option.

- * **Namespace** (*string*) –

A unique namespace identifying the option's associated AWS resource.

* **OptionName** (*string*) –

The name of the configuration option.

Return type dict

Returns

Response Syntax

```
{
  'EnvironmentName': 'string',
  'EnvironmentId': 'string',
  'ApplicationName': 'string',
  'VersionLabel': 'string',
  'SolutionStackName': 'string',
  'TemplateName': 'string',
  'Description': 'string',
  'EndpointURL': 'string',
  'CNAME': 'string',
  'DateCreated': datetime(2015, 1, 1),
  'DateUpdated': datetime(2015, 1, 1),
  'Status': 'Launching'|'Updating'|'Ready'|'Terminating'|'Terminated',
  'AbortableOperationInProgress': True|False,
  'Health': 'Green'|'Yellow'|'Red'|'Grey',
  'HealthStatus': 'NoData'|'Unknown'|'Pending'|'Ok'|'Info'|'Warning'|'Degraded'|
  'Resources': {
    'LoadBalancer': {
      'LoadBalancerName': 'string',
      'Domain': 'string',
      'Listeners': [
        {
          'Protocol': 'string',
          'Port': 123
        },
      ],
    },
  },
  'Tier': {
    'Name': 'string',
    'Type': 'string',
    'Version': 'string'
  }
}
```

Response Structure

- (*dict*) –

Describes the properties of an environment.

- **EnvironmentName** (*string*) –
The name of this environment.
- **EnvironmentId** (*string*) –
The ID of this environment.
- **ApplicationName** (*string*) –
The name of the application associated with this environment.
- **VersionLabel** (*string*) –
The application version deployed in this environment.
- **SolutionStackName** (*string*) –

The name of the `SolutionStack` deployed with this environment.

– **TemplateName** (*string*) –

The name of the configuration template used to originally launch this environment.

– **Description** (*string*) –

Describes this environment.

– **EndpointURL** (*string*) –

For load-balanced, autoscaling environments, the URL to the LoadBalancer. For single-instance environments, the IP address of the instance.

– **CNAME** (*string*) –

The URL to the CNAME for this environment.

– **DateCreated** (*datetime*) –

The creation date for this environment.

– **DateUpdated** (*datetime*) –

The last modified date for this environment.

– **Status** (*string*) –

The current operational status of the environment:

- * `Launching` : Environment is in the process of initial deployment.
- * `Updating` : Environment is in the process of updating its configuration settings or application version.
- * `Ready` : Environment is available to have an action performed on it, such as update or terminate.
- * `Terminating` : Environment is in the shut-down process.
- * `Terminated` : Environment is not running.

– **AbortableOperationInProgress** (*boolean*) –

Indicates if there is an in-progress environment configuration update or application version deployment that you can cancel.

`true` : There is an update in progress.

`false` : There are no updates currently in progress.

– **Health** (*string*) –

Describes the health status of the environment. AWS Elastic Beanstalk indicates the failure levels for a running environment:

`Red` : Indicates the environment is not working.

`Yellow` : Indicates that something is wrong, the application might not be available, but the instances appear running.

`Green` : Indicates the environment is healthy and fully functional.

- * `Red` : Indicates the environment is not responsive. Occurs when three or more consecutive failures occur for an environment.
- * `Yellow` : Indicates that something is wrong. Occurs when two consecutive failures occur for an environment.
- * `Green` : Indicates the environment is healthy and fully functional.
- * `Grey` : Default health for a new environment. The environment is not fully launched and health checks have not started or health checks are suspended during an `UpdateEnvironment` or `RestartEnvironment` request.

Default: `Grey`

- **HealthStatus** (*string*) –

Returns the health status of the application running in your environment.
For more information, see [Health Colors and Statuses](#) .

- **Resources** (*dict*) –

The description of the AWS resources used by this environment.

- * **LoadBalancer** (*dict*) –

Describes the LoadBalancer.

- **LoadBalancerName** (*string*) –

The name of the LoadBalancer.

- **Domain** (*string*) –

The domain name of the LoadBalancer.

- **Listeners** (*list*) –

A list of Listeners used by the LoadBalancer.

- (*dict*) –

Describes the properties of a Listener for the LoadBalancer.

- **Protocol** (*string*) –

The protocol that is used by the Listener.

- **Port** (*integer*) –

The port that is used by the Listener.

- **Tier** (*dict*) –

Describes the current tier of this environment.

- * **Name** (*string*) –

The name of this environment tier.

- * **Type** (*string*) –

The type of this environment tier.

- * **Version** (*string*) –

The version of this environment tier.

create_storage_location()

Creates the Amazon S3 storage location for the account.

This location is used to store user log files.

Request Syntax

```
response = client.create_storage_location()
```

Return type dict

Returns

Response Syntax

```
{
    'S3Bucket': 'string'
}
```

Response Structure

- (*dict*) –

Results of a CreateStorageLocationResult call.

– **S3Bucket** (*string*) –

The name of the Amazon S3 bucket created.

delete_application (***kwargs*)

Deletes the specified application along with all associated versions and configurations. The application versions will not be deleted from your Amazon S3 bucket.

Note: You cannot delete an application that has a running environment.

Request Syntax

```
response = client.delete_application(
    ApplicationName='string',
    TerminateEnvByForce=True|False
)
```

Parameters

- **ApplicationName** (*string*) – [REQUIRED]

The name of the application to delete.

- **TerminateEnvByForce** (*boolean*) – When set to true, running environments will be terminated before deleting the application.

Returns None

delete_application_version (***kwargs*)

Deletes the specified version from the specified application.

Note: You cannot delete an application version that is associated with a running environment.

Request Syntax

```
response = client.delete_application_version(
    ApplicationName='string',
    VersionLabel='string',
    DeleteSourceBundle=True|False
)
```

Parameters

- **ApplicationName** (*string*) – [REQUIRED]

The name of the application to delete releases from.

- **VersionLabel** (*string*) – [REQUIRED]

The label of the version to delete.

- **DeleteSourceBundle** (*boolean*) – Indicates whether to delete the associated source bundle from Amazon S3:
 - `true` : An attempt is made to delete the associated Amazon S3 source bundle specified at time of creation.
 - `false` : No action is taken on the Amazon S3 source bundle specified at time of creation.

Valid Values: `true` | `false`

Returns None

delete_configuration_template (***kwargs*)

Deletes the specified configuration template.

Note: When you launch an environment using a configuration template, the environment gets a copy

of the template. You can delete or modify the environment's copy of the template without affecting the running environment.

Request Syntax

```
response = client.delete_configuration_template(
    ApplicationName='string',
    TemplateName='string'
)
```

Parameters

- **ApplicationName** (*string*) – [REQUIRED]
The name of the application to delete the configuration template from.
- **TemplateName** (*string*) – [REQUIRED]
The name of the configuration template to delete.

Returns None

delete_environment_configuration (***kwargs*)

Deletes the draft configuration associated with the running environment.

Updating a running environment with any configuration changes creates a draft configuration set. You can get the draft configuration using `DescribeConfigurationSettings` while the update is in progress or if the update fails. The `DeploymentStatus` for the draft configuration indicates whether the deployment is in process or has failed. The draft configuration remains in existence until it is deleted with this action.

Request Syntax

```
response = client.delete_environment_configuration(
    ApplicationName='string',
    EnvironmentName='string'
)
```

Parameters

- **ApplicationName** (*string*) – [REQUIRED]
The name of the application the environment is associated with.
- **EnvironmentName** (*string*) – [REQUIRED]
The name of the environment to delete the draft configuration from.

Returns None

describe_application_versions (***kwargs*)

Returns descriptions for existing application versions.

Request Syntax

```
response = client.describe_application_versions(
    ApplicationName='string',
    VersionLabels=[
        'string',
    ]
)
```

Parameters

- **ApplicationName** (*string*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to only include ones that are associated with the specified application.

- **VersionLabels** (*list*) – If specified, restricts the returned descriptions to only include ones that have the specified version labels.
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'ApplicationVersions': [
    {
      'ApplicationName': 'string',
      'Description': 'string',
      'VersionLabel': 'string',
      'SourceBundle': {
        'S3Bucket': 'string',
        'S3Key': 'string'
      },
      'DateCreated': datetime(2015, 1, 1),
      'DateUpdated': datetime(2015, 1, 1)
    },
  ]
}
```

Response Structure

- (*dict*) –

Result message wrapping a list of application version descriptions.

- **ApplicationVersions** (*list*) –

A list of ApplicationVersionDescription .

- * (*dict*) –

Describes the properties of an application version.

- **ApplicationName** (*string*) –

The name of the application associated with this release.

- **Description** (*string*) –

The description of this application version.

- **VersionLabel** (*string*) –

A label uniquely identifying the version for the associated application.

- **SourceBundle** (*dict*) –

The location where the source bundle is located for this version.

- **S3Bucket** (*string*) –

The Amazon S3 bucket where the data is located.

- **S3Key** (*string*) –

The Amazon S3 key where the data is located.

- **DateCreated** (*datetime*) –

The creation date of the application version.

- **DateUpdated** (*datetime*) –

The last modified date of the application version.

describe_applications (**kwargs)

Returns the descriptions of existing applications.

Request Syntax

```
response = client.describe_applications(
    ApplicationNames=[
        'string',
    ]
)
```

Parameters **ApplicationNames** (*list*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to only include those with the specified names.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'Applications': [
        {
            'ApplicationName': 'string',
            'Description': 'string',
            'DateCreated': datetime(2015, 1, 1),
            'DateUpdated': datetime(2015, 1, 1),
            'Versions': [
                'string',
            ],
            'ConfigurationTemplates': [
                'string',
            ]
        },
    ]
}
```

Response Structure

- (*dict*) –

Result message containing a list of application descriptions.

- **Applications** (*list*) –

This parameter contains a list of ApplicationDescription .

- * (*dict*) –

Describes the properties of an application.

- **ApplicationName** (*string*) –

The name of the application.

- **Description** (*string*) –

User-defined description of the application.

- **DateCreated** (*datetime*) –

The date when the application was created.

- **DateUpdated** (*datetime*) –

The date when the application was last modified.

- **Versions** (*list*) –

The names of the versions for this application.

- (*string*) –
- **ConfigurationTemplates** (*list*) –
The names of the configuration templates associated with this application.
- (*string*) –

describe_configuration_options (***kwargs*)

Describes the configuration options that are used in a particular configuration template or environment, or that a specified solution stack defines. The description includes the values the options, their default values, and an indication of the required action on a running environment if an option value is changed.

Request Syntax

```
response = client.describe_configuration_options(  
    ApplicationName='string',  
    TemplateName='string',  
    EnvironmentName='string',  
    SolutionStackName='string',  
    Options=[  
        {  
            'ResourceName': 'string',  
            'Namespace': 'string',  
            'OptionName': 'string'  
        },  
    ]  
)
```

Parameters

- **ApplicationName** (*string*) – The name of the application associated with the configuration template or environment. Only needed if you want to describe the configuration options associated with either the configuration template or environment.
- **TemplateName** (*string*) – The name of the configuration template whose configuration options you want to describe.
- **EnvironmentName** (*string*) – The name of the environment whose configuration options you want to describe.
- **SolutionStackName** (*string*) – The name of the solution stack whose configuration options you want to describe.
- **Options** (*list*) – If specified, restricts the descriptions to only the specified options.
 - (*dict*) –

A specification identifying an individual configuration option.

- * **ResourceName** (*string*) –

A unique resource name for a time-based scaling configuration option.

- * **Namespace** (*string*) –

A unique namespace identifying the option's associated AWS resource.

- * **OptionName** (*string*) –

The name of the configuration option.

Return type dict

Returns

Response Syntax

```

{
  'SolutionStackName': 'string',
  'Options': [
    {
      'Namespace': 'string',
      'Name': 'string',
      'DefaultValue': 'string',
      'ChangeSeverity': 'string',
      'UserDefined': True|False,
      'ValueType': 'Scalar'|'List',
      'ValueOptions': [
        'string',
      ],
      'MinValue': 123,
      'MaxValue': 123,
      'MaxLength': 123,
      'Regex': {
        'Pattern': 'string',
        'Label': 'string'
      }
    },
  ],
}

```

Response Structure

- *(dict)* –

Describes the settings for a specified configuration set.

- **SolutionStackName** (*string*) –

The name of the solution stack these configuration options belong to.

- **Options** (*list*) –

A list of ConfigurationOptionDescription .

- * *(dict)* –

Describes the possible values for a configuration option.

- **Namespace** (*string*) –

A unique namespace identifying the option's associated AWS resource.

- **Name** (*string*) –

The name of the configuration option.

- **DefaultValue** (*string*) –

The default value for this configuration option.

- **ChangeSeverity** (*string*) –

An indication of which action is required if the value for this configuration option changes:

NoInterruption - There is no interruption to the environment or application availability.

RestartEnvironment - The environment is restarted, all AWS resources are deleted and recreated, and the environment is unavailable during the process.

RestartApplicationServer - The environment is available the entire time. However, a short application outage occurs when

the application servers on the running Amazon EC2 instances are restarted.

- `NoInterruption` : There is no interruption to the environment or application availability.
- `RestartEnvironment` : The environment is entirely restarted, all AWS resources are deleted and recreated, and the environment is unavailable during the process.
- `RestartApplicationServer` : The environment is available the entire time. However, a short application outage occurs when the application servers on the running Amazon EC2 instances are restarted.
- **UserDefined** (*boolean*) –

An indication of whether the user defined this configuration option:

`true` : This configuration option was defined by the user. It is a valid choice for specifying this as an `Option to Remove` when updating configuration settings.

`false` : This configuration was not defined by the user.

- `true` : This configuration option was defined by the user. It is a valid choice for specifying if this as an `Option to Remove` when updating configuration settings.
- `false` : This configuration was not defined by the user.
Constraint: You can remove only `UserDefined` options from a configuration.

Valid Values: `true|false`

- **ValueType** (*string*) –

An indication of which type of values this option has and whether it is allowable to select one or more than one of the possible values:

`Scalar` : Values for this option are a single selection from the possible values, or a unformatted string or numeric value governed by the `MIN/MAX/Regex` constraints:

`List` : Values for this option are multiple selections of the possible values.

`Boolean` : Values for this option are either `true` or `false`.

`Json` : Values for this option are a JSON representation of a `ConfigDocument`.

- `Scalar` : Values for this option are a single selection from the possible values, or an unformatted string, or numeric value governed by the `MIN/MAX/Regex` constraints.
- `List` : Values for this option are multiple selections from the possible values.
- `Boolean` : Values for this option are either `true` or `false`.
- `Json` : Values for this option are a JSON representation of a `ConfigDocument`.

- **ValueOptions** (*list*) –
If specified, values for the configuration option are selected from this list.
- (*string*) –
- **MinValue** (*integer*) –
If specified, the configuration option must be a numeric value greater than this value.
- **MaxValue** (*integer*) –
If specified, the configuration option must be a numeric value less than this value.
- **MaxLength** (*integer*) –
If specified, the configuration option must be a string value no longer than this value.
- **Regex** (*dict*) –
If specified, the configuration option must be a string value that satisfies this regular expression.
- **Pattern** (*string*) –
The regular expression pattern that a string configuration option value with this restriction must match.
- **Label** (*string*) –
A unique name representing this regular expression.

describe_configuration_settings (***kwargs*)

Returns a description of the settings for the specified configuration set, that is, either a configuration template or the configuration set associated with a running environment.

When describing the settings for the configuration set associated with a running environment, it is possible to receive two sets of setting descriptions. One is the deployed configuration set, and the other is a draft configuration of an environment that is either in the process of deployment or that failed to deploy.

Related Topics

- [DeleteEnvironmentConfiguration](#)

Request Syntax

```
response = client.describe_configuration_settings(
    ApplicationName='string',
    TemplateName='string',
    EnvironmentName='string'
)
```

Parameters

- **ApplicationName** (*string*) – **[REQUIRED]**
The application for the environment or configuration template.
- **TemplateName** (*string*) – The name of the configuration template to describe.
Conditional: You must specify either this parameter or an EnvironmentName, but not both. If you specify both, AWS Elastic Beanstalk returns an `InvalidParameterCombination` error. If you do not specify either, AWS Elastic Beanstalk returns a `MissingRequiredParameter` error.
- **EnvironmentName** (*string*) – The name of the environment to describe.
Condition: You must specify either this or a TemplateName, but not both. If you specify both, AWS Elastic Beanstalk returns an

InvalidParameterCombination error. If you do not specify either, AWS Elastic Beanstalk returns MissingRequiredParameter error.

Return type dict

Returns

Response Syntax

```
{
  'ConfigurationSettings': [
    {
      'SolutionStackName': 'string',
      'ApplicationName': 'string',
      'TemplateName': 'string',
      'Description': 'string',
      'EnvironmentName': 'string',
      'DeploymentStatus': 'deployed'|'pending'|'failed',
      'DateCreated': datetime(2015, 1, 1),
      'DateUpdated': datetime(2015, 1, 1),
      'OptionSettings': [
        {
          'ResourceName': 'string',
          'Namespace': 'string',
          'OptionName': 'string',
          'Value': 'string'
        },
      ],
    },
  ],
}
```

Response Structure

- (*dict*) –

The results from a request to change the configuration settings of an environment.

- **ConfigurationSettings** (*list*) –

A list of ConfigurationSettingsDescription .

- * (*dict*) –

Describes the settings for a configuration set.

- **SolutionStackName** (*string*) –

The name of the solution stack this configuration set uses.

- **ApplicationName** (*string*) –

The name of the application associated with this configuration set.

- **TemplateName** (*string*) –

If not null , the name of the configuration template for this configuration set.

- **Description** (*string*) –

Describes this configuration set.

- **EnvironmentName** (*string*) –

If not null , the name of the environment for this configuration set.

- **DeploymentStatus** (*string*) –

If this configuration set is associated with an environment, the `DeploymentStatus` parameter indicates the deployment status of this configuration set:

`null` : This configuration is not associated with a running environment.

`pending` : This is a draft configuration that is not deployed to the associated environment but is in the process of deploying.

`deployed` : This is the configuration that is currently deployed to the associated running environment.

`failed` : This is a draft configuration, that failed to successfully deploy.

- `null` : This configuration is not associated with a running environment.
- `pending` : This is a draft configuration that is not deployed to the associated environment but is in the process of deploying.
- `deployed` : This is the configuration that is currently deployed to the associated running environment.
- `failed` : This is a draft configuration that failed to successfully deploy.

- **DateCreated** (*datetime*) –

The date (in UTC time) when this configuration set was created.

- **DateUpdated** (*datetime*) –

The date (in UTC time) when this configuration set was last modified.

- **OptionSettings** (*list*) –

A list of the configuration options and their values in this configuration set.

- (*dict*) –

A specification identifying an individual configuration option along with its current value. For a list of possible option values, go to [Option Values](#) in the *AWS Elastic Beanstalk Developer Guide*.

- **ResourceName** (*string*) –

A unique resource name for a time-based scaling configuration option.

- **Namespace** (*string*) –

A unique namespace identifying the option's associated AWS resource.

- **OptionName** (*string*) –

The name of the configuration option.

- **Value** (*string*) –

The current value for the configuration option.

describe_environment_health (**kwargs)

Returns information about the overall health of the specified environment. The **DescribeEnvironmentHealth** operation is only available with AWS Elastic Beanstalk Enhanced Health.

Request Syntax

```
response = client.describe_environment_health(
    EnvironmentName='string',
    EnvironmentId='string',
    AttributeNames=[
        'Status'|'Color'|'Causes'|'ApplicationMetrics'|'InstancesHealth'|'All'|'HealthStatus'
    ]
)
```

Parameters

- **EnvironmentName** (*string*) – Specifies the AWS Elastic Beanstalk environment name.
- **EnvironmentId** (*string*) – Specifies the AWS Elastic Beanstalk environment ID.
- **AttributeNames** (*list*) – Specifies the response elements you wish to receive. If no attribute names are specified, AWS Elastic Beanstalk returns all response elements.
 - (*string*) –

Return type dict

Returns**Response Syntax**

```
{
    'EnvironmentName': 'string',
    'HealthStatus': 'string',
    'Status': 'Green'|'Yellow'|'Red'|'Grey',
    'Color': 'string',
    'Causes': [
        'string',
    ],
    'ApplicationMetrics': {
        'Duration': 123,
        'RequestCount': 123,
        'StatusCodes': {
            'Status2xx': 123,
            'Status3xx': 123,
            'Status4xx': 123,
            'Status5xx': 123
        },
        'Latency': {
            'P999': 123.0,
            'P99': 123.0,
            'P95': 123.0,
            'P90': 123.0,
            'P85': 123.0,
            'P75': 123.0,
            'P50': 123.0,
            'P10': 123.0
        }
    },
    'InstancesHealth': {
        'NoData': 123,
        'Unknown': 123,
```



```

        'Pending': 123,
        'Ok': 123,
        'Info': 123,
        'Warning': 123,
        'Degraded': 123,
        'Severe': 123
    },
    'RefreshedAt': datetime(2015, 1, 1)
}

```

Response Structure

- (*dict*) –

See the example below for a sample response.

- **EnvironmentName** (*string*) –

The AWS Elastic Beanstalk environment name.

- **HealthStatus** (*string*) –

Contains the response body with information about the health of the environment.

- **Status** (*string*) –

Returns the health status value of the environment. For more information, see [Health Colors and Statuses](#).

- **Color** (*string*) –

Returns the color indicator that tells you information about the health of the environment. For more information, see [Health Colors and Statuses](#).

- **Causes** (*list*) –

Returns potential causes for the reported status.

- * (*string*) –

- **ApplicationMetrics** (*dict*) –

Represents the application metrics for a specified environment.

- * **Duration** (*integer*) –

The amount of time that the metrics cover (usually 10 seconds). For example, you might have 5 requests (`request_count`) within the most recent time slice of 10 seconds (`duration`).

- * **RequestCount** (*integer*) –

Average number of requests handled by the web server per second over the last 10 seconds.

- * **StatusCodes** (*dict*) –

Represents the percentage of requests over the last 10 seconds that resulted in each type of status code response.

- **Status2xx** (*integer*) –

The percentage of requests over the last 10 seconds that resulted in a 2xx (200, 201, etc.) status code.

- **Status3xx** (*integer*) –

The percentage of requests over the last 10 seconds that resulted in a 3xx (300, 301, etc.) status code.

- **Status4xx** (*integer*) –

The percentage of requests over the last 10 seconds that resulted in a 4xx (400, 401, etc.) status code.

- **Status5xx** (*integer*) –

The percentage of requests over the last 10 seconds that resulted in a 5xx (500, 501, etc.) status code.

- * **Latency** (*dict*) –

Represents the average latency for the slowest X percent of requests over the last 10 seconds. Latencies are in seconds with one millisecond resolution.

- **P999** (*float*) –

The average latency for the slowest 0.1 percent of requests over the last 10 seconds.

- **P99** (*float*) –

The average latency for the slowest 1 percent of requests over the last 10 seconds.

- **P95** (*float*) –

The average latency for the slowest 5 percent of requests over the last 10 seconds.

- **P90** (*float*) –

The average latency for the slowest 10 percent of requests over the last 10 seconds.

- **P85** (*float*) –

The average latency for the slowest 15 percent of requests over the last 10 seconds.

- **P75** (*float*) –

The average latency for the slowest 25 percent of requests over the last 10 seconds.

- **P50** (*float*) –

The average latency for the slowest 50 percent of requests over the last 10 seconds.

- **P10** (*float*) –

The average latency for the slowest 90 percent of requests over the last 10 seconds.

- **InstancesHealth** (*dict*) –

Represents summary information about the health of an instance. For more information, see [Health Colors and Statuses](#) .

- * **NoData** (*integer*) –

Grey. AWS Elastic Beanstalk and the health agent are reporting no data on an instance.

- * **Unknown** (*integer*) –

Grey. AWS Elastic Beanstalk and the health agent are reporting an insufficient amount of data on an instance.

- * **Pending** (*integer*) –

Grey. An operation is in progress on an instance within the command timeout.

- * **Ok** (*integer*) –

Green. An instance is passing health checks and the health agent is not reporting any problems.

* **Info** (*integer*) –

Green. An operation is in progress on an instance.

* **Warning** (*integer*) –

Yellow. The health agent is reporting a moderate number of request failures or other issues for an instance or environment.

* **Degraded** (*integer*) –

Red. The health agent is reporting a high number of request failures or other issues for an instance or environment.

* **Severe** (*integer*) –

Red. The health agent is reporting a very high number of request failures or other issues for an instance or environment.

– **RefreshedAt** (*datetime*) –

The date and time the information was last refreshed.

describe_environment_resources (***kwargs*)

Returns AWS resources for this environment.

Request Syntax

```
response = client.describe_environment_resources(
    EnvironmentId='string',
    EnvironmentName='string'
)
```

Parameters

- **EnvironmentId** (*string*) – The ID of the environment to retrieve AWS resource usage data.

Condition: You must specify either this or an EnvironmentName, or both. If you do not specify either, AWS Elastic Beanstalk returns `MissingRequiredParameter` error.

- **EnvironmentName** (*string*) – The name of the environment to retrieve AWS resource usage data.

Condition: You must specify either this or an EnvironmentId, or both. If you do not specify either, AWS Elastic Beanstalk returns `MissingRequiredParameter` error.

Return type dict

Returns

Response Syntax

```
{
  'EnvironmentResources': {
    'EnvironmentName': 'string',
    'AutoScalingGroups': [
      {
        'Name': 'string'
      },
    ],
    'Instances': [
      {
        'Id': 'string'
      },
    ],
  },
}
```

```
    'LaunchConfigurations': [
        {
            'Name': 'string'
        },
    ],
    'LoadBalancers': [
        {
            'Name': 'string'
        },
    ],
    'Triggers': [
        {
            'Name': 'string'
        },
    ],
    'Queues': [
        {
            'Name': 'string',
            'URL': 'string'
        },
    ],
]
```

Response Structure

- *(dict)* –

Result message containing a list of environment resource descriptions.

- **EnvironmentResources** (*dict*) –

A list of EnvironmentResourceDescription .

- * **EnvironmentName** (*string*) –

The name of the environment.

- * **AutoScalingGroups** (*list*) –

The AutoScalingGroups used by this environment.

· (*dict*) –

Describes an Auto Scaling launch configuration.

· **Name** (*string*) –

The name of the AutoScalingGroup .

- * **Instances** (*list*) –

The Amazon EC2 instances used by this environment.

· (*dict*) –

The description of an Amazon EC2 instance.

· **Id** (*string*) –

The ID of the Amazon EC2 instance.

- * **LaunchConfigurations** (*list*) –

The Auto Scaling launch configurations in use by this environment.

· (*dict*) –

Describes an Auto Scaling launch configuration.

· **Name** (*string*) –

The name of the launch configuration.

* **LoadBalancers** (*list*) –

The LoadBalancers in use by this environment.

· (*dict*) –

Describes a LoadBalancer.

· **Name** (*string*) –

The name of the LoadBalancer.

* **Triggers** (*list*) –

The AutoScaling triggers in use by this environment.

· (*dict*) –

Describes a trigger.

· **Name** (*string*) –

The name of the trigger.

* **Queues** (*list*) –

The queues used by this environment.

· (*dict*) –

Describes a queue.

· **Name** (*string*) –

The name of the queue.

· **URL** (*string*) –

The URL of the queue.

describe_environments (***kwargs*)

Returns descriptions for existing environments.

Request Syntax

```
response = client.describe_environments(
    ApplicationName='string',
    VersionLabel='string',
    EnvironmentIds=[
        'string',
    ],
    EnvironmentNames=[
        'string',
    ],
    IncludeDeleted=True|False,
    IncludedDeletedBackTo=datetime(2015, 1, 1)
)
```

Parameters

- **ApplicationName** (*string*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to include only those that are associated with this application.
- **VersionLabel** (*string*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to include only those that are associated with this application version.
- **EnvironmentIds** (*list*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to include only those that have the specified IDs.
– (*string*) –
- **EnvironmentNames** (*list*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to include only those that have the specified names.

- (*string*) –
- **IncludeDeleted** (*boolean*) – Indicates whether to include deleted environments:
 - true : Environments that have been deleted after IncludedDeletedBackTo are displayed.
 - false : Do not include deleted environments.
- **IncludedDeletedBackTo** (*datetime*) – If specified when IncludeDeleted is set to true, then environments deleted after this date are displayed.

Return type dict

Returns

Response Syntax

```
{
  'Environments': [
    {
      'EnvironmentName': 'string',
      'EnvironmentId': 'string',
      'ApplicationName': 'string',
      'VersionLabel': 'string',
      'SolutionStackName': 'string',
      'TemplateName': 'string',
      'Description': 'string',
      'EndpointURL': 'string',
      'CNAME': 'string',
      'DateCreated': datetime(2015, 1, 1),
      'DateUpdated': datetime(2015, 1, 1),
      'Status': 'Launching'|'Updating'|'Ready'|'Terminating'|'Terminated',
      'AbortableOperationInProgress': True|False,
      'Health': 'Green'|'Yellow'|'Red'|'Grey',
      'HealthStatus': 'NoData'|'Unknown'|'Pending'|'Ok'|'Info'|'Warning'|'De
      'Resources': {
        'LoadBalancer': {
          'LoadBalancerName': 'string',
          'Domain': 'string',
          'Listeners': [
            {
              'Protocol': 'string',
              'Port': 123
            }
          ],
        }
      },
      'Tier': {
        'Name': 'string',
        'Type': 'string',
        'Version': 'string'
      }
    },
  ]
}
```

Response Structure

- (*dict*) –
 - Result message containing a list of environment descriptions.
 - **Environments** (*list*) –
 - Returns an EnvironmentDescription list.

* (*dict*) –

Describes the properties of an environment.

- **EnvironmentName** (*string*) –
The name of this environment.
- **EnvironmentId** (*string*) –
The ID of this environment.
- **ApplicationName** (*string*) –
The name of the application associated with this environment.
- **VersionLabel** (*string*) –
The application version deployed in this environment.
- **SolutionStackName** (*string*) –
The name of the `SolutionStack` deployed with this environment.
- **TemplateName** (*string*) –
The name of the configuration template used to originally launch this environment.
- **Description** (*string*) –
Describes this environment.
- **EndpointURL** (*string*) –
For load-balanced, autoscaling environments, the URL to the LoadBalancer. For single-instance environments, the IP address of the instance.
- **CNAME** (*string*) –
The URL to the CNAME for this environment.
- **DateCreated** (*datetime*) –
The creation date for this environment.
- **DateUpdated** (*datetime*) –
The last modified date for this environment.
- **Status** (*string*) –
The current operational status of the environment:
 - `Launching` : Environment is in the process of initial deployment.
 - `Updating` : Environment is in the process of updating its configuration settings or application version.
 - `Ready` : Environment is available to have an action performed on it, such as update or terminate.
 - `Terminating` : Environment is in the shut-down process.
 - `Terminated` : Environment is not running.
- **AbortableOperationInProgress** (*boolean*) –
Indicates if there is an in-progress environment configuration update or application version deployment that you can cancel.
 - `true` : There is an update in progress.
 - `false` : There are no updates currently in progress.
- **Health** (*string*) –

Describes the health status of the environment. AWS Elastic Beanstalk indicates the failure levels for a running environment:

Red : Indicates the environment is not working.

Yellow : Indicates that something is wrong, the application might not be available, but the instances appear running.

Green : Indicates the environment is healthy and fully functional.

- Red : Indicates the environment is not responsive. Occurs when three or more consecutive failures occur for an environment.
- Yellow : Indicates that something is wrong. Occurs when two consecutive failures occur for an environment.
- Green : Indicates the environment is healthy and fully functional.
- Grey : Default health for a new environment. The environment is not fully launched and health checks have not started or health checks are suspended during an `UpdateEnvironment` or `RestartEnvironment` request.

Default: Grey

- **HealthStatus** (*string*) –

Returns the health status of the application running in your environment. For more information, see [Health Colors and Statuses](#).

- **Resources** (*dict*) –

The description of the AWS resources used by this environment.

- **LoadBalancer** (*dict*) –

Describes the LoadBalancer.

- **LoadBalancerName** (*string*) –

The name of the LoadBalancer.

- **Domain** (*string*) –

The domain name of the LoadBalancer.

- **Listeners** (*list*) –

A list of Listeners used by the LoadBalancer.

- (*dict*) –

Describes the properties of a Listener for the LoadBalancer.

- **Protocol** (*string*) –

The protocol that is used by the Listener.

- **Port** (*integer*) –

The port that is used by the Listener.

- **Tier** (*dict*) –

Describes the current tier of this environment.

- **Name** (*string*) –

The name of this environment tier.

- **Type** (*string*) –
The type of this environment tier.
- **Version** (*string*) –
The version of this environment tier.

describe_events (***kwargs*)

Returns list of event descriptions matching criteria up to the last 6 weeks.

Note: This action returns the most recent 1,000 events from the specified `NextToken`.

Request Syntax

```
response = client.describe_events(
    ApplicationName='string',
    VersionLabel='string',
    TemplateName='string',
    EnvironmentId='string',
    EnvironmentName='string',
    RequestId='string',
    Severity='TRACE' | 'DEBUG' | 'INFO' | 'WARN' | 'ERROR' | 'FATAL',
    StartTime=datetime(2015, 1, 1),
    EndTime=datetime(2015, 1, 1),
    MaxRecords=123,
    NextToken='string'
)
```

Parameters

- **ApplicationName** (*string*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to include only those associated with this application.
- **VersionLabel** (*string*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to those associated with this application version.
- **TemplateName** (*string*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to those that are associated with this environment configuration.
- **EnvironmentId** (*string*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to those associated with this environment.
- **EnvironmentName** (*string*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to those associated with this environment.
- **RequestId** (*string*) – If specified, AWS Elastic Beanstalk restricts the described events to include only those associated with this request ID.
- **Severity** (*string*) – If specified, limits the events returned from this call to include only those with the specified severity or higher.
- **StartTime** (*datetime*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to those that occur on or after this time.
- **EndTime** (*datetime*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to those that occur up to, but not including, the `EndTime`.
- **MaxRecords** (*integer*) – Specifies the maximum number of events that can be returned, beginning with the most recent event.
- **NextToken** (*string*) – Pagination token. If specified, the events return the next batch of results.

Return type dict

Returns

Response Syntax

```
{
  'Events': [
    {
      'EventDate': datetime(2015, 1, 1),
      'Message': 'string',
      'ApplicationName': 'string',
      'VersionLabel': 'string',
      'TemplateName': 'string',
      'EnvironmentName': 'string',
      'RequestId': 'string',
      'Severity': 'TRACE' | 'DEBUG' | 'INFO' | 'WARN' | 'ERROR' | 'FATAL'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Result message wrapping a list of event descriptions.

- **Events** (*list*) –

A list of EventDescription .

- * (*dict*) –

Describes an event.

- **EventDate** (*datetime*) –

The date when the event occurred.

- **Message** (*string*) –

The event message.

- **ApplicationName** (*string*) –

The application associated with the event.

- **VersionLabel** (*string*) –

The release label for the application version associated with this event.

- **TemplateName** (*string*) –

The name of the configuration associated with this event.

- **EnvironmentName** (*string*) –

The name of the environment associated with this event.

- **RequestId** (*string*) –

The web service request ID for the activity of this event.

- **Severity** (*string*) –

The severity level of this event.

- **NextToken** (*string*) –

If returned, this indicates that there are more results to obtain. Use this token in the next DescribeEvents call to get the next batch of events.

describe_instances_health (***kwargs*)

Returns more detailed information about the health of the specified instances (for example, CPU utilization, load average, and causes). The **DescribeInstancesHealth** operation is only available with AWS Elastic Beanstalk Enhanced Health.

Request Syntax

```
response = client.describe_instances_health(
    EnvironmentName='string',
    EnvironmentId='string',
    AttributeNames=[
        'HealthStatus'|'Color'|'Causes'|'ApplicationMetrics'|'RefreshedAt'|'LaunchedAt'|'Sy
    ],
    NextToken='string'
)
```

Parameters

- **EnvironmentName** (*string*) – Specifies the AWS Elastic Beanstalk environment name.
- **EnvironmentId** (*string*) – Specifies the AWS Elastic Beanstalk environment ID.
- **AttributeNames** (*list*) – Specifies the response elements you wish to receive. If no attribute names are specified, AWS Elastic Beanstalk returns all response elements.
 - (*string*) –
- **NextToken** (*string*) – Specifies the next token of the request.

Return type dict

Returns

Response Syntax

```
{
    'InstanceHealthList': [
        {
            'InstanceId': 'string',
            'HealthStatus': 'string',
            'Color': 'string',
            'Causes': [
                'string',
            ],
            'LaunchedAt': datetime(2015, 1, 1),
            'ApplicationMetrics': {
                'Duration': 123,
                'RequestCount': 123,
                'StatusCodes': {
                    'Status2xx': 123,
                    'Status3xx': 123,
                    'Status4xx': 123,
                    'Status5xx': 123
                },
                'Latency': {
                    'P999': 123.0,
                    'P99': 123.0,
                    'P95': 123.0,
                    'P90': 123.0,
                    'P85': 123.0,
                    'P75': 123.0,
                    'P50': 123.0,
                    'P10': 123.0
                }
            },
            'System': {
                'CPUUtilization': {
                    'User': 123.0,
```

```
        'Nice': 123.0,
        'System': 123.0,
        'Idle': 123.0,
        'IOWait': 123.0,
        'IRQ': 123.0,
        'SoftIRQ': 123.0
    },
    'LoadAverage': [
        123.0,
    ]
}
},
],
'RefreshedAt': datetime(2015, 1, 1),
'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

See the example below for a sample response.

- **InstanceHealthList** (*list*) –

Contains the response body with information about the health of the instance.

- * (*dict*) –

Represents health information from the specified instance that belongs to the AWS Elastic Beanstalk environment. Use the `InstanceId` property to specify the application instance for which you'd like to return data.

- **InstanceId** (*string*) –

The ID of the Amazon EC2 instance.

- **HealthStatus** (*string*) –

Returns the health status of the specified instance. For more information, see <http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/health-enhanced-status.html>.

- **Color** (*string*) –

Represents the color indicator that gives you information about the health of the EC2 instance. For more information, see [Health Colors and Statuses](#).

- **Causes** (*list*) –

Represents the causes, which provide more information about the current health status.

- (*string*) –

- **LaunchedAt** (*datetime*) –

The time at which the EC2 instance was launched.

- **ApplicationMetrics** (*dict*) –

Represents the application metrics for a specified environment.

- **Duration** (*integer*) –

The amount of time that the metrics cover (usually 10

seconds). For example, you might have 5 requests (`request_count`) within the most recent time slice of 10 seconds (`duration`).

- **RequestCount** (*integer*) –

Average number of requests handled by the web server per second over the last 10 seconds.

- **StatusCodes** (*dict*) –

Represents the percentage of requests over the last 10 seconds that resulted in each type of status code response.

- **Status2xx** (*integer*) –

The percentage of requests over the last 10 seconds that resulted in a 2xx (200, 201, etc.) status code.

- **Status3xx** (*integer*) –

The percentage of requests over the last 10 seconds that resulted in a 3xx (300, 301, etc.) status code.

- **Status4xx** (*integer*) –

The percentage of requests over the last 10 seconds that resulted in a 4xx (400, 401, etc.) status code.

- **Status5xx** (*integer*) –

The percentage of requests over the last 10 seconds that resulted in a 5xx (500, 501, etc.) status code.

- **Latency** (*dict*) –

Represents the average latency for the slowest X percent of requests over the last 10 seconds. Latencies are in seconds with one millisecond resolution.

- **P999** (*float*) –

The average latency for the slowest 0.1 percent of requests over the last 10 seconds.

- **P99** (*float*) –

The average latency for the slowest 1 percent of requests over the last 10 seconds.

- **P95** (*float*) –

The average latency for the slowest 5 percent of requests over the last 10 seconds.

- **P90** (*float*) –

The average latency for the slowest 10 percent of requests over the last 10 seconds.

- **P85** (*float*) –

The average latency for the slowest 15 percent of requests over the last 10 seconds.

- **P75** (*float*) –

The average latency for the slowest 25 percent of requests over the last 10 seconds.

- **P50** (*float*) –

The average latency for the slowest 50 percent of requests over the last 10 seconds.

- **P10** (*float*) –
The average latency for the slowest 90 percent of requests over the last 10 seconds.
- **System** (*dict*) –
Represents CPU utilization and load average information for applications running in the specified environment.
- **CPUUtilization** (*dict*) –
Represents CPU utilization information from the specified instance that belongs to the AWS Elastic Beanstalk environment. Use the `instanceId` property to specify the application instance for which you'd like to return data.
- **User** (*float*) –
Percentage of time that the CPU has spent in the `User` state over the last 10 seconds.
- **Nice** (*float*) –
Percentage of time that the CPU has spent in the `Nice` state over the last 10 seconds.
- **System** (*float*) –
Percentage of time that the CPU has spent in the `System` state over the last 10 seconds.
- **Idle** (*float*) –
Percentage of time that the CPU has spent in the `Idle` state over the last 10 seconds.
- **IOWait** (*float*) –
Percentage of time that the CPU has spent in the `I/O Wait` state over the last 10 seconds.
- **IRQ** (*float*) –
Percentage of time that the CPU has spent in the `IRQ` state over the last 10 seconds.
- **SoftIRQ** (*float*) –
Percentage of time that the CPU has spent in the `SoftIRQ` state over the last 10 seconds.
- **LoadAverage** (*list*) –
Load average in the last 1-minute and 5-minute periods. For more information, see [Operating System Metrics](#).
- (*float*) –
- **RefreshedAt** (*datetime*) –
The date and time the information was last refreshed.
- **NextToken** (*string*) –
The next token.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)

- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type L{botocore.paginate.Paginator}

Returns A paginator object.

get_waiter (*waiter_name*)

list_available_solution_stacks ()

Returns a list of the available solution stack names.

Request Syntax

```
response = client.list_available_solution_stacks()
```

Return type dict

Returns

Response Syntax

```
{
  'SolutionStacks': [
    'string',
  ],
  'SolutionStackDetails': [
    {
      'SolutionStackName': 'string',
      'PermittedFileTypes': [
        'string',
      ]
    },
  ]
}
```

Response Structure

- (*dict*) –

A list of available AWS Elastic Beanstalk solution stacks.

- **SolutionStacks** (*list*) –

A list of available solution stacks.

- * (*string*) –

- **SolutionStackDetails** (*list*) –

A list of available solution stacks and their `SolutionStackDescription`.

- * (*dict*) –

Describes the solution stack.

- **SolutionStackName** (*string*) –
The name of the solution stack.
- **PermittedFileTypes** (*list*) –
The permitted file types allowed for a solution stack.
- (*string*) –

rebuild_environment (***kwargs*)

Deletes and recreates all of the AWS resources (for example: the Auto Scaling group, load balancer, etc.) for a specified environment and forces a restart.

Request Syntax

```
response = client.rebuild_environment(  
    EnvironmentId='string',  
    EnvironmentName='string'  
)
```

Parameters

- **EnvironmentId** (*string*) – The ID of the environment to rebuild.
Condition: You must specify either this or an EnvironmentName, or both. If you do not specify either, AWS Elastic Beanstalk returns `MissingRequiredParameter` error.
- **EnvironmentName** (*string*) – The name of the environment to rebuild.
Condition: You must specify either this or an EnvironmentId, or both. If you do not specify either, AWS Elastic Beanstalk returns `MissingRequiredParameter` error.

Returns None

request_environment_info (***kwargs*)

Initiates a request to compile the specified type of information of the deployed environment.

Setting the `InfoType` to `tail` compiles the last lines from the application server log files of every Amazon EC2 instance in your environment.

Setting the `InfoType` to `bundle` compresses the application server log files for every Amazon EC2 instance into a `.zip` file. Legacy and .NET containers do not support bundle logs.

Use `RetrieveEnvironmentInfo` to obtain the set of logs.

Related Topics

- `RetrieveEnvironmentInfo`

Request Syntax

```
response = client.request_environment_info(  
    EnvironmentId='string',  
    EnvironmentName='string',  
    InfoType='tail'|'bundle'  
)
```

Parameters

- **EnvironmentId** (*string*) – The ID of the environment of the requested data.
If no such environment is found, `RequestEnvironmentInfo` returns an `InvalidParameterValue` error.

Condition: You must specify either this or an `EnvironmentName`, or both. If you do not specify either, AWS Elastic Beanstalk returns `MissingRequiredParameter` error.

- **EnvironmentName** (*string*) – The name of the environment of the requested data.

If no such environment is found, `RequestEnvironmentInfo` returns an `InvalidParameterValue` error.

Condition: You must specify either this or an `EnvironmentId`, or both. If you do not specify either, AWS Elastic Beanstalk returns `MissingRequiredParameter` error.

- **InfoType** (*string*) – [REQUIRED]

The type of information to request.

Returns None

restart_app_server (***kwargs*)

Causes the environment to restart the application container server running on each Amazon EC2 instance.

Request Syntax

```
response = client.restart_app_server(
    EnvironmentId='string',
    EnvironmentName='string'
)
```

Parameters

- **EnvironmentId** (*string*) – The ID of the environment to restart the server for.

Condition: You must specify either this or an `EnvironmentName`, or both. If you do not specify either, AWS Elastic Beanstalk returns `MissingRequiredParameter` error.

- **EnvironmentName** (*string*) – The name of the environment to restart the server for.

Condition: You must specify either this or an `EnvironmentId`, or both. If you do not specify either, AWS Elastic Beanstalk returns `MissingRequiredParameter` error.

Returns None

retrieve_environment_info (***kwargs*)

Retrieves the compiled information from a `RequestEnvironmentInfo` request.

Related Topics

- `RequestEnvironmentInfo`

Request Syntax

```
response = client.retrieve_environment_info(
    EnvironmentId='string',
    EnvironmentName='string',
    InfoType='tail'|'bundle'
)
```

Parameters

- **EnvironmentId** (*string*) – The ID of the data's environment.

If no such environment is found, returns an `InvalidParameterValue` error.

Condition: You must specify either this or an `EnvironmentName`, or both. If you do not specify either, AWS Elastic Beanstalk returns `MissingRequiredParameter` error.

- **EnvironmentName** (*string*) – The name of the data's environment.

If no such environment is found, returns an `InvalidParameterValue` error.

Condition: You must specify either this or an `EnvironmentId`, or both. If you do not specify either, AWS Elastic Beanstalk returns `MissingRequiredParameter` error.

- **InfoType** (*string*) – **[REQUIRED]**

The type of information to retrieve.

Return type dict

Returns

Response Syntax

```
{
  'EnvironmentInfo': [
    {
      'InfoType': 'tail'|'bundle',
      'Ec2InstanceId': 'string',
      'SampleTimestamp': datetime(2015, 1, 1),
      'Message': 'string'
    },
  ]
}
```

Response Structure

- (*dict*) –

Result message containing a description of the requested environment info.

- **EnvironmentInfo** (*list*) –

The `EnvironmentInfoDescription` of the environment.

- * (*dict*) –

The information retrieved from the Amazon EC2 instances.

- **InfoType** (*string*) –

The type of information retrieved.

- **Ec2InstanceId** (*string*) –

The Amazon EC2 Instance ID for this information.

- **SampleTimestamp** (*datetime*) –

The time stamp when this information was retrieved.

- **Message** (*string*) –

The retrieved information.

swap_environment_cnames (***kwargs*)

Swaps the CNAMEs of two environments.

Request Syntax

```
response = client.swap_environment_cnames(
    SourceEnvironmentId='string',
    SourceEnvironmentName='string',
    DestinationEnvironmentId='string',
```

```
DestinationEnvironmentName='string'
)
```

Parameters

- **SourceEnvironmentId** (*string*) – The ID of the source environment.
Condition: You must specify at least the SourceEnvironmentID or the SourceEnvironmentName . You may also specify both. If you specify the SourceEnvironmentId , you must specify the DestinationEnvironmentId.
- **SourceEnvironmentName** (*string*) – The name of the source environment.
Condition: You must specify at least the SourceEnvironmentID or the SourceEnvironmentName . You may also specify both. If you specify the SourceEnvironmentName , you must specify the DestinationEnvironmentName.
- **DestinationEnvironmentId** (*string*) – The ID of the destination environment.
Condition: You must specify at least the DestinationEnvironmentID or the DestinationEnvironmentName . You may also specify both. You must specify the SourceEnvironmentId with the DestinationEnvironmentId.
- **DestinationEnvironmentName** (*string*) – The name of the destination environment.
Condition: You must specify at least the DestinationEnvironmentID or the DestinationEnvironmentName . You may also specify both. You must specify the SourceEnvironmentName with the DestinationEnvironmentName.

Returns None

terminate_environment (**kwargs)
Terminates the specified environment.

Request Syntax

```
response = client.terminate_environment(
    EnvironmentId='string',
    EnvironmentName='string',
    TerminateResources=True|False
)
```

Parameters

- **EnvironmentId** (*string*) – The ID of the environment to terminate.
Condition: You must specify either this or an EnvironmentName, or both. If you do not specify either, AWS Elastic Beanstalk returns MissingRequiredParameter error.
- **EnvironmentName** (*string*) – The name of the environment to terminate.
Condition: You must specify either this or an EnvironmentId, or both. If you do not specify either, AWS Elastic Beanstalk returns MissingRequiredParameter error.
- **TerminateResources** (*boolean*) – Indicates whether the associated AWS resources should shut down when the environment is terminated:
true : (default) The user AWS resources (for example, the Auto Scaling group, LoadBalancer, etc.) are terminated along with the environment.

`false` : The environment is removed from the AWS Elastic Beanstalk but the AWS resources continue to operate.

- `true` : The specified environment as well as the associated AWS resources, such as Auto Scaling group and LoadBalancer, are terminated.
- `false` : AWS Elastic Beanstalk resource management is removed from the environment, but the AWS resources continue to operate.

For more information, see the [AWS Elastic Beanstalk User Guide](#).

Default: `true`

Valid Values: `true|false`

Return type dict

Returns

Response Syntax

```
{
  'EnvironmentName': 'string',
  'EnvironmentId': 'string',
  'ApplicationName': 'string',
  'VersionLabel': 'string',
  'SolutionStackName': 'string',
  'TemplateName': 'string',
  'Description': 'string',
  'EndpointURL': 'string',
  'CNAME': 'string',
  'DateCreated': datetime(2015, 1, 1),
  'DateUpdated': datetime(2015, 1, 1),
  'Status': 'Launching'|'Updating'|'Ready'|'Terminating'|'Terminated',
  'AbortableOperationInProgress': True|False,
  'Health': 'Green'|'Yellow'|'Red'|'Grey',
  'HealthStatus': 'NoData'|'Unknown'|'Pending'|'Ok'|'Info'|'Warning'|'Degraded'|
  'Resources': {
    'LoadBalancer': {
      'LoadBalancerName': 'string',
      'Domain': 'string',
      'Listeners': [
        {
          'Protocol': 'string',
          'Port': 123
        },
      ],
    },
  },
  'Tier': {
    'Name': 'string',
    'Type': 'string',
    'Version': 'string'
  }
}
```

Response Structure

- (*dict*) –

Describes the properties of an environment.

- **EnvironmentName** (*string*) –

The name of this environment.

- **EnvironmentId** (*string*) –
The ID of this environment.
- **ApplicationName** (*string*) –
The name of the application associated with this environment.
- **VersionLabel** (*string*) –
The application version deployed in this environment.
- **SolutionStackName** (*string*) –
The name of the `SolutionStack` deployed with this environment.
- **TemplateName** (*string*) –
The name of the configuration template used to originally launch this environment.
- **Description** (*string*) –
Describes this environment.
- **EndpointURL** (*string*) –
For load-balanced, autoscaling environments, the URL to the LoadBalancer. For single-instance environments, the IP address of the instance.
- **CNAME** (*string*) –
The URL to the CNAME for this environment.
- **DateCreated** (*datetime*) –
The creation date for this environment.
- **DateUpdated** (*datetime*) –
The last modified date for this environment.
- **Status** (*string*) –
The current operational status of the environment:
 - * `Launching` : Environment is in the process of initial deployment.
 - * `Updating` : Environment is in the process of updating its configuration settings or application version.
 - * `Ready` : Environment is available to have an action performed on it, such as update or terminate.
 - * `Terminating` : Environment is in the shut-down process.
 - * `Terminated` : Environment is not running.
- **AbortableOperationInProgress** (*boolean*) –
Indicates if there is an in-progress environment configuration update or application version deployment that you can cancel.
 - `true` : There is an update in progress.
 - `false` : There are no updates currently in progress.
- **Health** (*string*) –
Describes the health status of the environment. AWS Elastic Beanstalk indicates the failure levels for a running environment:
 - `Red` : Indicates the environment is not working.
 - `Yellow` : Indicates that something is wrong, the application might not be available, but the instances appear running.
 - `Green` : Indicates the environment is healthy and fully functional.

- * **Red** : Indicates the environment is not responsive. Occurs when three or more consecutive failures occur for an environment.
- * **Yellow** : Indicates that something is wrong. Occurs when two consecutive failures occur for an environment.
- * **Green** : Indicates the environment is healthy and fully functional.
- * **Grey** : Default health for a new environment. The environment is not fully launched and health checks have not started or health checks are suspended during an `UpdateEnvironment` or `RestartEnvironment` request.

Default: `Grey`

– **HealthStatus** (*string*) –

Returns the health status of the application running in your environment. For more information, see [Health Colors and Statuses](#) .

– **Resources** (*dict*) –

The description of the AWS resources used by this environment.

* **LoadBalancer** (*dict*) –

Describes the LoadBalancer.

· **LoadBalancerName** (*string*) –

The name of the LoadBalancer.

· **Domain** (*string*) –

The domain name of the LoadBalancer.

· **Listeners** (*list*) –

A list of Listeners used by the LoadBalancer.

· (*dict*) –

Describes the properties of a Listener for the LoadBalancer.

· **Protocol** (*string*) –

The protocol that is used by the Listener.

· **Port** (*integer*) –

The port that is used by the Listener.

– **Tier** (*dict*) –

Describes the current tier of this environment.

* **Name** (*string*) –

The name of this environment tier.

* **Type** (*string*) –

The type of this environment tier.

* **Version** (*string*) –

The version of this environment tier.

update_application (***kwargs*)

Updates the specified application to have the specified properties.

Note: If a property (for example, `description`) is not provided, the value remains unchanged. To clear these properties, specify an empty string.

Request Syntax

```
response = client.update_application(
    ApplicationName='string',
    Description='string'
)
```

Parameters

- **ApplicationName** (*string*) – [REQUIRED]

The name of the application to update. If no such application is found, `UpdateApplication` returns an `InvalidParameterValue` error.

- **Description** (*string*) – A new description for the application.

Default: If not specified, AWS Elastic Beanstalk does not update the description.

Return type dict

Returns

Response Syntax

```
{
  'Application': {
    'ApplicationName': 'string',
    'Description': 'string',
    'DateCreated': datetime(2015, 1, 1),
    'DateUpdated': datetime(2015, 1, 1),
    'Versions': [
      'string',
    ],
    'ConfigurationTemplates': [
      'string',
    ]
  }
}
```

Response Structure

- (*dict*) –

Result message containing a single description of an application.

- **Application** (*dict*) –

The `ApplicationDescription` of the application.

- * **ApplicationName** (*string*) –

The name of the application.

- * **Description** (*string*) –

User-defined description of the application.

- * **DateCreated** (*datetime*) –

The date when the application was created.

- * **DateUpdated** (*datetime*) –

The date when the application was last modified.

- * **Versions** (*list*) –

The names of the versions for this application.

· (*string*) –

- * **ConfigurationTemplates** (*list*) –

The names of the configuration templates associated with this application.

· (*string*) –

update_application_version (***kwargs*)

Updates the specified application version to have the specified properties.

Note: If a property (for example, `description`) is not provided, the value remains unchanged. To clear properties, specify an empty string.

Request Syntax

```
response = client.update_application_version(
    ApplicationName='string',
    VersionLabel='string',
    Description='string'
)
```

Parameters

- **ApplicationName** (*string*) – [REQUIRED]

The name of the application associated with this version.

If no application is found with this name, `UpdateApplication` returns an `InvalidParameterValue` error.

- **VersionLabel** (*string*) – [REQUIRED]

The name of the version to update.

If no application version is found with this label, `UpdateApplication` returns an `InvalidParameterValue` error.

- **Description** (*string*) – A new description for this release.

Return type dict

Returns

Response Syntax

```
{
  'ApplicationVersion': {
    'ApplicationName': 'string',
    'Description': 'string',
    'VersionLabel': 'string',
    'SourceBundle': {
      'S3Bucket': 'string',
      'S3Key': 'string'
    },
    'DateCreated': datetime(2015, 1, 1),
    'DateUpdated': datetime(2015, 1, 1)
  }
}
```

Response Structure

- (*dict*) –

Result message wrapping a single description of an application version.

- **ApplicationVersion** (*dict*) –

The `ApplicationVersionDescription` of the application version.

- * **ApplicationName** (*string*) –

The name of the application associated with this release.

- * **Description** (*string*) –
The description of this application version.
- * **VersionLabel** (*string*) –
A label uniquely identifying the version for the associated application.
- * **SourceBundle** (*dict*) –
The location where the source bundle is located for this version.
 - **S3Bucket** (*string*) –
The Amazon S3 bucket where the data is located.
 - **S3Key** (*string*) –
The Amazon S3 key where the data is located.
- * **DateCreated** (*datetime*) –
The creation date of the application version.
- * **DateUpdated** (*datetime*) –
The last modified date of the application version.

update_configuration_template (***kwargs*)

Updates the specified configuration template to have the specified properties or configuration option values.

Note: If a property (for example, `ApplicationName`) is not provided, its value remains unchanged. To clear such properties, specify an empty string.

Related Topics

- [DescribeConfigurationOptions](#)

Request Syntax

```
response = client.update_configuration_template(
    ApplicationName='string',
    TemplateName='string',
    Description='string',
    OptionSettings=[
        {
            'ResourceName': 'string',
            'Namespace': 'string',
            'OptionName': 'string',
            'Value': 'string'
        },
    ],
    OptionsToRemove=[
        {
            'ResourceName': 'string',
            'Namespace': 'string',
            'OptionName': 'string'
        },
    ]
)
```

Parameters

- **ApplicationName** (*string*) – [REQUIRED]
The name of the application associated with the configuration template to update.

If no application is found with this name, `UpdateConfigurationTemplate` returns an `InvalidParameterValue` error.

- **TemplateName** (*string*) – [REQUIRED]

The name of the configuration template to update.

If no configuration template is found with this name, `UpdateConfigurationTemplate` returns an `InvalidParameterValue` error.

- **Description** (*string*) – A new description for the configuration.
- **OptionSettings** (*list*) – A list of configuration option settings to update with the new specified option value.
 - (*dict*) –

A specification identifying an individual configuration option along with its current value. For a list of possible option values, go to [Option Values](#) in the *AWS Elastic Beanstalk Developer Guide*.

- * **ResourceName** (*string*) –

A unique resource name for a time-based scaling configuration option.

- * **Namespace** (*string*) –

A unique namespace identifying the option's associated AWS resource.

- * **OptionName** (*string*) –

The name of the configuration option.

- * **Value** (*string*) –

The current value for the configuration option.

- **OptionsToRemove** (*list*) – A list of configuration options to remove from the configuration set.

Constraint: You can remove only `UserDefined` configuration options.

– (*dict*) –

A specification identifying an individual configuration option.

- * **ResourceName** (*string*) –

A unique resource name for a time-based scaling configuration option.

- * **Namespace** (*string*) –

A unique namespace identifying the option's associated AWS resource.

- * **OptionName** (*string*) –

The name of the configuration option.

Return type dict

Returns

Response Syntax

```
{
    'SolutionStackName': 'string',
    'ApplicationName': 'string',
    'TemplateName': 'string',
    'Description': 'string',
    'EnvironmentName': 'string',
```

```

'DeploymentStatus': 'deployed'|'pending'|'failed',
'DateCreated': datetime(2015, 1, 1),
'DateUpdated': datetime(2015, 1, 1),
'OptionSettings': [
    {
        'ResourceName': 'string',
        'Namespace': 'string',
        'OptionName': 'string',
        'Value': 'string'
    },
]
}

```

Response Structure

- (*dict*) –

Describes the settings for a configuration set.

- **SolutionStackName** (*string*) –

The name of the solution stack this configuration set uses.

- **ApplicationName** (*string*) –

The name of the application associated with this configuration set.

- **TemplateName** (*string*) –

If not `null`, the name of the configuration template for this configuration set.

- **Description** (*string*) –

Describes this configuration set.

- **EnvironmentName** (*string*) –

If not `null`, the name of the environment for this configuration set.

- **DeploymentStatus** (*string*) –

If this configuration set is associated with an environment, the `DeploymentStatus` parameter indicates the deployment status of this configuration set:

`null`: This configuration is not associated with a running environment.

`pending`: This is a draft configuration that is not deployed to the associated environment but is in the process of deploying.

`deployed`: This is the configuration that is currently deployed to the associated running environment.

`failed`: This is a draft configuration, that failed to successfully deploy.

* `null`: This configuration is not associated with a running environment.

* `pending`: This is a draft configuration that is not deployed to the associated environment but is in the process of deploying.

* `deployed`: This is the configuration that is currently deployed to the associated running environment.

* `failed`: This is a draft configuration that failed to successfully deploy.

- **DateCreated** (*datetime*) –

The date (in UTC time) when this configuration set was created.

- **DateUpdated** (*datetime*) –

The date (in UTC time) when this configuration set was last modified.

- **OptionSettings** (*list*) –

A list of the configuration options and their values in this configuration set.

* (*dict*) –

A specification identifying an individual configuration option along with its current value. For a list of possible option values, go to [Option Values](#) in the *AWS Elastic Beanstalk Developer Guide*.

- **ResourceName** (*string*) –

A unique resource name for a time-based scaling configuration option.

- **Namespace** (*string*) –

A unique namespace identifying the option's associated AWS resource.

- **OptionName** (*string*) –

The name of the configuration option.

- **Value** (*string*) –

The current value for the configuration option.

update_environment (***kwargs*)

Updates the environment description, deploys a new application version, updates the configuration settings to an entirely new configuration template, or updates select configuration option values in the running environment.

Attempting to update both the release and configuration is not allowed and AWS Elastic Beanstalk returns an `InvalidParameterCombination` error.

When updating the configuration settings to a new template or individual settings, a draft configuration is created and `DescribeConfigurationSettings` for this environment returns two setting descriptions with different `DeploymentStatus` values.

Request Syntax

```
response = client.update_environment(
    EnvironmentId='string',
    EnvironmentName='string',
    Description='string',
    Tier={
        'Name': 'string',
        'Type': 'string',
        'Version': 'string'
    },
    VersionLabel='string',
    TemplateName='string',
    SolutionStackName='string',
    OptionSettings=[
        {
            'ResourceName': 'string',
            'Namespace': 'string',
            'OptionName': 'string',
```

```

        'Value': 'string'
    },
],
OptionsToRemove=[
    {
        'ResourceName': 'string',
        'Namespace': 'string',
        'OptionName': 'string'
    },
]
)

```

Parameters

- **EnvironmentId** (*string*) – The ID of the environment to update.
 If no environment with this ID exists, AWS Elastic Beanstalk returns an `InvalidParameterValue` error.
 Condition: You must specify either this or an `EnvironmentName`, or both. If you do not specify either, AWS Elastic Beanstalk returns `MissingRequiredParameter` error.
- **EnvironmentName** (*string*) – The name of the environment to update. If no environment with this name exists, AWS Elastic Beanstalk returns an `InvalidParameterValue` error.
 Condition: You must specify either this or an `EnvironmentId`, or both. If you do not specify either, AWS Elastic Beanstalk returns `MissingRequiredParameter` error.
- **Description** (*string*) – If this parameter is specified, AWS Elastic Beanstalk updates the description of this environment.
- **Tier** (*dict*) – This specifies the tier to use to update the environment.
 Condition: At this time, if you change the tier version, name, or type, AWS Elastic Beanstalk returns `InvalidParameterValue` error.
 - **Name** (*string*) –
 The name of this environment tier.
 - **Type** (*string*) –
 The type of this environment tier.
 - **Version** (*string*) –
 The version of this environment tier.
- **VersionLabel** (*string*) – If this parameter is specified, AWS Elastic Beanstalk deploys the named application version to the environment. If no such application version is found, returns an `InvalidParameterValue` error.
- **TemplateName** (*string*) – If this parameter is specified, AWS Elastic Beanstalk deploys this configuration template to the environment. If no such configuration template is found, AWS Elastic Beanstalk returns an `InvalidParameterValue` error.
- **SolutionStackName** (*string*) – This specifies the platform version that the environment will run after the environment is updated.
- **OptionSettings** (*list*) – If specified, AWS Elastic Beanstalk updates the configuration set associated with the running environment and sets the specified configuration options to the requested value.
 - (*dict*) –
 A specification identifying an individual configuration option along with its current value. For a list of possible option values, go to [Option Values](#)

in the *AWS Elastic Beanstalk Developer Guide* .

- * **ResourceName** (*string*) –

A unique resource name for a time-based scaling configuration option.

- * **Namespace** (*string*) –

A unique namespace identifying the option's associated AWS resource.

- * **OptionName** (*string*) –

The name of the configuration option.

- * **Value** (*string*) –

The current value for the configuration option.

- **OptionsToRemove** (*list*) – A list of custom user-defined configuration options to remove from the configuration set for this environment.

- (*dict*) –

A specification identifying an individual configuration option.

- * **ResourceName** (*string*) –

A unique resource name for a time-based scaling configuration option.

- * **Namespace** (*string*) –

A unique namespace identifying the option's associated AWS resource.

- * **OptionName** (*string*) –

The name of the configuration option.

Return type dict

Returns

Response Syntax

```
{
    'EnvironmentName': 'string',
    'EnvironmentId': 'string',
    'ApplicationName': 'string',
    'VersionLabel': 'string',
    'SolutionStackName': 'string',
    'TemplateName': 'string',
    'Description': 'string',
    'EndpointURL': 'string',
    'CNAME': 'string',
    'DateCreated': datetime(2015, 1, 1),
    'DateUpdated': datetime(2015, 1, 1),
    'Status': 'Launching'|'Updating'|'Ready'|'Terminating'|'Terminated',
    'AbortableOperationInProgress': True|False,
    'Health': 'Green'|'Yellow'|'Red'|'Grey',
    'HealthStatus': 'NoData'|'Unknown'|'Pending'|'Ok'|'Info'|'Warning'|'Degraded'|
    'Resources': {
        'LoadBalancer': {
            'LoadBalancerName': 'string',
            'Domain': 'string',
            'Listeners': [
                {
                    'Protocol': 'string',
                    'Port': 123
```

```

        },
    ],
}
},
'Tier': {
    'Name': 'string',
    'Type': 'string',
    'Version': 'string'
}
}

```

Response Structure

- (*dict*) –

Describes the properties of an environment.

- **EnvironmentName** (*string*) –

The name of this environment.

- **EnvironmentId** (*string*) –

The ID of this environment.

- **ApplicationName** (*string*) –

The name of the application associated with this environment.

- **VersionLabel** (*string*) –

The application version deployed in this environment.

- **SolutionStackName** (*string*) –

The name of the `SolutionStack` deployed with this environment.

- **TemplateName** (*string*) –

The name of the configuration template used to originally launch this environment.

- **Description** (*string*) –

Describes this environment.

- **EndpointURL** (*string*) –

For load-balanced, autoscaling environments, the URL to the LoadBalancer. For single-instance environments, the IP address of the instance.

- **CNAME** (*string*) –

The URL to the CNAME for this environment.

- **DateCreated** (*datetime*) –

The creation date for this environment.

- **DateUpdated** (*datetime*) –

The last modified date for this environment.

- **Status** (*string*) –

The current operational status of the environment:

- * `Launching` : Environment is in the process of initial deployment.
- * `Updating` : Environment is in the process of updating its configuration settings or application version.
- * `Ready` : Environment is available to have an action performed on it, such as update or terminate.
- * `Terminating` : Environment is in the shut-down process.
- * `Terminated` : Environment is not running.

– **AbortableOperationInProgress** (*boolean*) –

Indicates if there is an in-progress environment configuration update or application version deployment that you can cancel.

`true`: There is an update in progress.

`false`: There are no updates currently in progress.

– **Health** (*string*) –

Describes the health status of the environment. AWS Elastic Beanstalk indicates the failure levels for a running environment:

`Red`: Indicates the environment is not working.

`Yellow`: Indicates that something is wrong, the application might not be available, but the instances appear running.

`Green`: Indicates the environment is healthy and fully functional.

* `Red`: Indicates the environment is not responsive. Occurs when three or more consecutive failures occur for an environment.

* `Yellow`: Indicates that something is wrong. Occurs when two consecutive failures occur for an environment.

* `Green`: Indicates the environment is healthy and fully functional.

* `Grey`: Default health for a new environment. The environment is not fully launched and health checks have not started or health checks are suspended during an `UpdateEnvironment` or `RestartEnvironment` request.

Default: `Grey`

– **HealthStatus** (*string*) –

Returns the health status of the application running in your environment. For more information, see [Health Colors and Statuses](#).

– **Resources** (*dict*) –

The description of the AWS resources used by this environment.

* **LoadBalancer** (*dict*) –

Describes the LoadBalancer.

· **LoadBalancerName** (*string*) –

The name of the LoadBalancer.

· **Domain** (*string*) –

The domain name of the LoadBalancer.

· **Listeners** (*list*) –

A list of Listeners used by the LoadBalancer.

· (*dict*) –

Describes the properties of a Listener for the LoadBalancer.

· **Protocol** (*string*) –

The protocol that is used by the Listener.

· **Port** (*integer*) –

The port that is used by the Listener.

– **Tier** (*dict*) –

Describes the current tier of this environment.

- * **Name** (*string*) –
The name of this environment tier.
- * **Type** (*string*) –
The type of this environment tier.
- * **Version** (*string*) –
The version of this environment tier.

validate_configuration_settings (***kwargs*)

Takes a set of configuration settings and either a configuration template or environment, and determines whether those values are valid.

This action returns a list of messages indicating any errors or warnings associated with the selection of option values.

Request Syntax

```
response = client.validate_configuration_settings(
    ApplicationName='string',
    TemplateName='string',
    EnvironmentName='string',
    OptionSettings=[
        {
            'ResourceName': 'string',
            'Namespace': 'string',
            'OptionName': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **ApplicationName** (*string*) – [REQUIRED]
The name of the application that the configuration template or environment belongs to.
- **TemplateName** (*string*) – The name of the configuration template to validate the settings against.
Condition: You cannot specify both this and an environment name.
- **EnvironmentName** (*string*) – The name of the environment to validate the settings against.
Condition: You cannot specify both this and a configuration template name.
- **OptionSettings** (*list*) – [REQUIRED]
A list of the options and desired values to evaluate.
 - (*dict*) –
A specification identifying an individual configuration option along with its current value. For a list of possible option values, go to [Option Values](#) in the *AWS Elastic Beanstalk Developer Guide* .
 - * **ResourceName** (*string*) –
A unique resource name for a time-based scaling configuration option.
 - * **Namespace** (*string*) –
A unique namespace identifying the option's associated AWS resource.

- * **OptionName** (*string*) –

The name of the configuration option.

- * **Value** (*string*) –

The current value for the configuration option.

Return type dict

Returns

Response Syntax

```
{
  'Messages': [
    {
      'Message': 'string',
      'Severity': 'error'|'warning',
      'Namespace': 'string',
      'OptionName': 'string'
    },
  ]
}
```

Response Structure

- (*dict*) –

Provides a list of validation messages.

- **Messages** (*list*) –

A list of ValidationMessage .

- * (*dict*) –

An error or warning for a desired configuration option value.

- **Message** (*string*) –

A message describing the error or warning.

- **Severity** (*string*) –

An indication of the severity of this message:

error: This message indicates that this is not a valid setting for an option.

warning: This message is providing information you should take into account.

- error: This message indicates that this is not a valid setting for an option.

- warning: This message is providing information you should take into account.

- **Namespace** (*string*) –

- **OptionName** (*string*) –

Paginators

The available paginators are:

- `ElasticBeanstalk.Paginator.DescribeEvents`

class ElasticBeanstalk.Paginator.**DescribeEvents**

```
paginator = client.get_paginator('describe_events')
```

paginate (**kwargs)

Creates an iterator that will paginate through responses from `ElasticBeanstalk.Client.describe_events()`.

Request Syntax

```
response_iterator = paginator.paginate(
    ApplicationName='string',
    VersionLabel='string',
    TemplateName='string',
    EnvironmentId='string',
    EnvironmentName='string',
    RequestId='string',
    Severity='TRACE'|'DEBUG'|'INFO'|'WARN'|'ERROR'|'FATAL',
    StartTime=datetime(2015, 1, 1),
    EndTime=datetime(2015, 1, 1),
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **ApplicationName** (*string*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to include only those associated with this application.
- **VersionLabel** (*string*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to those associated with this application version.
- **TemplateName** (*string*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to those that are associated with this environment configuration.
- **EnvironmentId** (*string*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to those associated with this environment.
- **EnvironmentName** (*string*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to those associated with this environment.
- **RequestId** (*string*) – If specified, AWS Elastic Beanstalk restricts the described events to include only those associated with this request ID.
- **Severity** (*string*) – If specified, limits the events returned from this call to include only those with the specified severity or higher.
- **StartTime** (*datetime*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to those that occur on or after this time.
- **EndTime** (*datetime*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to those that occur up to, but not including, the `EndTime`.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Events': [
    {
      'EventDate': datetime(2015, 1, 1),
      'Message': 'string',
      'ApplicationName': 'string',
      'VersionLabel': 'string',
      'TemplateName': 'string',
      'EnvironmentName': 'string',
      'RequestId': 'string',
      'Severity': 'TRACE' | 'DEBUG' | 'INFO' | 'WARN' | 'ERROR' | 'FATAL'
    },
  ],
}
```

Response Structure

- (*dict*) –

Result message wrapping a list of event descriptions.

- **Events** (*list*) –

A list of `EventDescription` .

- * (*dict*) –

Describes an event.

- **EventDate** (*datetime*) –
The date when the event occurred.
- **Message** (*string*) –
The event message.
- **ApplicationName** (*string*) –
The application associated with the event.
- **VersionLabel** (*string*) –
The release label for the application version associated with this event.
- **TemplateName** (*string*) –
The name of the configuration associated with this event.
- **EnvironmentName** (*string*) –
The name of the environment associated with this event.
- **RequestId** (*string*) –
The web service request ID for the activity of this event.
- **Severity** (*string*) –
The severity level of this event.

ElasticTranscoder

Table of Contents

- ElasticTranscoder
 - Client
 - Paginators
 - Waiters

Client

class ElasticTranscoder.Client

A low-level client representing Amazon Elastic Transcoder:

```
import boto3

client = boto3.client('elastictranscoder')
```

These are the available methods:

- `can_paginate()`
- `cancel_job()`
- `create_job()`
- `create_pipeline()`
- `create_preset()`
- `delete_pipeline()`
- `delete_preset()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_waiter()`
- `list_jobs_by_pipeline()`
- `list_jobs_by_status()`
- `list_pipelines()`
- `list_presets()`
- `read_job()`
- `read_pipeline()`
- `read_preset()`
- `test_role()`
- `update_pipeline()`
- `update_pipeline_notifications()`
- `update_pipeline_status()`

`can_paginate(operation_name)`

Check if an operation can be paginated.

Parameters `operation_name` (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

`cancel_job(**kwargs)`

The CancelJob operation cancels an unfinished job.

Note: You can only cancel a job that has a status of `Submitted`. To prevent a pipeline from starting to process a job while you're getting the job identifier, use `UpdatePipelineStatus` to temporarily pause the pipeline.

Request Syntax

```
response = client.cancel_job(  
    Id='string'  
)
```

Parameters `Id` (*string*) – [REQUIRED]

The identifier of the job that you want to cancel.

To get a list of the jobs (including their `jobId`) that have a status of `Submitted`, use the `ListJobsByStatus` API action.

Return type dict

Returns

Response Syntax

```
{}
```

Response Structure

- (*dict*) –

The response body contains a JSON object. If the job is successfully canceled, the value of `Success` is `true`.

`create_job` (***kwargs*)

When you create a job, Elastic Transcoder returns JSON data that includes the values that you specified plus information about the job that is created.

If you have specified more than one output for your jobs (for example, one output for the Kindle Fire and another output for the Apple iPhone 4s), you currently must use the Elastic Transcoder API to list the jobs (as opposed to the AWS Console).

Request Syntax

```
response = client.create_job(  
    PipelineId='string',  
    Input={  
        'Key': 'string',  
        'FrameRate': 'string',  
        'Resolution': 'string',  
        'AspectRatio': 'string',  
        'Interlaced': 'string',  
        'Container': 'string',  
        'Encryption': {  
            'Mode': 'string',  
            'Key': 'string',  
            'KeyMd5': 'string',  
            'InitializationVector': 'string'  
        },  
    },  
    'DetectedProperties': {  
        'Width': 123,  
        'Height': 123,  
        'FrameRate': 'string',  
        'FileSize': 123,
```

```

        'DurationMillis': 123
    }
},
Output={
    'Key': 'string',
    'ThumbnailPattern': 'string',
    'ThumbnailEncryption': {
        'Mode': 'string',
        'Key': 'string',
        'KeyMd5': 'string',
        'InitializationVector': 'string'
    },
    'Rotate': 'string',
    'PresetId': 'string',
    'SegmentDuration': 'string',
    'Watermarks': [
        {
            'PresetWatermarkId': 'string',
            'InputKey': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        }
    ],
    'AlbumArt': {
        'MergePolicy': 'string',
        'Artwork': [
            {
                'InputKey': 'string',
                'MaxWidth': 'string',
                'MaxHeight': 'string',
                'SizingPolicy': 'string',
                'PaddingPolicy': 'string',
                'AlbumArtFormat': 'string',
                'Encryption': {
                    'Mode': 'string',
                    'Key': 'string',
                    'KeyMd5': 'string',
                    'InitializationVector': 'string'
                }
            }
        ],
    },
    'Composition': [
        {
            'TimeSpan': {
                'StartTime': 'string',
                'Duration': 'string'
            }
        }
    ],
    'Captions': {
        'MergePolicy': 'string',
        'CaptionSources': [
            {

```

```
        'Key': 'string',
        'Language': 'string',
        'TimeOffset': 'string',
        'Label': 'string',
        'Encryption': {
            'Mode': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'InitializationVector': 'string'
        }
    },
],
'CaptionFormats': [
    {
        'Format': 'string',
        'Pattern': 'string',
        'Encryption': {
            'Mode': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'InitializationVector': 'string'
        }
    }
],
'Encryption': {
    'Mode': 'string',
    'Key': 'string',
    'KeyMd5': 'string',
    'InitializationVector': 'string'
},
Outputs=[
    {
        'Key': 'string',
        'ThumbnailPattern': 'string',
        'ThumbnailEncryption': {
            'Mode': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'InitializationVector': 'string'
        },
        'Rotate': 'string',
        'PresetId': 'string',
        'SegmentDuration': 'string',
        'Watermarks': [
            {
                'PresetWatermarkId': 'string',
                'InputKey': 'string',
                'Encryption': {
                    'Mode': 'string',
                    'Key': 'string',
                    'KeyMd5': 'string',
                    'InitializationVector': 'string'
                }
            }
        ],
        'AlbumArt': {
```



```

'MergePolicy': 'string',
'Artwork': [
    {
        'InputKey': 'string',
        'MaxWidth': 'string',
        'MaxHeight': 'string',
        'SizingPolicy': 'string',
        'PaddingPolicy': 'string',
        'AlbumArtFormat': 'string',
        'Encryption': {
            'Mode': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'InitializationVector': 'string'
        }
    },
],
},
'Composition': [
    {
        'TimeSpan': {
            'StartTime': 'string',
            'Duration': 'string'
        }
    },
],
},
'Captions': {
    'MergePolicy': 'string',
    'CaptionSources': [
        {
            'Key': 'string',
            'Language': 'string',
            'TimeOffset': 'string',
            'Label': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        }
    ],
},
'CaptionFormats': [
    {
        'Format': 'string',
        'Pattern': 'string',
        'Encryption': {
            'Mode': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'InitializationVector': 'string'
        }
    },
],
},
'Encryption': {
    'Mode': 'string',
    'Key': 'string',

```

```
        'KeyMd5': 'string',
        'InitializationVector': 'string'
    },
    ],
    OutputKeyPrefix='string',
    Playlists=[
        {
            'Name': 'string',
            'Format': 'string',
            'OutputKeys': [
                'string',
            ],
            'HlsContentProtection': {
                'Method': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string',
                'LicenseAcquisitionUrl': 'string',
                'KeyStoragePolicy': 'string'
            },
            'PlayReadyDrm': {
                'Format': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'KeyId': 'string',
                'InitializationVector': 'string',
                'LicenseAcquisitionUrl': 'string'
            }
        },
    ],
    UserMetadata={
        'string': 'string'
    }
)
```

Parameters

- **PipelineId** (*string*) – [REQUIRED]

The Id of the pipeline that you want Elastic Transcoder to use for transcoding. The pipeline determines several settings, including the Amazon S3 bucket from which Elastic Transcoder gets the files to transcode and the bucket into which Elastic Transcoder puts the transcoded files.

- **Input** (*dict*) – [REQUIRED]

A section of the request body that provides information about the file that is being transcoded.

- **Key** (*string*) –

The name of the file to transcode. Elsewhere in the body of the JSON block is the the ID of the pipeline to use for processing the job. The `InputBucket` object in that pipeline tells Elastic Transcoder which Amazon S3 bucket to get the file from.

If the file name includes a prefix, such as `cooking/lasagna.mpg`, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

- **FrameRate** (*string*) –

The frame rate of the input file. If you want Elastic Transcoder to automatically detect the frame rate of the input file, specify `auto`. If you want to specify the frame rate for the input file, enter one of the following values:

`10, 15, 23.97, 24, 25, 29.97, 30, 60`

If you specify a value other than `auto`, Elastic Transcoder disables automatic detection of the frame rate.

– **Resolution** (*string*) –

This value must be `auto`, which causes Elastic Transcoder to automatically detect the resolution of the input file.

– **AspectRatio** (*string*) –

The aspect ratio of the input file. If you want Elastic Transcoder to automatically detect the aspect ratio of the input file, specify `auto`. If you want to specify the aspect ratio for the output file, enter one of the following values:

`1:1, 4:3, 3:2, 16:9`

If you specify a value other than `auto`, Elastic Transcoder disables automatic detection of the aspect ratio.

– **Interlaced** (*string*) –

Whether the input file is interlaced. If you want Elastic Transcoder to automatically detect whether the input file is interlaced, specify `auto`. If you want to specify whether the input file is interlaced, enter one of the following values:

`true, false`

If you specify a value other than `auto`, Elastic Transcoder disables automatic detection of interlacing.

– **Container** (*string*) –

The container type for the input file. If you want Elastic Transcoder to automatically detect the container type of the input file, specify `auto`. If you want to specify the container type for the input file, enter one of the following values:

`3gp, aac, asf, avi, divx, flv, m4a, mkv, mov, mp3, mp4, mpeg, mpeg-ps, mpeg-ts, mxf, ogg, vob, wav, webm`

– **Encryption** (*dict*) –

The encryption settings, if any, that are used for decrypting your input files. If your input file is encrypted, you must specify the mode that Elastic Transcoder will use to decrypt your file.

* **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.

- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

* **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

* **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

* **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

– **DetectedProperties** (*dict*) –

The detected properties of the input file.

* **Width** (*integer*) –

The detected width of the input file, in pixels.

* **Height** (*integer*) –

The detected height of the input file, in pixels.

* **FrameRate** (*string*) –

The detected frame rate of the input file, in frames per second.

* **FileSize** (*integer*) –

The detected file size of the input file, in bytes.

* **DurationMillis** (*integer*) –

The detected duration of the input file, in milliseconds.

- **Output** (*dict*) – The `CreateJobOutput` structure.

- **Key** (*string*) –

The name to assign to the transcoded file. Elastic Transcoder saves the file in the Amazon S3 bucket specified by the `OutputBucket` object in the pipeline that is specified by the pipeline ID. If a file with the specified name already exists in the output bucket, the job fails.

- **ThumbnailPattern** (*string*) –

Whether you want Elastic Transcoder to create thumbnails for your videos and, if so, how you want Elastic Transcoder to name the files.

If you don't want Elastic Transcoder to create thumbnails, specify "".

If you do want Elastic Transcoder to create thumbnails, specify the information that you want to include in the file name for each thumbnail. You can specify the following values in any sequence:

- * **{{count}}** (**Required**) : If you want to create thumbnails, you must include `{count}` in the `ThumbnailPattern` object. Wherever you specify `{count}`, Elastic Transcoder adds a five-digit sequence number (beginning with **00001**) to thumbnail file names. The number indicates where a given thumbnail appears in the sequence of thumbnails for a transcoded file. .. warning::If you specify a literal value and/or `{resolution}` but you omit `{count}`, Elastic Transcoder returns a validation error and does not create the job.
- * **Literal values** (**Optional**) : You can specify literal values anywhere in the `ThumbnailPattern` object. For example, you can include them as a file name prefix or as a delimiter between `{resolution}` and `{count}`.
- * **{{resolution}}** (**Optional**) : If you want Elastic Transcoder to include the resolution in the file name, include `{resolution}` in the `ThumbnailPattern` object.

When creating thumbnails, Elastic Transcoder automatically saves the files in the format (.jpg or .png) that appears in the preset that you specified in the `PresetID` value of `CreateJobOutput`. Elastic Transcoder also appends the applicable file name extension.

- **ThumbnailEncryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your thumbnail.

- * **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation

that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

* **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

* **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

* **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

– **Rotate** (*string*) –

The number of degrees clockwise by which you want Elastic Transcoder to rotate the output relative to the input. Enter one of the following values: `auto` , `0` , `90` , `180` , `270` . The value `auto` generally works only if the file that you're transcoding contains rotation metadata.

– **PresetId** (*string*) –

The `Id` of the preset to use for this job. The preset determines the audio, video, and thumbnail settings that Elastic Transcoder uses for transcoding.

– **SegmentDuration** (*string*) –

Warning: (Outputs in Fragmented MP4 or MPEG-TS format only.)

If you specify a preset in `PresetId` for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `SegmentDuration` is the target maximum duration of each segment in seconds. For `HLSv3` format playlists, each media segment is stored in a separate `.ts` file.

For `HLSv4` and `Smooth` playlists, all media segments for an output are stored in a single file. Each segment is approximately the length of the `SegmentDuration`, though individual segments might be shorter or longer.

The range of valid values is 1 to 60 seconds. If the duration of the video is not evenly divisible by `SegmentDuration`, the duration of the last segment is the remainder of total length/`SegmentDuration`.

Elastic Transcoder creates an output-specific playlist for each output HLS output that you specify in `OutputKeys`. To add an output to the master playlist for this job, include it in the `OutputKeys` of the associated playlist.

– **Watermarks** (*list*) –

Information about the watermarks that you want Elastic Transcoder to add to the video during transcoding. You can specify up to four watermarks for each output. Settings for each watermark must be defined in the preset for the current output.

* (*dict*) –

Watermarks can be in .png or .jpg format. If you want to display a watermark that is not rectangular, use the .png format, which supports transparency.

· **PresetWatermarkId** (*string*) –

The ID of the watermark settings that Elastic Transcoder uses to add watermarks to the video during transcoding. The settings are in the preset specified by `Preset` for the current output. In that preset, the value of `Watermarks Id` tells Elastic Transcoder which settings to use.

· **InputKey** (*string*) –

The name of the .png or .jpg file that you want to use for the watermark. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `Pipeline`; the `Input Bucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, **logos/128x64.png**, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

· **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your watermarks.

· **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the

AWS-KMS key that you want to use to your pipeline.

- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AlbumArt** (*dict*) –

Information about the album art that you want Elastic Transcoder to add to the file during transcoding. You can specify up to twenty album artworks for each output. Settings for each artwork must be defined in the job for the current output.

- * **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder will handle the existence of multiple album artwork files.

- **Replace**: The specified album art will replace any existing album art.
- **Prepend**: The specified album art will be placed in front of any existing album art.
- **Append**: The specified album art will be placed after any existing album art.
- **Fallback**: If the original input file contains artwork, Elastic Transcoder will use that artwork for the output. If the original input does not contain artwork, Elastic Transcoder will use the specified album art file.

* **Artwork** (*list*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20. Valid formats are `.jpg` and `.png`

- (*dict*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20.

To remove artwork or leave the artwork empty, you can either set `Artwork` to `null`, or set the `Merge Policy` to “Replace” and use an empty `Artwork` array.

To pass through existing artwork unchanged, set the `Merge Policy` to “Prepend”, “Append”, or “Fallback”, and use an empty `Artwork` array.

- **InputKey** (*string*) –

The name of the file to be used as album art. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/pie.jpg`, include the prefix in the key. If the file isn’t in the specified bucket, Elastic Transcoder returns an error.

- **MaxWidth** (*string*) –

The maximum width of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

- **MaxHeight** (*string*) –

The maximum height of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.

- **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of the output album art:

- **Fit**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.

- **Fill:** Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output art and then crops it in the dimension (if any) that exceeds the maximum value.
- **Stretch:** Elastic Transcoder stretches the output art to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input art and the output art are different, the output art will be distorted.
- **Keep:** Elastic Transcoder does not scale the output art. If either dimension of the input art exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output art.
- **ShrinkToFit:** Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the art up.
- **ShrinkToFit** Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the art up.
- **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder may add white bars to the top and bottom and/or left and right sides of the output album art to make the total size of the output art match the values that you specified for `MaxWidth` and `MaxHeight`.

- **AlbumArtFormat** (*string*) –

The format of album art, if any. Valid formats are `.jpg` and `.png`.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your artwork.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file,

key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Composition** (*list*) –

You can create an output file that contains an excerpt from the input file. This excerpt, called a clip, can come from the beginning, middle, or end of the file. The Composition object contains settings for the clips that make up an output file. For the current release, you can only specify settings for a single clip per output file. The Composition object cannot be null.

- **dict** –

Settings for one clip in a composition. All jobs in a playlist must have the same clip settings.

- **TimeSpan** (*dict*) –

Settings that determine when a clip begins and how long it lasts.

- **StartTime** (*string*) –

The place in the input file where you want a clip to start. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder starts at the beginning of the input file.

· **Duration** (*string*) –

The duration of the clip. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder creates an output file from StartTime to the end of the file.

If you specify a value longer than the duration of the input file, Elastic Transcoder transcodes the file and returns a warning message.

– **Captions** (*dict*) –

You can configure Elastic Transcoder to transcode captions, or subtitles, from one format to another. All captions must be in UTF-8. Elastic Transcoder supports two types of captions:

- * **Embedded:** Embedded captions are included in the same file as the audio and video. Elastic Transcoder supports only one embedded caption per language, to a maximum of 300 embedded captions per file. Valid input values include: CEA-608 (EIA-608, first non-empty channel only), CEA-708 (EIA-708, first non-empty channel only), and mov-text. Valid outputs include: mov-text. Elastic Transcoder supports a maximum of one embedded format per output.
- * **Sidecar:** Sidecar captions are kept in a separate metadata file from the audio and video data. Sidecar captions require a player that is capable of understanding the relationship between the video file and the sidecar file. Elastic Transcoder supports only one sidecar caption per language, to a maximum of 20 sidecar captions per file. Valid input values include: dfxp (first div element only), ebu-tt, scc, smpt, srt, ttml (first div element only), and webvtt. Valid outputs include: dfxp (first div element only), scc, srt, and webvtt.

If you want ttml or smpte-tt compatible captions, specify dfxp as your output format.

Elastic Transcoder does not support OCR (Optical Character Recognition), does not accept pictures as a valid input for captions, and is not available for audio-only transcoding. Elastic Transcoder does not preserve text formatting (for example, italics) during the transcoding process.

To remove captions or leave the captions empty, set `Captions` to null. To pass through existing captions unchanged, set the `MergePolicy` to `MergeRetain`, and pass in a null `CaptionSources` array.

For more information on embedded files, see the Subtitles Wikipedia page.

For more information on sidecar files, see the Extensible Metadata Platform and Sidecar file Wikipedia pages.

* **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder handles the existence of multiple captions.

- **MergeOverride:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the sidecar captions and ignores the embedded captions for that language.
- **MergeRetain:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the embedded captions and ignores the sidecar captions for that language. If `CaptionSources` is empty, Elastic Transcoder omits all sidecar captions from the output files.
- **Override:** Elastic Transcoder transcodes only the sidecar captions that you specify in `CaptionSources`.

`MergePolicy` cannot be null.

* **CaptionSources** (*list*) –

Source files for the input sidecar captions used during the transcoding process. To omit all sidecar captions, leave `CaptionSources` blank.

· (*dict*) –

A source file for the input sidecar captions used during the transcoding process.

· **Key** (*string*) –

The name of the sidecar caption file that you want Elastic Transcoder to include in the output file.

· **Language** (*string*) –

A string that specifies the language of the caption. Specify this as one of:

- 2-character ISO 639-1 code
- 3-character ISO 639-2 code

For more information on ISO language codes and language names, see the List of ISO 639-1 codes.

· **TimeOffset** (*string*) –

For clip generation or captions that do not start at the same time as the associated video file, the `TimeOffset` tells Elastic Transcoder how much of the video to encode before including captions.

Specify the `TimeOffset` in the form `[+-]SS.sss` or `[+-]HH:mm:SS.ss`.

· **Label** (*string*) –

The label of the caption shown in the player when choosing a language. We recommend that you put the caption language name here, in the language of the captions.

· **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption sources.

· **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or en-

crypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

* **CaptionFormats** (*list*) –

The array of file formats for the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- *(dict)* –

The file format of the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- **Format** (*string*) –

The format you specify determines whether Elastic Transcoder generates an embedded or sidecar caption for this output.

- **Valid Embedded Caption Formats:** * **for FLAC** : None

- **For MP3** : None

- **For MP4** : mov-text

- **For MPEG-TS** : None

- **For ogg** : None

- **For webm** : None

- **Valid Sidecar Caption Formats:** Elastic Transcoder supports dfxp (first div element only), scc, srt, and webvtt. If you want ttml or smpte-tt compatible captions, specify dfxp as your output format. * **For FMP4** : dfxp

- **Non-FMP4 outputs** : All sidecar types

fmp4 captions have an extension of .ismt

- **Pattern** (*string*) –

The prefix for caption filenames, in the form *description* - {language} , where:

- *description* is a description of the video.

- {language} is a literal value that Elastic Transcoder replaces with the two- or three-letter code for the language of the caption in the output file names.

If you don't include {language} in the file name pattern, Elastic Transcoder automatically appends "{language} " to the value that you specify for the description. In addition, Elastic Transcoder automatically appends the count to the end of the segment files.

For example, suppose you're transcoding into srt format. When you enter "Sydney-{language}-sunrise", and the language of the captions is English (en), the name of the first caption file will be Sydney-en-sunrise00000.srt.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption formats.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.

- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify S3-AWS-KMS and you don't want to use the default key, you must add the

AWS-KMS key that you want to use to your pipeline.

- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

– **Encryption** (*dict*) –

You can specify encryption settings for any output files that you want to use for a transcoding job. This includes the output file and any watermarks, thumbnails, album art, or captions that you want to use. You must specify encryption settings for each file individually.

- * **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting

your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

* **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

* **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

* **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Outputs** (*list*) – A section of the request body that provides information about the transcoded (target) files. We recommend that you use the `Outputs` syntax instead of the `Output` syntax.

– (*dict*) –

The `CreateJobOutput` structure.

* **Key** (*string*) –

The name to assign to the transcoded file. Elastic Transcoder saves the file in the Amazon S3 bucket specified by the `OutputBucket` object in the pipeline that is specified by the pipeline ID. If a file with the specified name already exists in the output bucket, the job fails.

* **ThumbnailPattern** (*string*) –

Whether you want Elastic Transcoder to create thumbnails for your videos and, if so, how you want Elastic Transcoder to name the files.

If you don't want Elastic Transcoder to create thumbnails, specify "".

If you do want Elastic Transcoder to create thumbnails, specify the information that you want to include in the file name for each thumbnail. You can specify the following values in any sequence:

- **“{count}” (Required)** : If you want to create thumbnails, you must include `{count}` in the `ThumbnailPattern` object. Wherever you specify `{count}`, Elastic Transcoder adds a five-digit sequence number (beginning with **00001**) to thumbnail file names. The number indicates where a given thumbnail appears in the sequence of thumbnails for a transcoded file. .. warning::If you specify a literal value and/or `{resolution}` but you omit `{count}`, Elastic Transcoder returns a validation error and does not create the job.
- **Literal values (Optional)** : You can specify literal values anywhere in the `ThumbnailPattern` object. For example, you can include them as a file name prefix or as a delimiter between `{resolution}` and `{count}`.
- **“{resolution}” (Optional)** : If you want Elastic Transcoder to include the resolution in the file name, include `{resolution}` in the `ThumbnailPattern` object.

When creating thumbnails, Elastic Transcoder automatically saves the files in the format (.jpg or .png) that appears in the preset that you specified in the `PresetID` value of `CreateJobOutput`. Elastic Transcoder also appends the applicable file name extension.

* **ThumbnailEncryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your thumbnail.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3**: Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS**: Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.

- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- * **Rotate** (*string*) –

The number of degrees clockwise by which you want Elastic Transcoder to rotate the output relative to the input. Enter one of the following values: `auto` , 0 , 90 , 180 , 270 . The value `auto` generally works only if the file that you're transcoding contains rotation metadata.

- * **PresetId** (*string*) –

The `Id` of the preset to use for this job. The preset determines the audio, video, and thumbnail settings that Elastic Transcoder uses for

transcoding.

* **SegmentDuration** (*string*) –

Warning: (Outputs in Fragmented MP4 or MPEG-TS format only.

If you specify a preset in `PresetId` for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `SegmentDuration` is the target maximum duration of each segment in seconds. For `HLSv3` format playlists, each media segment is stored in a separate `.ts` file. For `HLSv4` and `Smooth` playlists, all media segments for an output are stored in a single file. Each segment is approximately the length of the `SegmentDuration`, though individual segments might be shorter or longer.

The range of valid values is 1 to 60 seconds. If the duration of the video is not evenly divisible by `SegmentDuration`, the duration of the last segment is the remainder of total length/`SegmentDuration`.

Elastic Transcoder creates an output-specific playlist for each output `HLS` output that you specify in `OutputKeys`. To add an output to the master playlist for this job, include it in the `OutputKeys` of the associated playlist.

* **Watermarks** (*list*) –

Information about the watermarks that you want Elastic Transcoder to add to the video during transcoding. You can specify up to four watermarks for each output. Settings for each watermark must be defined in the preset for the current output.

· (*dict*) –

Watermarks can be in `.png` or `.jpg` format. If you want to display a watermark that is not rectangular, use the `.png` format, which supports transparency.

· **PresetWatermarkId** (*string*) –

The ID of the watermark settings that Elastic Transcoder uses to add watermarks to the video during transcoding. The settings are in the preset specified by `Preset` for the current output. In that preset, the value of `Watermarks Id` tells Elastic Transcoder which settings to use.

· **InputKey** (*string*) –

The name of the `.png` or `.jpg` file that you want to use for the watermark. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `Pipeline`; the `Input Bucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `logos/128x64.png`, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

· **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your watermarks.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector

must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

* **AlbumArt** (*dict*) –

Information about the album art that you want Elastic Transcoder to add to the file during transcoding. You can specify up to twenty album artworks for each output. Settings for each artwork must be defined in the job for the current output.

· **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder will handle the existence of multiple album artwork files.

- **Replace**: The specified album art will replace any existing album art.
- **Prepend**: The specified album art will be placed in front of any existing album art.
- **Append**: The specified album art will be placed after any existing album art.
- **Fallback**: If the original input file contains artwork, Elastic Transcoder will use that artwork for the output. If the original input does not contain artwork, Elastic Transcoder will use the specified album art file.

· **Artwork** (*list*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20. Valid formats are .jpg and .png

· (*dict*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20.

To remove artwork or leave the artwork empty, you can either set `Artwork` to null, or set the `Merge Policy` to “Replace” and use an empty `Artwork` array.

To pass through existing artwork unchanged, set the `Merge Policy` to “Prepend”, “Append”, or “Fallback”, and use an empty `Artwork` array.

· **InputKey** (*string*) –

The name of the file to be used as album art. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/pie.jpg`, include the prefix in the key. If the file isn’t in the specified bucket, Elastic Transcoder returns an error.

· **MaxWidth** (*string*) –

The maximum width of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

· **MaxHeight** (*string*) –

The maximum height of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.

- **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of the output album art:

- **Fit**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Fill**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output art and then crops it in the dimension (if any) that exceeds the maximum value.
- **Stretch**: Elastic Transcoder stretches the output art to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input art and the output art are different, the output art will be distorted.
- **Keep**: Elastic Transcoder does not scale the output art. If either dimension of the input art exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output art.
- **ShrinkToFit**: Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the art up.
- **ShrinkToFill**: Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the art up.

- **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder may add white bars to the top and bottom and/or left and right sides of the output album art to make the total size of the output art match the values that you specified for `MaxWidth` and `MaxHeight`.

- **AlbumArtFormat** (*string*) –

The format of album art, if any. Valid formats are `.jpg` and `.png`.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your artwork.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3**: Amazon S3 creates and manages the keys used for en-

crypting your files.

- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- * **Composition** (*list*) –

You can create an output file that contains an excerpt from the input file. This excerpt, called a clip, can come from the beginning, middle, or end of the file. The Composition object contains settings

for the clips that make up an output file. For the current release, you can only specify settings for a single clip per output file. The Composition object cannot be null.

- *(dict)* –

Settings for one clip in a composition. All jobs in a playlist must have the same clip settings.

- **TimeSpan** (*dict*) –

Settings that determine when a clip begins and how long it lasts.

- **StartTime** (*string*) –

The place in the input file where you want a clip to start. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder starts at the beginning of the input file.

- **Duration** (*string*) –

The duration of the clip. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder creates an output file from StartTime to the end of the file.

If you specify a value longer than the duration of the input file, Elastic Transcoder transcodes the file and returns a warning message.

- * **Captions** (*dict*) –

You can configure Elastic Transcoder to transcode captions, or subtitles, from one format to another. All captions must be in UTF-8. Elastic Transcoder supports two types of captions:

- **Embedded:** Embedded captions are included in the same file as the audio and video. Elastic Transcoder supports only one embedded caption per language, to a maximum of 300 embedded captions per file. Valid input values include: CEA-608 (EIA-608 , first non-empty channel only), CEA-708 (EIA-708 , first non-empty channel only), and mov-text. Valid outputs include: mov-text. Elastic Transcoder supports a maximum of one embedded format per output.
- **Sidecar:** Sidecar captions are kept in a separate metadata file from the audio and video data. Sidecar captions require a player that is capable of understanding the relationship between the video file and the sidecar file. Elastic Transcoder supports only one sidecar caption per language, to a maximum of 20 sidecar captions per file. Valid input values include: dfxp (first div element only), ebu-tt , scc , smpt , srt , ttml (first div element only), and webvtt. Valid outputs include: dfxp (first div element only), scc , srt , and webvtt .

If you want ttml or smpte-tt compatible captions, specify dfxp as your output format.

Elastic Transcoder does not support OCR (Optical Character Recog-

niton), does not accept pictures as a valid input for captions, and is not available for audio-only transcoding. Elastic Transcoder does not preserve text formatting (for example, italics) during the transcoding process.

To remove captions or leave the captions empty, set `Captions` to null. To pass through existing captions unchanged, set the `MergePolicy` to `MergeRetain`, and pass in a null `CaptionSources` array.

For more information on embedded files, see the Subtitles Wikipedia page.

For more information on sidcar files, see the Extensible Metadata Platform and Sidcar file Wikipedia pages.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder handles the existence of multiple captions.

- **MergeOverride:** Elastic Transcoder transcodes both embedded and sidcar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidcar file, Elastic Transcoder uses the sidcar captions and ignores the embedded captions for that language.

- **MergeRetain:** Elastic Transcoder transcodes both embedded and sidcar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidcar file, Elastic Transcoder uses the embedded captions and ignores the sidcar captions for that language. If `CaptionSources` is empty, Elastic Transcoder omits all sidcar captions from the output files.

- **Override:** Elastic Transcoder transcodes only the sidcar captions that you specify in `CaptionSources`. `MergePolicy` cannot be null.

- **CaptionSources** (*list*) –

Source files for the input sidcar captions used during the transcoding process. To omit all sidcar captions, leave `CaptionSources` blank.

- (*dict*) –

A source file for the input sidcar captions used during the transcoding process.

- **Key** (*string*) –

The name of the sidcar caption file that you want Elastic Transcoder to include in the output file.

- **Language** (*string*) –

A string that specifies the language of the caption. Specify this as one of:

- 2-character ISO 639-1 code

- 3-character ISO 639-2 code

For more information on ISO language codes and language names, see the List of ISO 639-1 codes.

- **TimeOffset** (*string*) –

For clip generation or captions that do not start at the same

time as the associated video file, the `TimeOffset` tells Elastic Transcoder how much of the video to encode before including captions.

Specify the `TimeOffset` in the form `[+]SS.sss or [+]HH:mm:SS.ss.`

- **Label** (*string*) –

The label of the caption shown in the player when choosing a language. We recommend that you put the caption language name here, in the language of the captions.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption sources.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key

Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **CaptionFormats** (*list*) –

The array of file formats for the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- (*dict*) –

The file format of the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- **Format** (*string*) –

The format you specify determines whether Elastic Transcoder generates an embedded or sidecar caption for this output.

- **Valid Embedded Caption Formats:** * for **FLAC** : None

- **For MP3** : None

- **For MP4** : mov-text

- **For MPEG-TS** : None

- **For ogg** : None

- **For webm** : None

- **Valid Sidecar Caption Formats:** Elastic Transcoder supports dfxp (first div element only), scc, srt, and webvtt. If you want ttml or smpte-tt compatible captions, specify dfxp as your output format. * **For FMP4** : dfxp

- **Non-FMP4 outputs** : All sidecar types

fm4 captions have an extension of .ismt

- **Pattern** (*string*) –

The prefix for caption filenames, in the form *description* - {language} , where:

- *description* is a description of the video.

- {language} is a literal value that Elastic Transcoder replaces with the two- or three-letter code for the language of the caption in the output file names.

If you don't include {language} in the file name pattern, Elastic Transcoder automatically appends "{language} " to the value that you specify for the description. In addition, Elastic Transcoder automatically appends the count to the end of the segment files.

For example, suppose you're transcoding into srt format.

When you enter “Sydney-{language}-sunrise”, and the language of the captions is English (en), the name of the first caption file will be Sydney-en-sunrise00000.srt.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption formats.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify S3-AWS-KMS and you don’t want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won’t be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly

16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- * **Encryption** (*dict*) –

You can specify encryption settings for any output files that you want to use for a transcoding job. This includes the output file and any watermarks, thumbnails, album art, or captions that you want to use. You must specify encryption settings for each file individually.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3**: Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS**: Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7**: A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR**: AES Counter Mode.
- **AES-GCM**: AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key

Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **OutputKeyPrefix** (*string*) – The value, if any, that you want Elastic Transcoder to prepend to the names of all files that this job creates, including output files, thumbnails, and playlists.
- **Playlists** (*list*) – If you specify a preset in `PresetId` for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), Playlists contains information about the master playlists that you want Elastic Transcoder to create.

The maximum number of master playlists in a job is 30.

– (*dict*) –

Information about the master playlist.

- * **Name** (*string*) –

The name that you want Elastic Transcoder to assign to the master playlist, for example, `nyc-vacation.m3u8`. If the name includes a `/` character, the section of the name before the last `/` must be identical for all `Name` objects. If you create more than one master playlist, the values of all `Name` objects must be unique.

Note: Elastic Transcoder automatically appends the relevant file extension to the file name (`.m3u8` for `HLSv3` and `HLSv4` playlists, and `.ism` and `.ismc` for `Smooth` playlists). If you include a file extension in `Name`, the file name will have two extensions.

- * **Format** (*string*) –

The format of the output playlist. Valid formats include `HLSv3`, `HLSv4`, and `Smooth`.

- * **OutputKeys** (*list*) –

For each output in this job that you want to include in a master playlist, the value of the `Outputs:Key` object.

- If your output is not `HLS` or does not have a segment duration set, the name of the output file is a concatenation of `OutputKeyPrefix` and `Outputs:Key:OutputKeyPrefix“Outputs:Key“`
- If your output is `HLSv3` and has a segment duration set, or is not included in a playlist, Elastic Transcoder creates an output playlist file with a file extension of `.m3u8`, and a series of `.ts` files that include a five-digit sequential counter beginning with 00000: `OutputKeyPrefix“Outputs:Key“ .m3u8 OutputKeyPrefix“Outputs:Key“ 00000.ts`

- If your output is `HLSv4` , has a segment duration set, and is included in an `HLSv4` playlist, Elastic Transcoder creates an output playlist file with a file extension of `_v4.m3u8` . If the output is video, Elastic Transcoder also creates an output file with an extension of `_iframe.m3u8` : `OutputKeyPrefix“Outputs:Key“_v4.m3u8` `OutputKeyPrefix“Outputs:Key“_iframe.m3u8` `OutputKeyPrefix“Outputs:Key“ .ts`

Elastic Transcoder automatically appends the relevant file extension to the file name. If you include a file extension in Output Key, the file name will have two extensions.

If you include more than one output in a playlist, any segment duration settings, clip settings, or caption settings must be the same for all outputs in the playlist. For Smooth playlists, the `Audio:Profile` , `Video:Profile` , and `Video:FrameRate` to `Video:KeyframesMaxDist` ratio must be the same for all outputs.

- *(string)* –

* **HlsContentProtection** (*dict*) –

The HLS content protection settings, if any, that you want Elastic Transcoder to apply to the output files associated with this playlist.

- **Method** (*string*) –

The content protection method for your output. The only valid value is: `aes-128` .

This value will be written into the method attribute of the `EXT-X-KEY` metadata tag in the output playlist.

- **Key** (*string*) –

If you want Elastic Transcoder to generate a key for you, leave this field blank.

If you choose to supply your own key, you must encrypt the key by using AWS KMS. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

`128` , `192` , or `256` .

- **KeyMd5** (*string*) –

If Elastic Transcoder is generating your key for you, you must leave this field blank.

The MD5 digest of the key that you want Elastic Transcoder to use to encrypt your output file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

- **InitializationVector** (*string*) –

If Elastic Transcoder is generating your key for you, you must leave this field blank.

The series of random bits created by a random bit generator, unique for every encryption operation, that you want Elastic

Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

- **LicenseAcquisitionUrl** (*string*) –

The location of the license key required to decrypt your HLS playlist. The URL must be an absolute path, and is referenced in the URI attribute of the EXT-X-KEY metadata tag in the playlist file.

- **KeyStoragePolicy** (*string*) –

Specify whether you want Elastic Transcoder to write your HLS license key to an Amazon S3 bucket. If you choose `WithVariantPlaylists`, `LicenseAcquisitionUrl` must be left blank and Elastic Transcoder writes your data key into the same bucket as the associated playlist.

- * **PlayReadyDrm** (*dict*) –

The DRM settings, if any, that you want Elastic Transcoder to apply to the output files associated with this playlist.

- **Format** (*string*) –

The type of DRM, if any, that you want Elastic Transcoder to apply to the output files associated with this playlist.

- **Key** (*string*) –

The DRM key for your file, provided by your DRM license provider. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using AWS KMS.

- **KeyMd5** (*string*) –

The MD5 digest of the key used for DRM on your file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

- **KeyId** (*string*) –

The ID for your DRM key, so that your DRM license provider knows which key to provide.

The key ID must be provided in big endian, and Elastic Transcoder will convert it to little endian before inserting it into the PlayReady DRM headers. If you are unsure whether your license server provides your key ID in big or little endian, check with your DRM provider.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your files. The initialization vector must be base64-encoded, and it must be exactly 8 bytes long before being base64-encoded. If no initialization vector is provided, Elastic Transcoder generates one for you.

- **LicenseAcquisitionUrl** (*string*) –

The location of the license key required to play DRM content. The URL must be an absolute path, and is referenced by the PlayReady header. The PlayReady header is referenced in the protection header of the client manifest for Smooth Streaming outputs, and in the EXT-X-DXDRM and EXT-XDXDRMINFO metadata tags for HLS playlist outputs. An example URL looks like this: <https://www.example.com/exampleKey/>

- **UserMetadata** (*dict*) – User-defined metadata that you want to associate with an Elastic Transcoder job. You specify metadata in *key/value* pairs, and you can add up to 10 *key/value* pairs per job. Elastic Transcoder does not guarantee that *key/value* pairs will be returned in the same order in which you specify them.

- (*string*) –

- * (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'Job': {
    'Id': 'string',
    'Arn': 'string',
    'PipelineId': 'string',
    'Input': {
      'Key': 'string',
      'FrameRate': 'string',
      'Resolution': 'string',
      'AspectRatio': 'string',
      'Interlaced': 'string',
      'Container': 'string',
      'Encryption': {
        'Mode': 'string',
        'Key': 'string',
        'KeyMd5': 'string',
        'InitializationVector': 'string'
      },
      'DetectedProperties': {
        'Width': 123,
        'Height': 123,
        'FrameRate': 'string',
        'FileSize': 123,
        'DurationMillis': 123
      }
    },
    'Output': {
      'Id': 'string',
      'Key': 'string',
      'ThumbnailPattern': 'string',
      'ThumbnailEncryption': {
        'Mode': 'string',
        'Key': 'string',
        'KeyMd5': 'string',
        'InitializationVector': 'string'
      }
    }
  },
}
```

```

'Rotate': 'string',
'PresetId': 'string',
'SegmentDuration': 'string',
'Status': 'string',
'StatusDetail': 'string',
'Duration': 123,
'Width': 123,
'Height': 123,
'FrameRate': 'string',
'FileSize': 123,
'DurationMillis': 123,
'Watermarks': [
    {
        'PresetWatermarkId': 'string',
        'InputKey': 'string',
        'Encryption': {
            'Mode': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'InitializationVector': 'string'
        }
    },
],
'AlbumArt': {
    'MergePolicy': 'string',
    'Artwork': [
        {
            'InputKey': 'string',
            'MaxWidth': 'string',
            'MaxHeight': 'string',
            'SizingPolicy': 'string',
            'PaddingPolicy': 'string',
            'AlbumArtFormat': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        },
    ],
},
'Composition': [
    {
        'TimeSpan': {
            'StartTime': 'string',
            'Duration': 'string'
        }
    },
],
'Captions': {
    'MergePolicy': 'string',
    'CaptionSources': [
        {
            'Key': 'string',
            'Language': 'string',
            'TimeOffset': 'string',
            'Label': 'string',

```

```
        'Encryption': {
            'Mode': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'InitializationVector': 'string'
        },
    ],
    'CaptionFormats': [
        {
            'Format': 'string',
            'Pattern': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        },
    ],
},
'Encryption': {
    'Mode': 'string',
    'Key': 'string',
    'KeyMd5': 'string',
    'InitializationVector': 'string'
},
'AppliedColorSpaceConversion': 'string'
},
'Outputs': [
    {
        'Id': 'string',
        'Key': 'string',
        'ThumbnailPattern': 'string',
        'ThumbnailEncryption': {
            'Mode': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'InitializationVector': 'string'
        },
        'Rotate': 'string',
        'PresetId': 'string',
        'SegmentDuration': 'string',
        'Status': 'string',
        'StatusDetail': 'string',
        'Duration': 123,
        'Width': 123,
        'Height': 123,
        'FrameRate': 'string',
        'FileSize': 123,
        'DurationMillis': 123,
        'Watermarks': [
            {
                'PresetWatermarkId': 'string',
                'InputKey': 'string',
                'Encryption': {
                    'Mode': 'string',
                    'Key': 'string',
```

```

        'KeyMd5': 'string',
        'InitializationVector': 'string'
    },
    ],
    'AlbumArt': {
        'MergePolicy': 'string',
        'Artwork': [
            {
                'InputKey': 'string',
                'MaxWidth': 'string',
                'MaxHeight': 'string',
                'SizingPolicy': 'string',
                'PaddingPolicy': 'string',
                'AlbumArtFormat': 'string',
                'Encryption': {
                    'Mode': 'string',
                    'Key': 'string',
                    'KeyMd5': 'string',
                    'InitializationVector': 'string'
                }
            },
        ]
    },
    'Composition': [
        {
            'TimeSpan': {
                'StartTime': 'string',
                'Duration': 'string'
            }
        },
    ],
    'Captions': {
        'MergePolicy': 'string',
        'CaptionSources': [
            {
                'Key': 'string',
                'Language': 'string',
                'TimeOffset': 'string',
                'Label': 'string',
                'Encryption': {
                    'Mode': 'string',
                    'Key': 'string',
                    'KeyMd5': 'string',
                    'InitializationVector': 'string'
                }
            },
        ],
        'CaptionFormats': [
            {
                'Format': 'string',
                'Pattern': 'string',
                'Encryption': {
                    'Mode': 'string',
                    'Key': 'string',
                    'KeyMd5': 'string',
                    'InitializationVector': 'string'
                }
            }
        ]
    }
}

```

```
        },
    ],
    'Encryption': {
        'Mode': 'string',
        'Key': 'string',
        'KeyMd5': 'string',
        'InitializationVector': 'string'
    },
    'AppliedColorSpaceConversion': 'string'
},
],
'OutputKeyPrefix': 'string',
'Playlists': [
    {
        'Name': 'string',
        'Format': 'string',
        'OutputKeys': [
            'string',
        ],
        'HlsContentProtection': {
            'Method': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'InitializationVector': 'string',
            'LicenseAcquisitionUrl': 'string',
            'KeyStoragePolicy': 'string'
        },
        'PlayReadyDrm': {
            'Format': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'KeyId': 'string',
            'InitializationVector': 'string',
            'LicenseAcquisitionUrl': 'string'
        },
        'Status': 'string',
        'StatusDetail': 'string'
    },
],
'Status': 'string',
'UserMetadata': {
    'string': 'string'
},
'Timing': {
    'SubmitTimeMillis': 123,
    'StartTimeMillis': 123,
    'FinishTimeMillis': 123
}
}
```

Response Structure

- (*dict*) –

The CreateJobResponse structure.

- **Job** (*dict*) –

A section of the response body that provides information about the job that

is created.

- * **Id** (*string*) –

The identifier that Elastic Transcoder assigned to the job. You use this value to get settings for the job or to delete the job.

- * **Arn** (*string*) –

The Amazon Resource Name (ARN) for the job.

- * **PipelineId** (*string*) –

The `Id` of the pipeline that you want Elastic Transcoder to use for transcoding. The pipeline determines several settings, including the Amazon S3 bucket from which Elastic Transcoder gets the files to transcode and the bucket into which Elastic Transcoder puts the transcoded files.

- * **Input** (*dict*) –

A section of the request or response body that provides information about the file that is being transcoded.

- **Key** (*string*) –

The name of the file to transcode. Elsewhere in the body of the JSON block is the the ID of the pipeline to use for processing the job. The `InputBucket` object in that pipeline tells Elastic Transcoder which Amazon S3 bucket to get the file from.

If the file name includes a prefix, such as `cooking/lasagna.mpg`, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

- **FrameRate** (*string*) –

The frame rate of the input file. If you want Elastic Transcoder to automatically detect the frame rate of the input file, specify `auto`. If you want to specify the frame rate for the input file, enter one of the following values:

10, 15, 23.97, 24, 25, 29.97, 30, 60

If you specify a value other than `auto`, Elastic Transcoder disables automatic detection of the frame rate.

- **Resolution** (*string*) –

This value must be `auto`, which causes Elastic Transcoder to automatically detect the resolution of the input file.

- **AspectRatio** (*string*) –

The aspect ratio of the input file. If you want Elastic Transcoder to automatically detect the aspect ratio of the input file, specify `auto`. If you want to specify the aspect ratio for the output file, enter one of the following values:

1:1, 4:3, 3:2, 16:9

If you specify a value other than `auto`, Elastic Transcoder disables automatic detection of the aspect ratio.

- **Interlaced** (*string*) –

Whether the input file is interlaced. If you want Elastic Transcoder to automatically detect whether the input file is interlaced, specify `auto`. If you want to specify whether the input file is interlaced, enter one of the following values:

`true, false`

If you specify a value other than `auto`, Elastic Transcoder disables automatic detection of interlacing.

- **Container** (*string*) –

The container type for the input file. If you want Elastic Transcoder to automatically detect the container type of the input file, specify `auto`. If you want to specify the container type for the input file, enter one of the following values:

`3gp, aac, asf, avi, divx, flv, m4a, mkv, mov, mp3, mp4, mpeg, mpeg-ps, mpeg-ts, mxf, ogg, vob, wav, webm`

- **Encryption** (*dict*) –

The encryption settings, if any, that are used for decrypting your input files. If your input file is encrypted, you must specify the mode that Elastic Transcoder will use to decrypt your file.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-

encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **DetectedProperties** (*dict*) –

The detected properties of the input file.

- **Width** (*integer*) –

The detected width of the input file, in pixels.

- **Height** (*integer*) –

The detected height of the input file, in pixels.

- **FrameRate** (*string*) –

The detected frame rate of the input file, in frames per second.

- **FileSize** (*integer*) –

The detected file size of the input file, in bytes.

- **DurationMillis** (*integer*) –

The detected duration of the input file, in milliseconds.

- * **Output** (*dict*) –

If you specified one output for a job, information about that output. If you specified multiple outputs for a job, the Output object lists information about the first output. This duplicates the information that is listed for the first output in the Outputs object.

Warning: Outputs recommended instead.

A section of the request or response body that provides information about the transcoded (target) file.

- **Id** (*string*) –

A sequential counter, starting with 1, that identifies an output among the outputs from the current job. In the Output syntax, this value is always 1.

- **Key** (*string*) –

The name to assign to the transcoded file. Elastic Transcoder saves the file in the Amazon S3 bucket specified by the

OutputBucket object in the pipeline that is specified by the pipeline ID.

- **ThumbnailPattern** (*string*) –

Whether you want Elastic Transcoder to create thumbnails for your videos and, if so, how you want Elastic Transcoder to name the files.

If you don't want Elastic Transcoder to create thumbnails, specify "".

If you do want Elastic Transcoder to create thumbnails, specify the information that you want to include in the file name for each thumbnail. You can specify the following values in any sequence:

- **"{count}" (Required)** : If you want to create thumbnails, you must include {count} in the ThumbnailPattern object. Wherever you specify {count}, Elastic Transcoder adds a five-digit sequence number (beginning with **00001**) to thumbnail file names. The number indicates where a given thumbnail appears in the sequence of thumbnails for a transcoded file. .. warning::If you specify a literal value and/or {resolution} but you omit {count}, Elastic Transcoder returns a validation error and does not create the job.
- **Literal values (Optional)** : You can specify literal values anywhere in the ThumbnailPattern object. For example, you can include them as a file name prefix or as a delimiter between {resolution} and {count}.
- **"{resolution}" (Optional)** : If you want Elastic Transcoder to include the resolution in the file name, include {resolution} in the ThumbnailPattern object. When creating thumbnails, Elastic Transcoder automatically saves the files in the format (.jpg or .png) that appears in the preset that you specified in the PresetID value of CreateJobOutput. Elastic Transcoder also appends the applicable file name extension.
- **ThumbnailEncryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your thumbnail.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3**: Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS**: Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify S3-AWS-KMS and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7**: A padded cipher-block mode of operation originally used for HLS files.

- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Rotate** (*string*) –

The number of degrees clockwise by which you want Elastic Transcoder to rotate the output relative to the input. Enter one of the following values:

auto , 0 , 90 , 180 , 270

The value `auto` generally works only if the file that you're transcoding contains rotation metadata.

- **PresetId** (*string*) –

The value of the `Id` object for the preset that you want to use for this job. The preset determines the audio, video, and

thumbnail settings that Elastic Transcoder uses for transcoding. To use a preset that you created, specify the preset ID that Elastic Transcoder returned in the response when you created the preset. You can also use the Elastic Transcoder system presets, which you can get with `ListPresets`.

- **SegmentDuration** (*string*) –

Warning: (Outputs in Fragmented MP4 or MPEG-TS format only.

If you specify a preset in `PresetId` for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `SegmentDuration` is the target maximum duration of each segment in seconds. For `HLSv3` format playlists, each media segment is stored in a separate `.ts` file. For `HLSv4` and `Smooth` playlists, all media segments for an output are stored in a single file. Each segment is approximately the length of the `SegmentDuration`, though individual segments might be shorter or longer.

The range of valid values is 1 to 60 seconds. If the duration of the video is not evenly divisible by `SegmentDuration`, the duration of the last segment is the remainder of `total length/SegmentDuration`.

Elastic Transcoder creates an output-specific playlist for each output `HLS` output that you specify in `OutputKeys`. To add an output to the master playlist for this job, include it in the `OutputKeys` of the associated playlist.

- **Status** (*string*) –

The status of one output in a job. If you specified only one output for the job, `Outputs:Status` is always the same as `Job:Status`. If you specified more than one output:

- `Job:Status` and `Outputs:Status` for all of the outputs is `Submitted` until Elastic Transcoder starts to process the first output.
- When Elastic Transcoder starts to process the first output, `Outputs:Status` for that output and `Job:Status` both change to `Progressing`. For each output, the value of `Outputs:Status` remains `Submitted` until Elastic Transcoder starts to process the output.
- `Job:Status` remains `Progressing` until all of the outputs reach a terminal status, either `Complete` or `Error`.
- When all of the outputs reach a terminal status, `Job:Status` changes to `Complete` only if `Outputs:Status` for all of the outputs is `Complete`. If `Outputs:Status` for one or more outputs is `Error`, the terminal status for `Job:Status` is also `Error`.

The value of `Status` is one of the following: `Submitted`, `Progressing`, `Complete`, `Canceled`, or `Error`.

- **StatusDetail** (*string*) –

Information that further explains `Status`.

- **Duration** (*integer*) –

Duration of the output file, in seconds.

- **Width** (*integer*) –

Specifies the width of the output file in pixels.

- **Height** (*integer*) –

Height of the output file, in pixels.

- **FrameRate** (*string*) –

Frame rate of the output file, in frames per second.

- **FileSize** (*integer*) –

File size of the output file, in bytes.

- **DurationMillis** (*integer*) –

Duration of the output file, in milliseconds.

- **Watermarks** (*list*) –

Information about the watermarks that you want Elastic Transcoder to add to the video during transcoding. You can specify up to four watermarks for each output. Settings for each watermark must be defined in the preset that you specify in `Preset` for the current output.

Watermarks are added to the output video in the sequence in which you list them in the job output. The first watermark in the list is added to the output video first, the second watermark in the list is added next, and so on. As a result, if the settings in a preset cause Elastic Transcoder to place all watermarks in the same location, the second watermark that you add will cover the first one, the third one will cover the second, and the fourth one will cover the third.

- (*dict*) –

Watermarks can be in .png or .jpg format. If you want to display a watermark that is not rectangular, use the .png format, which supports transparency.

- **PresetWatermarkId** (*string*) –

The ID of the watermark settings that Elastic Transcoder uses to add watermarks to the video during transcoding. The settings are in the preset specified by `Preset` for the current output. In that preset, the value of `Watermarks Id` tells Elastic Transcoder which settings to use.

- **InputKey** (*string*) –

The name of the .png or .jpg file that you want to use for the watermark. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `Pipeline`; the `Input Bucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, **logos/128x64.png**, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your watermarks.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector

must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AlbumArt** (*dict*) –

The album art to be associated with the output file, if any.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder will handle the existence of multiple album artwork files.

- **Replace**: The specified album art will replace any existing album art.
- **Prepend**: The specified album art will be placed in front of any existing album art.
- **Append**: The specified album art will be placed after any existing album art.
- **Fallback**: If the original input file contains artwork, Elastic Transcoder will use that artwork for the output. If the original input does not contain artwork, Elastic Transcoder will use the specified album art file.

- **Artwork** (*list*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20. Valid formats are .jpg and .png

- (*dict*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20.

To remove artwork or leave the artwork empty, you can either set `Artwork` to null, or set the `MergePolicy` to “Replace” and use an empty `Artwork` array.

To pass through existing artwork unchanged, set the `MergePolicy` to “Prepend”, “Append”, or “Fallback”, and use an empty `Artwork` array.

- **InputKey** (*string*) –

The name of the file to be used as album art. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/pie.jpg`, include the prefix in the key. If the file isn’t in the specified bucket, Elastic Transcoder returns an error.

- **MaxWidth** (*string*) –

The maximum width of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

- **MaxHeight** (*string*) –

The maximum height of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default

value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.

- **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of the output album art:

- **Fit**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Fill**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output art and then crops it in the dimension (if any) that exceeds the maximum value.
- **Stretch**: Elastic Transcoder stretches the output art to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input art and the output art are different, the output art will be distorted.
- **Keep**: Elastic Transcoder does not scale the output art. If either dimension of the input art exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output art.
- **ShrinkToFit**: Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the art up.
- **ShrinkToFill**: Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the art up.

- **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder may add white bars to the top and bottom and/or left and right sides of the output album art to make the total size of the output art match the values that you specified for `MaxWidth` and `MaxHeight`.

- **AlbumArtFormat** (*string*) –

The format of album art, if any. Valid formats are `.jpg` and `.png`.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your artwork.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3**: Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS**: Amazon S3 calls the Amazon Key Manage-

ment Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.

- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Composition** (*list*) –

You can create an output file that contains an excerpt from the input file. This excerpt, called a clip, can come from the beginning, middle, or end of the file. The Composition object contains settings for the clips that make up an output file. For the current release, you can only specify settings for a single

clip per output file. The Composition object cannot be null.

- **(dict)** –

Settings for one clip in a composition. All jobs in a playlist must have the same clip settings.

- **TimeSpan** (dict) –

Settings that determine when a clip begins and how long it lasts.

- **StartTime** (string) –

The place in the input file where you want a clip to start. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder starts at the beginning of the input file.

- **Duration** (string) –

The duration of the clip. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder creates an output file from StartTime to the end of the file.

If you specify a value longer than the duration of the input file, Elastic Transcoder transcodes the file and returns a warning message.

- **Captions** (dict) –

You can configure Elastic Transcoder to transcode captions, or subtitles, from one format to another. All captions must be in UTF-8. Elastic Transcoder supports two types of captions:

- **Embedded:** Embedded captions are included in the same file as the audio and video. Elastic Transcoder supports only one embedded caption per language, to a maximum of 300 embedded captions per file. Valid input values include: CEA-608 (EIA-608 , first non-empty channel only), CEA-708 (EIA-708 , first non-empty channel only), and mov-text. Valid outputs include: mov-text. Elastic Transcoder supports a maximum of one embedded format per output.

- **Sidecar:** Sidecar captions are kept in a separate metadata file from the audio and video data. Sidecar captions require a player that is capable of understanding the relationship between the video file and the sidecar file. Elastic Transcoder supports only one sidecar caption per language, to a maximum of 20 sidecar captions per file. Valid input values include: dfxp (first div element only), ebu-tt , scc , smpt , srt , ttml (first div element only), and webvtt. Valid outputs include: dfxp (first div element only), scc , srt , and webvtt .

If you want ttml or smpte-tt compatible captions, specify dfxp as your output format.

Elastic Transcoder does not support OCR (Optical Character Recognition), does not accept pictures as a valid input for captions, and is not available for audio-only transcoding. Elastic

Transcoder does not preserve text formatting (for example, italics) during the transcoding process.

To remove captions or leave the captions empty, set `Captions` to `null`. To pass through existing captions unchanged, set the `MergePolicy` to `MergeRetain`, and pass in a null `CaptionSources` array.

For more information on embedded files, see the Subtitles Wikipedia page.

For more information on sidecar files, see the Extensible Metadata Platform and Sidecar file Wikipedia pages.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder handles the existence of multiple captions.

- **MergeOverride:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the sidecar captions and ignores the embedded captions for that language.

- **MergeRetain:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the embedded captions and ignores the sidecar captions for that language. If `CaptionSources` is empty, Elastic Transcoder omits all sidecar captions from the output files.

- **Override:** Elastic Transcoder transcodes only the sidecar captions that you specify in `CaptionSources`. `MergePolicy` cannot be null.

- **CaptionSources** (*list*) –

Source files for the input sidecar captions used during the transcoding process. To omit all sidecar captions, leave `CaptionSources` blank.

- (*dict*) –

A source file for the input sidecar captions used during the transcoding process.

- **Key** (*string*) –

The name of the sidecar caption file that you want Elastic Transcoder to include in the output file.

- **Language** (*string*) –

A string that specifies the language of the caption. Specify this as one of:

- 2-character ISO 639-1 code
- 3-character ISO 639-2 code

For more information on ISO language codes and language names, see the List of ISO 639-1 codes.

- **TimeOffset** (*string*) –

For clip generation or captions that do not start at the same time as the associated video file, the `TimeOffset` tells Elastic Transcoder how much of the video to encode before in-

cluding captions.

Specify the TimeOffset in the form `[+-]SS.sss` or `[+-]HH:mm:ss.ss`.

- **Label** (*string*) –

The label of the caption shown in the player when choosing a language. We recommend that you put the caption language name here, in the language of the captions.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption sources.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3**: Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS**: Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7**: A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR**: AES Counter Mode.
- **AES-GCM**: AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **CaptionFormats** (*list*) –

The array of file formats for the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- (*dict*) –

The file format of the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- **Format** (*string*) –

The format you specify determines whether Elastic Transcoder generates an embedded or sidecar caption for this output.

- **Valid Embedded Caption Formats:** * **for FLAC** : None

- **For MP3** : None

- **For MP4** : mov-text

- **For MPEG-TS** : None

- **For ogg** : None

- **For webm** : None

- **Valid Sidecar Caption Formats:** Elastic Transcoder supports dfxp (first div element only), scc, srt, and webvtt. If you want ttml or smpte-tt compatible captions, specify dfxp as your output format. * **For FMP4** : dfxp

- **Non-FMP4 outputs** : All sidecar types

fmp4 captions have an extension of .ismt

- **Pattern** (*string*) –

The prefix for caption filenames, in the form *description* - {language} , where:

- *description* is a description of the video.

- {language} is a literal value that Elastic Transcoder replaces with the two- or three-letter code for the language of the caption in the output file names.

If you don't include {language} in the file name pattern, Elastic Transcoder automatically appends "{language}" to the value that you specify for the description. In addition, Elastic Transcoder automatically appends the count to the end of the segment files.

For example, suppose you're transcoding into srt format. When you enter "Sydney-{language}-sunrise", and the language of the captions is English (en), the name of the first caption file will be Sydney-en-sunrise00000.srt.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption formats.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**

- **Key MD5**

- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your output files. If you choose to use encryption, you must specify a mode to use. If you choose not to use encryption, Elastic Transcoder will write an unencrypted file to your Amazon S3 bucket.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AppliedColorSpaceConversion** (*string*) –

If Elastic Transcoder used a preset with a `ColorSpaceConversionMode` to transcode the output file, the `AppliedColorSpaceConversion` parameter shows the conversion used. If no `ColorSpaceConversionMode` was defined in the preset, this parameter will not be included in the job response.

- * **Outputs** (*list*) –

Information about the output files. We recommend that you use the `Outputs` syntax for all jobs, even when you want Elastic Transcoder to transcode a file into only one format. Do not use both the `Outputs` and `Output` syntaxes in the same request. You can create a maximum of 30 outputs per job.

If you specify more than one output for a job, Elastic Transcoder creates the files for each output in the order in which you specify them in the job.

- (*dict*) –

Warning: Outputs recommended instead.

If you specified one output for a job, information about that output. If you specified multiple outputs for a job, the `Output` object lists information about the first output. This duplicates the information that is listed for the first output in the `Outputs` object.

- **Id** (*string*) –

A sequential counter, starting with 1, that identifies an output among the outputs from the current job. In the `Output` syntax, this value is always 1.

- **Key** (*string*) –

The name to assign to the transcoded file. Elastic Transcoder saves the file in the Amazon S3 bucket specified by the `OutputBucket` object in the pipeline that is specified by the pipeline ID.

- **ThumbnailPattern** (*string*) –

Whether you want Elastic Transcoder to create thumbnails for your videos and, if so, how you want Elastic Transcoder to name the files.

If you don't want Elastic Transcoder to create thumbnails, specify `""`.

If you do want Elastic Transcoder to create thumbnails, specify the information that you want to include in the file name for each thumbnail. You can specify the following values in any sequence:

- **“{count}” (Required)** : If you want to create thumbnails, you must include `{count}` in the `ThumbnailPattern` object. Wherever you specify `{count}`, Elastic Transcoder adds a five-digit sequence number (beginning with **00001**) to thumbnail file names. The number indicates where a given thumbnail appears in the sequence of thumbnails for a transcoded file. .. warning::If you specify a literal value and/or `{resolution}` but you omit `{count}`, Elastic Transcoder returns a validation error and does not create the job.
- **Literal values (Optional)** : You can specify literal values anywhere in the `ThumbnailPattern` object. For example, you can include them as a file name prefix or as a delimiter between `{resolution}` and `{count}`.
- **“{resolution}” (Optional)** : If you want Elastic Transcoder to include the resolution in the file name, include `{resolution}` in the `ThumbnailPattern` object. When creating thumbnails, Elastic Transcoder automatically saves the files in the format (.jpg or .png) that appears in the preset that you specified in the `PresetID` value of `CreateJobOutput`. Elastic Transcoder also appends the applicable file name extension.

- **ThumbnailEncryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your thumbnail.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3**: Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS**: Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7**: A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR**: AES Counter Mode.
- **AES-GCM**: AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will

fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Rotate** (*string*) –

The number of degrees clockwise by which you want Elastic Transcoder to rotate the output relative to the input. Enter one of the following values:

`auto` , 0 , 90 , 180 , 270

The value `auto` generally works only if the file that you're transcoding contains rotation metadata.

- **PresetId** (*string*) –

The value of the `Id` object for the preset that you want to use for this job. The preset determines the audio, video, and thumbnail settings that Elastic Transcoder uses for transcoding. To use a preset that you created, specify the preset ID that Elastic Transcoder returned in the response when you created the preset. You can also use the Elastic Transcoder system

presets, which you can get with `ListPresets`.

- **SegmentDuration** (*string*) –

Warning: (Outputs in Fragmented MP4 or MPEG-TS format only.

If you specify a preset in `PresetId` for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `SegmentDuration` is the target maximum duration of each segment in seconds. For `HLSv3` format playlists, each media segment is stored in a separate `.ts` file. For `HLSv4` and `Smooth` playlists, all media segments for an output are stored in a single file. Each segment is approximately the length of the `SegmentDuration`, though individual segments might be shorter or longer.

The range of valid values is 1 to 60 seconds. If the duration of the video is not evenly divisible by `SegmentDuration`, the duration of the last segment is the remainder of `total length/SegmentDuration`.

Elastic Transcoder creates an output-specific playlist for each output `HLS` output that you specify in `OutputKeys`. To add an output to the master playlist for this job, include it in the `OutputKeys` of the associated playlist.

- **Status** (*string*) –

The status of one output in a job. If you specified only one output for the job, `Outputs:Status` is always the same as `Job:Status`. If you specified more than one output:

- `Job:Status` and `Outputs:Status` for all of the outputs is `Submitted` until Elastic Transcoder starts to process the first output.
- When Elastic Transcoder starts to process the first output, `Outputs:Status` for that output and `Job:Status` both change to `Progressing`. For each output, the value of `Outputs:Status` remains `Submitted` until Elastic Transcoder starts to process the output.
- `Job:Status` remains `Progressing` until all of the outputs reach a terminal status, either `Complete` or `Error`.
- When all of the outputs reach a terminal status, `Job:Status` changes to `Complete` only if `Outputs:Status` for all of the outputs is `Complete`. If `Outputs:Status` for one or more outputs is `Error`, the terminal status for `Job:Status` is also `Error`.

The value of `Status` is one of the following: `Submitted`, `Progressing`, `Complete`, `Canceled`, or `Error`.

- **StatusDetail** (*string*) –

Information that further explains `Status`.

- **Duration** (*integer*) –

Duration of the output file, in seconds.

- **Width** (*integer*) –

Specifies the width of the output file in pixels.

- **Height** (*integer*) –

Height of the output file, in pixels.

- **FrameRate** (*string*) –

Frame rate of the output file, in frames per second.

- **FileSize** (*integer*) –

File size of the output file, in bytes.

- **DurationMillis** (*integer*) –

Duration of the output file, in milliseconds.

- **Watermarks** (*list*) –

Information about the watermarks that you want Elastic Transcoder to add to the video during transcoding. You can specify up to four watermarks for each output. Settings for each watermark must be defined in the preset that you specify in `Preset` for the current output.

Watermarks are added to the output video in the sequence in which you list them in the job output. The first watermark in the list is added to the output video first, the second watermark in the list is added next, and so on. As a result, if the settings in a preset cause Elastic Transcoder to place all watermarks in the same location, the second watermark that you add will cover the first one, the third one will cover the second, and the fourth one will cover the third.

- (*dict*) –

Watermarks can be in .png or .jpg format. If you want to display a watermark that is not rectangular, use the .png format, which supports transparency.

- **PresetWatermarkId** (*string*) –

The ID of the watermark settings that Elastic Transcoder uses to add watermarks to the video during transcoding. The settings are in the preset specified by `Preset` for the current output. In that preset, the value of `Watermarks Id` tells Elastic Transcoder which settings to use.

- **InputKey** (*string*) –

The name of the .png or .jpg file that you want to use for the watermark. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `Pipeline`; the `Input Bucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, **logos/128x64.png**, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your watermarks.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AlbumArt** (*dict*) –

The album art to be associated with the output file, if any.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder will handle the existence of multiple album artwork files.

- **Replace**: The specified album art will replace any existing album art.
- **Prepend**: The specified album art will be placed in front of any existing album art.
- **Append**: The specified album art will be placed after any existing album art.
- **Fallback**: If the original input file contains artwork, Elastic Transcoder will use that artwork for the output. If the original input does not contain artwork, Elastic Transcoder will use the specified album art file.

- **Artwork** (*list*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20. Valid formats are .jpg and .png

- (*dict*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20.

To remove artwork or leave the artwork empty, you can either set `Artwork` to null, or set the `Merge Policy` to “Replace” and use an empty `Artwork` array.

To pass through existing artwork unchanged, set the `Merge Policy` to “Prepend”, “Append”, or “Fallback”, and use an empty `Artwork` array.

- **InputKey** (*string*) –

The name of the file to be used as album art. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/pie.jpg`, include the prefix in the key. If the file isn’t in the specified bucket, Elastic Transcoder returns an error.

- **MaxWidth** (*string*) –

The maximum width of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

- **MaxHeight** (*string*) –

The maximum height of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.

- **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of the output album art:

- **Fit:** Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Fill:** Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output art and then crops it in the dimension (if any) that exceeds the maximum value.
- **Stretch:** Elastic Transcoder stretches the output art to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input art and the output art are different, the output art will be distorted.
- **Keep:** Elastic Transcoder does not scale the output art. If either dimension of the input art exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output art.
- **ShrinkToFit:** Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the art up.
- **ShrinkToFill** Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the art up.

- **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder may add white bars to the top and bottom and/or left and right sides of the output album art to make the total size of the output art match the values that you specified for `MaxWidth` and `MaxHeight`.

- **AlbumArtFormat** (*string*) –

The format of album art, if any. Valid formats are `.jpg` and `.png`.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your artwork.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.

- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Composition** (*list*) –

You can create an output file that contains an excerpt from the input file. This excerpt, called a clip, can come from the beginning, middle, or end of the file. The Composition object contains settings for the clips that make up an output file. For the current release, you can only specify settings for a single clip per output file. The Composition object cannot be null.

- (*dict*) –

Settings for one clip in a composition. All jobs in a playlist must have the same clip settings.

- **TimeSpan** (*dict*) –

Settings that determine when a clip begins and how long it lasts.

- **StartTime** (*string*) –

The place in the input file where you want a clip to start. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder starts at the beginning of the input file.

- **Duration** (*string*) –

The duration of the clip. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder creates an output file from StartTime to the end of the file.

If you specify a value longer than the duration of the input file, Elastic Transcoder transcodes the file and returns a warning message.

- **Captions** (*dict*) –

You can configure Elastic Transcoder to transcode captions, or subtitles, from one format to another. All captions must be in UTF-8. Elastic Transcoder supports two types of captions:

- **Embedded:** Embedded captions are included in the same file as the audio and video. Elastic Transcoder supports only one embedded caption per language, to a maximum of 300 embedded captions per file. Valid input values include: CEA-608 (EIA-608 , first non-empty channel only), CEA-708 (EIA-708 , first non-empty channel only), and mov-text. Valid outputs include: mov-text. Elastic Transcoder supports a maximum of one embedded format per output.
- **Sidecar:** Sidecar captions are kept in a separate metadata file from the audio and video data. Sidecar captions require a player that is capable of understanding the relationship between the video file and the sidecar file. Elastic Transcoder supports only one sidecar caption per language, to a maximum of 20 sidecar captions per file. Valid input values include: dfxp (first div element only), ebu-tt , scc , smpt , srt , ttl (first div element only), and webvtt. Valid outputs include: dfxp (first div element only), scc , srt , and webvtt .

If you want ttl or smpte-tt compatible captions, specify dfxp as your output format.

Elastic Transcoder does not support OCR (Optical Character Recognition), does not accept pictures as a valid input for captions, and is not available for audio-only transcoding. Elastic Transcoder does not preserve text formatting (for example, italics) during the transcoding process.

To remove captions or leave the captions empty, set Captions to null. To pass through existing captions unchanged, set the MergePolicy to MergeRetain , and pass in a null CaptionSources array.

For more information on embedded files, see the Subtitles Wikipedia page.

For more information on sidecar files, see the Extensible Metadata Platform and Sidecar file Wikipedia pages.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder handles the existence of multiple captions.

- **MergeOverride:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the sidecar captions and ignores the embedded captions for that language.

- **MergeRetain:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the embedded captions and ignores the sidecar captions for that language. If `CaptionSources` is empty, Elastic Transcoder omits all sidecar captions from the output files.

- **Override:** Elastic Transcoder transcodes only the sidecar captions that you specify in `CaptionSources`.

`MergePolicy` cannot be null.

- **CaptionSources** (*list*) –

Source files for the input sidecar captions used during the transcoding process. To omit all sidecar captions, leave `CaptionSources` blank.

- (*dict*) –

A source file for the input sidecar captions used during the transcoding process.

- **Key** (*string*) –

The name of the sidecar caption file that you want Elastic Transcoder to include in the output file.

- **Language** (*string*) –

A string that specifies the language of the caption. Specify this as one of:

- 2-character ISO 639-1 code
- 3-character ISO 639-2 code

For more information on ISO language codes and language names, see the List of ISO 639-1 codes.

- **TimeOffset** (*string*) –

For clip generation or captions that do not start at the same time as the associated video file, the `TimeOffset` tells Elastic Transcoder how much of the video to encode before including captions.

Specify the `TimeOffset` in the form `[+-]SS.sss` or `[+-]HH:mm:SS.ss`.

- **Label** (*string*) –

The label of the caption shown in the player when choosing a language. We recommend that you put the caption language

name here, in the language of the captions.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption sources.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **CaptionFormats** (*list*) –

The array of file formats for the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- (*dict*) –

The file format of the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- **Format** (*string*) –

The format you specify determines whether Elastic Transcoder generates an embedded or sidecar caption for this output.

- **Valid Embedded Caption Formats:** * **for FLAC** : None

- **For MP3** : None

- **For MP4** : mov-text

- **For MPEG-TS** : None

- **For ogg** : None

- **For webm** : None

- **Valid Sidecar Caption Formats:** Elastic Transcoder supports dfxp (first div element only), scc, srt, and webvtt. If you want ttlml or smpte-tt compatible captions, specify dfxp as your output format. * **For FMP4** : dfxp

- **Non-FMP4 outputs** : All sidecar types

fm4 captions have an extension of .ismt

- **Pattern** (*string*) –

The prefix for caption filenames, in the form *description* - {language}, where:

- *description* is a description of the video.

- {language} is a literal value that Elastic Transcoder replaces with the two- or three-letter code for the language of the caption in the output file names.

If you don't include {language} in the file name pattern, Elastic Transcoder automatically appends "{language}" to the value that you specify for the description. In addition, Elastic Transcoder automatically appends the count to the end of the segment files.

For example, suppose you're transcoding into srt format. When you enter "Sydney-{language}-sunrise", and the language of the captions is English (en), the name of the first caption file will be Sydney-en-sunrise00000.srt.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption formats.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or en-

crypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your output files. If you choose to use encryption, you must specify a mode to use. If you choose not to use encryption, Elastic Transcoder will write an unencrypted file to your Amazon S3 bucket.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AppliedColorSpaceConversion** (*string*) –

If Elastic Transcoder used a preset with a `ColorSpaceConversionMode` to transcode the output file, the `AppliedColorSpaceConversion` parameter shows the conversion used. If no `ColorSpaceConversionMode` was defined in the preset, this parameter will not be included in the job response.

- * **OutputKeyPrefix** (*string*) –

The value, if any, that you want Elastic Transcoder to prepend to the names of all files that this job creates, including output files, thumbnails, and playlists. We recommend that you add a / or some other delimiter to the end of the `OutputKeyPrefix`.

- * **Playlists** (*list*) –

Warning: Outputs in Fragmented MP4 or MPEG-TS format only.

If you specify a preset in `PresetId` for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `Playlists` contains information about the master playlists that you want Elastic Transcoder to create.

The maximum number of master playlists in a job is 30.

- (*dict*) –

Use Only for Fragmented MP4 or MPEG-TS Outputs. If you specify a preset for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `Playlists` contains information about the master playlists that you want Elastic Transcoder to create. We recommend that you create only one master playlist per output format. The maximum number of master playlists in a job is 30.

- **Name** (*string*) –

The name that you want Elastic Transcoder to assign to the master playlist, for example, `nyc-vacation.m3u8`. If the name includes a / character, the section of the name before the last / must be identical for all `Name` objects. If you create more than one master playlist, the values of all `Name` objects must be unique.

Note : Elastic Transcoder automatically appends the relevant file extension to the file name (`.m3u8` for `HLSv3` and `HLSv4` playlists, and `.ism` and `.ismc` for `Smooth` playlists). If you include a file extension in `Name`, the file name will have two extensions.

- **Format** (*string*) –

The format of the output playlist. Valid formats include `HLSv3`, `HLSv4`, and `Smooth`.

- **OutputKeys** (*list*) –

For each output in this job that you want to include in a master playlist, the value of the `Outputs:Key` object.

- If your output is not `HLS` or does not have a segment duration set, the name of the output file is a concatenation of `OutputKeyPrefix` and `Outputs:Key:OutputKeyPrefix“Outputs:Key“`

- If your output is `HLSv3` and has a segment duration set, or is not included in a playlist, Elastic Transcoder creates an output playlist file with a file extension of `.m3u8`, and a series of `.ts` files that include a five-digit sequential counter beginning with 00000: `OutputKeyPrefix“Outputs:Key“ .m3u8`
`OutputKeyPrefix“Outputs:Key“ 00000.ts`

- If your output is `HLSv4`, has a segment duration set, and is included in an `HLSv4` playlist, Elastic Transcoder creates an output playlist file with a file extension of `_v4.m3u8`. If the output is video, Elastic Transcoder also creates an output file with an extension of `_iframe.m3u8`: `OutputKeyPrefix“Outputs:Key“ _v4.m3u8`
`OutputKeyPrefix“Outputs:Key“ _iframe.m3u8`
`OutputKeyPrefix“Outputs:Key“ .ts`

Elastic Transcoder automatically appends the relevant file extension to the file name. If you include a file extension in `Output Key`, the file name will have two extensions.

If you include more than one output in a playlist, any segment duration settings, clip settings, or caption settings must be the same for all outputs in the playlist. For `Smooth` playlists, the `Audio:Profile`, `Video:Profile`, and `Video:FrameRate` to `Video:KeyframesMaxDist` ratio must be the same for all outputs.

- (*string*) –

- **HlsContentProtection** (*dict*) –

The HLS content protection settings, if any, that you want Elastic Transcoder to apply to the output files associated with this playlist.

- **Method** (*string*) –

The content protection method for your output. The only valid value is: `aes-128`.

This value will be written into the method attribute of the `EXT-X-KEY` metadata tag in the output playlist.

- **Key** (*string*) –

If you want Elastic Transcoder to generate a key for you, leave this field blank.

If you choose to supply your own key, you must encrypt the key by using AWS KMS. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

- **KeyMd5** (*string*) –

If Elastic Transcoder is generating your key for you, you must leave this field blank.

The MD5 digest of the key that you want Elastic Transcoder to use to encrypt your output file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

- **InitializationVector** (*string*) –

If Elastic Transcoder is generating your key for you, you must leave this field blank.

The series of random bits created by a random bit generator, unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

- **LicenseAcquisitionUrl** (*string*) –

The location of the license key required to decrypt your HLS playlist. The URL must be an absolute path, and is referenced in the URI attribute of the EXT-X-KEY metadata tag in the playlist file.

- **KeyStoragePolicy** (*string*) –

Specify whether you want Elastic Transcoder to write your HLS license key to an Amazon S3 bucket. If you choose `WithVariantPlaylists` , `LicenseAcquisitionUrl` must be left blank and Elastic Transcoder writes your data key into the same bucket as the associated playlist.

- **PlayReadyDrm** (*dict*) –

The DRM settings, if any, that you want Elastic Transcoder to apply to the output files associated with this playlist.

- **Format** (*string*) –

The type of DRM, if any, that you want Elastic Transcoder to apply to the output files associated with this playlist.

- **Key** (*string*) –

The DRM key for your file, provided by your DRM license provider. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using AWS KMS.

- **KeyMd5** (*string*) –

The MD5 digest of the key used for DRM on your file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

- **KeyId** (*string*) –

The ID for your DRM key, so that your DRM license provider knows which key to provide.

The key ID must be provided in big endian, and Elastic Transcoder will convert it to little endian before inserting it into the PlayReady DRM headers. If you are unsure whether your license server provides your key ID in big or little endian, check with your DRM provider.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your files. The initialization vector must be base64-encoded, and it must be exactly 8 bytes long before being base64-encoded. If no initialization vector is provided, Elastic Transcoder generates one for you.

- **LicenseAcquisitionUrl** (*string*) –

The location of the license key required to play DRM content. The URL must be an absolute path, and is referenced by the PlayReady header. The PlayReady header is referenced in the protection header of the client manifest for Smooth Streaming outputs, and in the EXT-X-DXDRM and EXT-XDXDRMINFO metadata tags for HLS playlist outputs. An example URL looks like this: <https://www.example.com/exampleKey/>

- **Status** (*string*) –

The status of the job with which the playlist is associated.

- **StatusDetail** (*string*) –

Information that further explains the status.

- * **Status** (*string*) –

The status of the job: Submitted, Progressing, Complete, Canceled, or Error.

- * **UserMetadata** (*dict*) –

User-defined metadata that you want to associate with an Elastic Transcoder job. You specify metadata in `key/value` pairs, and you can add up to 10 `key/value` pairs per job. Elastic Transcoder does not guarantee that `key/value` pairs will be returned in the same order in which you specify them.

Metadata keys and values must use characters from the following list:

- 0–9
- A–Z and a–z
- Space
- The following symbols: `_ . : / = + - % @`
- (*string*) –
- (*string*) –

- * **Timing** (*dict*) –

Details about the timing of a job.

- **SubmitTimeMillis** (*integer*) –

The time the job was submitted to Elastic Transcoder, in epoch milliseconds.

- **StartTimeMillis** (*integer*) –

The time the job began transcoding, in epoch milliseconds.

- **FinishTimeMillis** (*integer*) –

The time the job finished transcoding, in epoch milliseconds.

create_pipeline (***kwargs*)

The CreatePipeline operation creates a pipeline with settings that you specify.

Request Syntax

```
response = client.create_pipeline(
    Name='string',
    InputBucket='string',
    OutputBucket='string',
    Role='string',
    AwsKmsKeyArn='string',
    Notifications={
        'Progressing': 'string',
        'Completed': 'string',
        'Warning': 'string',
        'Error': 'string'
    },
    ContentConfig={
        'Bucket': 'string',
        'StorageClass': 'string',
        'Permissions': [
            {
                'GranteeType': 'string',
                'Grantee': 'string',
                'Access': [
                    'string',
                ]
            }
        ],
    },
    ThumbnailConfig={
        'Bucket': 'string',
        'StorageClass': 'string',
        'Permissions': [
            {
                'GranteeType': 'string',
                'Grantee': 'string',
                'Access': [
                    'string',
                ]
            }
        ],
    },
)
```

Parameters

- **Name** (*string*) – **[REQUIRED]**

The name of the pipeline. We recommend that the name be unique within the AWS account, but uniqueness is not enforced.

Constraints: Maximum 40 characters.

- **InputBucket** (*string*) – [REQUIRED]

The Amazon S3 bucket in which you saved the media files that you want to transcode.

- **OutputBucket** (*string*) – The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. (Use this, or use `ContentConfig:Bucket` plus `ThumbnailConfig:Bucket`.)

Specify this value when all of the following are true:

- You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.
- You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- You do not want to specify the permissions that Elastic Transcoder grants to the files. .. warning::When Elastic Transcoder saves files in `OutputBucket`, it grants full control over the files only to the AWS account that owns the role that is specified by `Role`.
- You want to associate the transcoded files and thumbnails with the Amazon S3 Standard storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit `OutputBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

- **Role** (*string*) – [REQUIRED]

The IAM Amazon Resource Name (ARN) for the role that you want Elastic Transcoder to use to create the pipeline.

- **AwsKmsKeyArn** (*string*) – The AWS Key Management Service (AWS KMS) key that you want to use with this pipeline.

If you use either `S3` or `S3-AWS-KMS` as your `Encryption:Mode`, you don't need to provide a key with your job because a default key, known as an AWS-KMS key, is created for you automatically. You need to provide an AWS-KMS key only if you want to use a non-default AWS-KMS key, or if you are using an `Encryption:Mode` of `AES-PKCS7`, `AES-CTR`, or `AES-GCM`.

- **Notifications** (*dict*) – The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify to report job status.

Warning: To receive notifications, you must also subscribe to the new topic in the Amazon SNS console.

- **Progressing** : The topic ARN for the Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic. For more information, see [Create a Topic in the Amazon Simple Notification Service Developer Guide](#).
- **Completed** : The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic.
- **Warning** : The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition while processing a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic.
- **Error** : The topic ARN for the Amazon SNS topic that you want to notify

when Elastic Transcoder encounters an error condition while processing a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic.

– **Progressing** (*string*) –

The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process the job.

– **Completed** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing the job.

– **Warning** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition.

– **Error** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition.

- **ContentConfig** (*dict*) – The optional `ContentConfig` object specifies information about the Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists: which bucket to use, which users you want to have access to the files, the type of access you want users to have, and the storage class that you want to assign to the files.

If you specify values for `ContentConfig`, you must also specify values for `ThumbnailConfig`.

If you specify values for `ContentConfig` and `ThumbnailConfig`, omit the `OutputBucket` object.

- **Bucket** : The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists.
- **Permissions** (Optional): The `Permissions` object specifies which users you want to have access to transcoded files and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.
- **Grantee Type** : Specify the type of value that appears in the `Grantee` object:
 - * **Canonical** : The value in the `Grantee` object is either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. For more information about canonical user IDs, see [Access Control List \(ACL\) Overview](#) in the [Amazon Simple Storage Service Developer Guide](#). For more information about using CloudFront origin access identities to require that users use CloudFront URLs instead of Amazon S3 URLs, see [Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content](#). .. warning::A canonical user ID is not the same as an AWS account number.
- **Email** : The value in the `Grantee` object is the registered email address of an AWS account.
- **Group** : The value in the `Grantee` object is one of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.
- **Grantee** : The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group

- **Access** : The permission that you want to give to the AWS user that you specified in `Grantee` . Permissions are granted on the files that Elastic Transcoder adds to the bucket, including playlists and video files. Valid values include: * `READ` : The grantee can read the objects and metadata for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- `READ_ACP` : The grantee can read the object ACL for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- `WRITE_ACP` : The grantee can write the ACL for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
- `FULL_CONTROL` : The grantee has `READ` , `READ_ACP` , and `WRITE_ACP` permissions for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **StorageClass** : The Amazon S3 storage class, `Standard` or `ReducedRedundancy` , that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

– **Bucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. Specify this value when all of the following are true:

- * You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.
- * You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- * You do not want to specify the permissions that Elastic Transcoder grants to the files.
- * You want to associate the transcoded files and thumbnails with the Amazon S3 Standard storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit `OutputBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

– **StorageClass** (*string*) –

The Amazon S3 storage class, `Standard` or `ReducedRedundancy` , that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

– **Permissions** (*list*) –

Optional. The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions` , Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role` . If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions` , Elastic Transcoder grants full control over the transcoded files and playlists to the owner of the role specified by `Role` , and grants no other permissions to any other user or group.

- * (*dict*) –

The `Permission` structure.

- **GranteeType** (*string*) –

The type of value that appears in the Grantee object:

- **Canonical** : Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.
- **Email** : The registered email address of an AWS account.
- **Group** : One of the following predefined Amazon S3 groups: `AllUsers` , `AuthenticatedUsers` , or `LogDelivery` .
- **Grantee** (*string*) –

The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group.

- **Access** (*list*) –

The permission that you want to give to the AWS user that is listed in Grantee. Valid values include:

- **READ** : The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **READ_ACP** : The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **WRITE_ACP** : The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **FULL_CONTROL** : The grantee has **READ**, **READ_ACP**, and **WRITE_ACP** permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- (*string*) –

- **ThumbnailConfig** (*dict*) – The `ThumbnailConfig` object specifies several values, including the Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files, which users you want to have access to the files, the type of access you want users to have, and the storage class that you want to assign to the files.

If you specify values for `ContentConfig` , you must also specify values for `ThumbnailConfig` even if you don't want to create thumbnails.

If you specify values for `ContentConfig` and `ThumbnailConfig` , omit the `OutputBucket` object.

- **Bucket** : The Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files.
- **Permissions** (Optional): The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to thumbnail files, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.
- **GranteeType** : Specify the type of value that appears in the Grantee object:
 - * **Canonical** : The value in the Grantee object is either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.
- **Email** : The value in the Grantee object is the registered email address

of an AWS account.

- **Group** : The value in the `Grantee` object is one of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.
- **Grantee** : The AWS user or group that you want to have access to thumbnail files. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group.
- **Access** : The permission that you want to give to the AWS user that you specified in `Grantee`. Permissions are granted on the thumbnail files that Elastic Transcoder adds to the bucket. Valid values include: `* READ` : The grantee can read the thumbnails and metadata for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- `READ_ACP` : The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `WRITE_ACP` : The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `FULL_CONTROL` : The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **StorageClass** : The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the thumbnails that it stores in your Amazon S3 bucket.
- **Bucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. Specify this value when all of the following are true:

- * You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.
- * You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- * You do not want to specify the permissions that Elastic Transcoder grants to the files.
- * You want to associate the transcoded files and thumbnails with the Amazon S3 Standard storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit `OutputBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

- **StorageClass** (*string*) –

The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

- **Permissions** (*list*) –

Optional. The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions`, Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner

of the role specified by `Role`. If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions`, Elastic Transcoder grants full control over the transcoded files and playlists to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

* (*dict*) –

The `Permission` structure.

- **GranteeType** (*string*) –

The type of value that appears in the Grantee object:

- `Canonical`: Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.
- `Email`: The registered email address of an AWS account.
- `Group`: One of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.
- **Grantee** (*string*) –

The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group.

- **Access** (*list*) –

The permission that you want to give to the AWS user that is listed in Grantee. Valid values include:

- `READ`: The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `READ_ACP`: The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `WRITE_ACP`: The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `FULL_CONTROL`: The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'Pipeline': {
    'Id': 'string',
    'Arn': 'string',
    'Name': 'string',
    'Status': 'string',
    'InputBucket': 'string',
    'OutputBucket': 'string',
    'Role': 'string',
```

```
'AwsKmsKeyArn': 'string',
'Notifications': {
    'Progressing': 'string',
    'Completed': 'string',
    'Warning': 'string',
    'Error': 'string'
},
'ContentConfig': {
    'Bucket': 'string',
    'StorageClass': 'string',
    'Permissions': [
        {
            'GranteeType': 'string',
            'Grantee': 'string',
            'Access': [
                'string',
            ]
        }
    ],
},
'ThumbnailConfig': {
    'Bucket': 'string',
    'StorageClass': 'string',
    'Permissions': [
        {
            'GranteeType': 'string',
            'Grantee': 'string',
            'Access': [
                'string',
            ]
        }
    ],
},
},
'Warnings': [
    {
        'Code': 'string',
        'Message': 'string'
    },
],
]
```

Response Structure

- (*dict*) –

When you create a pipeline, Elastic Transcoder returns the values that you specified in the request.

- **Pipeline** (*dict*) –

A section of the response body that provides information about the pipeline that is created.

- * **Id** (*string*) –

The identifier for the pipeline. You use this value to identify the pipeline in which you want to perform a variety of operations, such as creating a job or a preset.

- * **Arn** (*string*) –

The Amazon Resource Name (ARN) for the pipeline.

* **Name** (*string*) –

The name of the pipeline. We recommend that the name be unique within the AWS account, but uniqueness is not enforced.

Constraints: Maximum 40 characters

* **Status** (*string*) –

The current status of the pipeline:

- **Active** : The pipeline is processing jobs.
- **Paused** : The pipeline is not currently processing jobs.

* **InputBucket** (*string*) –

The Amazon S3 bucket from which Elastic Transcoder gets media files for transcoding and the graphics files, if any, that you want to use for watermarks.

* **OutputBucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files, thumbnails, and playlists. Either you specify this value, or you specify both `ContentConfig` and `ThumbnailConfig`.

* **Role** (*string*) –

The IAM Amazon Resource Name (ARN) for the role that Elastic Transcoder uses to transcode jobs for this pipeline.

* **AwsKmsKeyArn** (*string*) –

The AWS Key Management Service (AWS KMS) key that you want to use with this pipeline.

If you use either `S3` or `S3-AWS-KMS` as your `Encryption:Mode`, you don't need to provide a key with your job because a default key, known as an AWS-KMS key, is created for you automatically. You need to provide an AWS-KMS key only if you want to use a non-default AWS-KMS key, or if you are using an `Encryption:Mode` of `AES-PKCS7`, `AES-CTR`, or `AES-GCM`.

* **Notifications** (*dict*) –

The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify to report job status.

Warning: To receive notifications, you must also subscribe to the new topic in the Amazon SNS console.

- **Progressing** (optional): The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process the job.
- **Completed** (optional): The Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing the job.
- **Warning** (optional): The Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition.
- **Error** (optional): The Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition.
- **Progressing** (*string*) –

The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process the job.

- **Completed** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing the job.

- **Warning** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition.

- **Error** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition.

- * **ContentConfig** (*dict*) –

Information about the Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists. Either you specify both `ContentConfig` and `ThumbnailConfig`, or you specify `OutputBucket`.

- **Bucket** : The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists.
- **Permissions** : A list of the users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access that you want them to have. * **GranteeType**: The type of value that appears in the `Grantee` object: * **Canonical** : Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution.
- **Email** : The registered email address of an AWS account.
- **Group** : One of the following predefined Amazon S3 groups: `AllUsers` , `AuthenticatedUsers` , or `LogDelivery` .
- **Grantee** : The AWS user or group that you want to have access to transcoded files and playlists.
- **Access** : The permission that you want to give to the AWS user that is listed in `Grantee` . Valid values include: * **READ** : The grantee can read the objects and metadata for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **READ_ACP** : The grantee can read the object ACL for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **WRITE_ACP** : The grantee can write the ACL for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **FULL_CONTROL** : The grantee has `READ` , `READ_ACP` , and `WRITE_ACP` permissions for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **StorageClass** : The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.
- **Bucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. Specify this value when all of the following are true:

- You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.
- You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- You do not want to specify the permissions that Elastic Transcoder grants to the files.
- You want to associate the transcoded files and thumbnails with the Amazon S3 Standard storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit `OutputBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

- **StorageClass** (*string*) –

The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

- **Permissions** (*list*) –

Optional. The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions`, Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role`. If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions`, Elastic Transcoder grants full control over the transcoded files and playlists to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

- (*dict*) –

The `Permission` structure.

- **GranteeType** (*string*) –

The type of value that appears in the `Grantee` object:

- `Canonical`: Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning:: A canonical user ID is not the same as an AWS account number.
- `Email`: The registered email address of an AWS account.
- `Group`: One of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.
- **Grantee** (*string*) –

The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an

origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group.

- **Access** (*list*) –

The permission that you want to give to the AWS user that is listed in Grantee. Valid values include:

- **READ** : The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **READ_ACP** : The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **WRITE_ACP** : The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **FULL_CONTROL** : The grantee has **READ**, **READ_ACP**, and **WRITE_ACP** permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- (*string*) –

- * **ThumbnailConfig** (*dict*) –

Information about the Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files. Either you specify both `ContentConfig` and `ThumbnailConfig`, or you specify `OutputBucket`.

- **Bucket** : The Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files.
- **Permissions** : A list of the users and/or predefined Amazon S3 groups you want to have access to thumbnail files, and the type of access that you want them to have. * **GranteeType**: The type of value that appears in the Grantee object: * **Canonical** : Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.
- **Email** : The registered email address of an AWS account.
- **Group** : One of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.
- **Grantee** : The AWS user or group that you want to have access to thumbnail files.
- **Access**: The permission that you want to give to the AWS user that is listed in Grantee. Valid values include: * **READ** : The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **READ_ACP** : The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **WRITE_ACP** : The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **FULL_CONTROL** : The grantee has **READ**, **READ_ACP**, and **WRITE_ACP** permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **StorageClass** : The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the thumbnails that it stores in your Amazon S3 bucket.

- **Bucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. Specify this value when all of the following are true:

- You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.
- You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- You do not want to specify the permissions that Elastic Transcoder grants to the files.
- You want to associate the transcoded files and thumbnails with the Amazon S3 Standard storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit `OutputBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

- **StorageClass** (*string*) –

The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

- **Permissions** (*list*) –

Optional. The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions`, Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role`. If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions`, Elastic Transcoder grants full control over the transcoded files and playlists to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

- (*dict*) –

The `Permission` structure.

- **GranteeType** (*string*) –

The type of value that appears in the `Grantee` object:

- `Canonical`: Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.
- `Email`: The registered email address of an AWS account.
- `Group`: One of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.

- **Grantee** (*string*) –

The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group.

- **Access** (*list*) –

The permission that you want to give to the AWS user that is listed in Grantee. Valid values include:

- **READ** : The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **READ_ACP** : The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **WRITE_ACP** : The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **FULL_CONTROL** : The grantee has **READ**, **READ_ACP**, and **WRITE_ACP** permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.

- (*string*) –

- **Warnings** (*list*) –

Elastic Transcoder returns a warning if the resources used by your pipeline are not in the same region as the pipeline.

Using resources in the same region, such as your Amazon S3 buckets, Amazon SNS notification topics, and AWS KMS key, reduces processing time and prevents cross-regional charges.

- * (*dict*) –

Elastic Transcoder returns a warning if the resources used by your pipeline are not in the same region as the pipeline.

Using resources in the same region, such as your Amazon S3 buckets, Amazon SNS notification topics, and AWS KMS key, reduces processing time and prevents cross-regional charges.

- **Code** (*string*) –

The code of the cross-regional warning.

- **Message** (*string*) –

The message explaining what resources are in a different region from the pipeline.

Note: AWS KMS keys must be in the same region as the pipeline.

create_preset (***kwargs*)

The CreatePreset operation creates a preset with settings that you specify.

Warning: Elastic Transcoder checks the CreatePreset settings to ensure that they meet Elastic Transcoder requirements and to determine whether they comply with H.264 standards. If your settings are not valid for Elastic Transcoder, Elastic Transcoder returns an HTTP 400 response (`ValidationException`) and does not create the preset. If the settings are valid for Elastic Transcoder but aren't strictly compliant with the H.264 standard, Elastic Transcoder creates the preset and returns a warning message in the response. This helps you determine whether your settings comply with the H.264 standard while giving you greater flexibility with respect to the video that Elastic Transcoder produces.

Elastic Transcoder uses the H.264 video-compression format. For more information, see the International Telecommunication Union publication *Recommendation ITU-T H.264: Advanced video coding for generic audiovisual services*.

Request Syntax

```
response = client.create_preset(
    Name='string',
    Description='string',
    Container='string',
    Video={
        'Codec': 'string',
        'CodecOptions': {
            'string': 'string'
        },
        'KeyframesMaxDist': 'string',
        'FixedGOP': 'string',
        'BitRate': 'string',
        'FrameRate': 'string',
        'MaxFrameRate': 'string',
        'Resolution': 'string',
        'AspectRatio': 'string',
        'MaxWidth': 'string',
        'MaxHeight': 'string',
        'DisplayAspectRatio': 'string',
        'SizingPolicy': 'string',
        'PaddingPolicy': 'string',
        'Watermarks': [
            {
                'Id': 'string',
                'MaxWidth': 'string',
                'MaxHeight': 'string',
                'SizingPolicy': 'string',
                'HorizontalAlign': 'string',
                'HorizontalOffset': 'string',
                'VerticalAlign': 'string',
                'VerticalOffset': 'string',
                'Opacity': 'string',
                'Target': 'string'
            },
        ],
    },
    Audio={
        'Codec': 'string',
        'SampleRate': 'string',
        'BitRate': 'string',
        'Channels': 'string',
        'AudioPackingMode': 'string',
    },
)
```

```
        'CodecOptions': {
            'Profile': 'string',
            'BitDepth': 'string',
            'BitOrder': 'string',
            'Signed': 'string'
        }
    },
    Thumbnails={
        'Format': 'string',
        'Interval': 'string',
        'Resolution': 'string',
        'AspectRatio': 'string',
        'MaxWidth': 'string',
        'MaxHeight': 'string',
        'SizingPolicy': 'string',
        'PaddingPolicy': 'string'
    }
}
```

Parameters

- **Name** (*string*) – **[REQUIRED]**

The name of the preset. We recommend that the name be unique within the AWS account, but uniqueness is not enforced.

- **Description** (*string*) – A description of the preset.
- **Container** (*string*) – **[REQUIRED]**

The container type for the output file. Valid values include `flac`, `flv`, `mp4`, `gif`, `mp3`, `mp4`, `mpg`, `mxp`, `oga`, `ogg`, `ts`, and `webm`.

- **Video** (*dict*) – A section of the request body that specifies the video parameters.
 - **Codec** (*string*) –

The video codec for the output file. Valid values include `gif`, `H.264`, `mpeg2`, and `vp8`. You can only specify `vp8` when the container type is `webm`, `gif` when the container type is `gif`, and `mpeg2` when the container type is `mpg`.

- **CodecOptions** (*dict*) –

Profile (H.264/VP8 Only)

The H.264 profile that you want to use for the output file. Elastic Transcoder supports the following profiles:

- * `baseline`: The profile most commonly used for videoconferencing and for mobile applications.
- * `main`: The profile used for standard-definition digital TV broadcasts.
- * `high`: The profile used for high-definition digital TV broadcasts and for Blu-ray discs.

Level (H.264 Only)

The H.264 level that you want to use for the output file. Elastic Transcoder supports the following levels:

`1`, `1b`, `1.1`, `1.2`, `1.3`, `2`, `2.1`, `2.2`, `3`, `3.1`, `3.2`, `4`, `4.1`

MaxReferenceFrames (H.264 Only)

Applicable only when the value of `Video:Codec` is `H.264`. The maximum number of previously decoded frames to use as a reference for decoding future frames. Valid values are integers 0 through 16, but we recommend that you not use a value greater than the following:

```
Min(Floor(Maximum decoded picture buffer
in macroblocks * 256 / (Width in pixels
* Height in pixels)), 16)
```

where *Width in pixels* and *Height in pixels* represent either `MaxWidth` and `MaxHeight`, or `Resolution`. *Maximum decoded picture buffer in macroblocks* depends on the value of the `Level` object. See the list below. (A macroblock is a block of pixels measuring 16x16.)

```
* 1 - 396
* 1b - 396
* 1.1 - 900
* 1.2 - 2376
* 1.3 - 2376
* 2 - 2376
* 2.1 - 4752
* 2.2 - 8100
* 3 - 8100
* 3.1 - 18000
* 3.2 - 20480
* 4 - 32768
* 4.1 - 32768
```

MaxBitRate (Optional, H.264/MPEG2/VP8 only)

The maximum number of bits per second in a video buffer; the size of the buffer is specified by `BufferSize`. Specify a value between 16 and 62,500. You can reduce the bandwidth required to stream a video by reducing the maximum bit rate, but this also reduces the quality of the video.

BufferSize (Optional, H.264/MPEG2/VP8 only)

The maximum number of bits in any *x* seconds of the output video. This window is commonly 10 seconds, the standard segment duration when you're using FMP4 or MPEG-TS for the container type of the output video. Specify an integer greater than 0. If you specify `MaxBitRate` and omit `BufferSize`, Elastic Transcoder sets `BufferSize` to 10 times the value of `MaxBitRate`.

InterlacedMode (Optional, H.264/MPEG2 Only)

The interlace mode for the output video.

Interlaced video is used to double the perceived frame rate for a video by interlacing two fields (one field on every other line, the other field on the other lines) so that the human eye registers multiple pictures per frame. Interlacing reduces the bandwidth required for transmitting a video, but can result in blurred images and flickering.

Valid values include `Progressive` (no interlacing, top to bottom), `TopFirst` (top field first), `BottomFirst` (bottom field first), and `Auto`.

If `InterlaceMode` is not specified, Elastic Transcoder uses `Progressive` for the output. If `Auto` is specified, Elastic Transcoder interlaces the output.

ColorSpaceConversionMode (Optional, H.264/MPEG2 Only)

The color space conversion Elastic Transcoder applies to the output video. Color spaces are the algorithms used by the computer to store information about how to render color. `Bt.601` is the standard for standard definition

video, while `Bt.709` is the standard for high definition video.

Valid values include `None`, `Bt709toBt601`, `Bt601toBt709`, and `Auto`.

If you chose `Auto` for `ColorSpaceConversionMode` and your output is interlaced, your frame rate is one of `23.97`, `24`, `25`, `29.97`, `50`, or `60`, your `SegmentDuration` is null, and you are using one of the resolution changes from the list below, Elastic Transcoder applies the following color space conversions:

- * *Standard to HD, 720x480 to 1920x1080* - Elastic Transcoder applies `Bt601ToBt709`
- * *Standard to HD, 720x576 to 1920x1080* - Elastic Transcoder applies `Bt601ToBt709`
- * *HD to Standard, 1920x1080 to 720x480* - Elastic Transcoder applies `Bt709ToBt601`
- * *HD to Standard, 1920x1080 to 720x576* - Elastic Transcoder applies `Bt709ToBt601`

Note: Elastic Transcoder may change the behavior of the `ColorspaceConversionMode Auto` mode in the future. All outputs in a playlist must use the same `ColorSpaceConversionMode`.

If you do not specify a `ColorSpaceConversionMode`, Elastic Transcoder does not change the color space of a file. If you are unsure what `ColorSpaceConversionMode` was applied to your output file, you can check the `AppliedColorSpaceConversion` parameter included in your job response. If your job does not have an `AppliedColorSpaceConversion` in its response, no `ColorSpaceConversionMode` was applied.

ChromaSubsampling

The sampling pattern for the chroma (color) channels of the output video. Valid values include `yuv420p` and `yuv422p`.

`yuv420p` samples the chroma information of every other horizontal and every other vertical line, `yuv422p` samples the color information of every horizontal line and every other vertical line.

LoopCount (Gif Only)

The number of times you want the output gif to loop. Valid values include `Infinite` and integers between 0 and 100, inclusive.

- * *(string)* –
- *(string)* –

– KeyframesMaxDist (string) –

Applicable only when the value of `Video:Codec` is one of `H.264`, `MPEG2`, or `VP8`.

The maximum number of frames between key frames. Key frames are fully encoded frames; the frames between key frames are encoded based, in part, on the content of the key frames. The value is an integer formatted as a string; valid values are between 1 (every frame is a key frame) and 100000, inclusive. A higher value results in higher compression but may also discernibly decrease video quality.

For `Smooth` outputs, the `FrameRate` must have a constant ratio to the `KeyframesMaxDist`. This allows `Smooth` playlists to switch be-

tween different quality levels while the file is being played.

For example, an input file can have a `FrameRate` of 30 with a `KeyframesMaxDist` of 90. The output file then needs to have a ratio of 1:3. Valid outputs would have `FrameRate` of 30, 25, and 10, and `KeyframesMaxDist` of 90, 75, and 30, respectively.

Alternately, this can be achieved by setting `FrameRate` to `auto` and having the same values for `MaxFrameRate` and `KeyframesMaxDist`.

– **FixedGOP** (*string*) –

Applicable only when the value of `Video:Codec` is one of `H.264`, `MPEG2`, or `VP8`.

Whether to use a fixed value for `FixedGOP`. Valid values are `true` and `false`:

- * `true` : Elastic Transcoder uses the value of `KeyframesMaxDist` for the distance between key frames (the number of frames in a group of pictures, or GOP).
- * `false` : The distance between key frames can vary.

Warning: `FixedGOP` must be set to `true` for `mp4` containers.

– **BitRate** (*string*) –

The bit rate of the video stream in the output file, in kilobits/second. Valid values depend on the values of `Level` and `Profile`. If you specify `auto`, Elastic Transcoder uses the detected bit rate of the input source. If you specify a value other than `auto`, we recommend that you specify a value less than or equal to the maximum H.264-compliant value listed for your level and profile:

Level - Maximum video bit rate in kilobits/second (baseline and main Profile) : maximum video bit rate in kilobits/second (high Profile)

- * 1 - 64 : 80
- * 1b - 128 : 160
- * 1.1 - 192 : 240
- * 1.2 - 384 : 480
- * 1.3 - 768 : 960
- * 2 - 2000 : 2500
- * 3 - 10000 : 12500
- * 3.1 - 14000 : 17500
- * 3.2 - 20000 : 25000
- * 4 - 20000 : 25000
- * 4.1 - 50000 : 62500

– **FrameRate** (*string*) –

The frames per second for the video stream in the output file. Valid values include:

`auto`, `10`, `15`, `23.97`, `24`, `25`, `29.97`, `30`, `60`

If you specify `auto`, Elastic Transcoder uses the detected frame rate of the input source. If you specify a frame rate, we recommend that you perform the following calculation:

Frame rate = maximum recommended
decoding speed in luma samples/second
/ (width in pixels * height in pixels)

where:

- * *width in pixels* and *height in pixels* represent the Resolution of the output video.
- * *maximum recommended decoding speed in Luma samples/second* is less than or equal to the maximum value listed in the following table, based on the value that you specified for Level.

The maximum recommended decoding speed in Luma samples/second for each level is described in the following list (*Level - Decoding speed*):

- * 1 - 380160
- * 1b - 380160
- * 1.1 - 76800
- * 1.2 - 1536000
- * 1.3 - 3041280
- * 2 - 3041280
- * 2.1 - 5068800
- * 2.2 - 5184000
- * 3 - 10368000
- * 3.1 - 27648000
- * 3.2 - 55296000
- * 4 - 62914560
- * 4.1 - 62914560

– **MaxFrameRate** (*string*) –

If you specify `auto` for `FrameRate`, Elastic Transcoder uses the frame rate of the input video for the frame rate of the output video. Specify the maximum frame rate that you want Elastic Transcoder to use when the frame rate of the input video is greater than the desired maximum frame rate of the output video. Valid values include: 10, 15, 23.97, 24, 25, 29.97, 30, 60.

– **Resolution** (*string*) –

Warning: To better control resolution and aspect ratio of output videos, we recommend that you use the values `MaxWidth`, `MaxHeight`, `SizingPolicy`, `PaddingPolicy`, and `DisplayAspectRatio` instead of `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The width and height of the video in the output file, in pixels. Valid values are `auto` and *width x height*:

- * `auto`: Elastic Transcoder attempts to preserve the width and height of the input file, subject to the following rules.
- * `*width* x *height*`: The width and height of the output video in pixels.

Note the following about specifying the width and height:

- * The width must be an even integer between 128 and 4096, inclusive.
- * The height must be an even integer between 96 and 3072, inclusive.
- * If you specify a resolution that is less than the resolution of the input file, Elastic Transcoder rescales the output file to the lower resolution.
- * If you specify a resolution that is greater than the resolution of the input file, Elastic Transcoder rescales the output to the higher resolution.
- * We recommend that you specify a resolution for which the product

of width and height is less than or equal to the applicable value in the following list (*List - Max width x height value*):

- * 1 - 25344
- * 1b - 25344
- * 1.1 - 101376
- * 1.2 - 101376
- * 1.3 - 101376
- * 2 - 101376
- * 2.1 - 202752
- * 2.2 - 404720
- * 3 - 404720
- * 3.1 - 921600
- * 3.2 - 1310720
- * 4 - 2097152
- * 4.1 - 2097152

– **AspectRatio** (*string*) –

Warning: To better control resolution and aspect ratio of output videos, we recommend that you use the values `MaxWidth`, `MaxHeight`, `SizingPolicy`, `PaddingPolicy`, and `DisplayAspectRatio` instead of `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The display aspect ratio of the video in the output file. Valid values include:

`auto`, `1:1`, `4:3`, `3:2`, `16:9`

If you specify `auto`, Elastic Transcoder tries to preserve the aspect ratio of the input file.

If you specify an aspect ratio for the output file that differs from aspect ratio of the input file, Elastic Transcoder adds pillarboxing (black bars on the sides) or letterboxing (black bars on the top and bottom) to maintain the aspect ratio of the active region of the video.

– **MaxWidth** (*string*) –

The maximum width of the output video in pixels. If you specify `auto`, Elastic Transcoder uses 1920 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 128 and 4096.

– **MaxHeight** (*string*) –

The maximum height of the output video in pixels. If you specify `auto`, Elastic Transcoder uses 1080 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 96 and 3072.

– **DisplayAspectRatio** (*string*) –

The value that Elastic Transcoder adds to the metadata in the output file.

– **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of the output video:

- * `Fit`: Elastic Transcoder scales the output video so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- * `Fill`: Elastic Transcoder scales the output video so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and

matches or exceeds the other value. Elastic Transcoder centers the output video and then crops it in the dimension (if any) that exceeds the maximum value.

- * `Stretch` : Elastic Transcoder stretches the output video to match the values that you specified for `MaxWidth` and `MaxHeight` . If the relative proportions of the input video and the output video are different, the output video will be distorted.
- * `Keep` : Elastic Transcoder does not scale the output video. If either dimension of the input video exceeds the values that you specified for `MaxWidth` and `MaxHeight` , Elastic Transcoder crops the output video.
- * `ShrinkToFit` : Elastic Transcoder scales the output video down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the video up.
- * `ShrinkToFill` : Elastic Transcoder scales the output video down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the video up.

– **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad` , Elastic Transcoder may add black bars to the top and bottom and/or left and right sides of the output video to make the total size of the output video match the values that you specified for `MaxWidth` and `MaxHeight` .

– **Watermarks** (*list*) –

Settings for the size, location, and opacity of graphics that you want Elastic Transcoder to overlay over videos that are transcoded using this preset. You can specify settings for up to four watermarks. Watermarks appear in the specified size and location, and with the specified opacity for the duration of the transcoded video.

Watermarks can be in `.png` or `.jpg` format. If you want to display a watermark that is not rectangular, use the `.png` format, which supports transparency.

When you create a job that uses this preset, you specify the `.png` or `.jpg` graphics that you want Elastic Transcoder to include in the transcoded videos. You can specify fewer graphics in the job than you specify watermark settings in the preset, which allows you to use the same preset for up to four watermarks that have different dimensions.

* (*dict*) –

Settings for the size, location, and opacity of graphics that you want Elastic Transcoder to overlay over videos that are transcoded using this preset. You can specify settings for up to four watermarks. Watermarks appear in the specified size and location, and with the specified opacity for the duration of the transcoded video.

Watermarks can be in `.png` or `.jpg` format. If you want to display a watermark that is not rectangular, use the `.png` format, which supports transparency.

When you create a job that uses this preset, you specify the `.png`

or .jpg graphics that you want Elastic Transcoder to include in the transcoded videos. You can specify fewer graphics in the job than you specify watermark settings in the preset, which allows you to use the same preset for up to four watermarks that have different dimensions.

- **Id** (*string*) – A unique identifier for the settings for one watermark. The value of `Id` can be up to 40 characters long.

- **MaxWidth** (*string*) –

The maximum width of the watermark in one of the following formats:

- number of pixels (px): The minimum value is 16 pixels, and the maximum value is the value of `MaxWidth`.
- integer percentage (%): The range of valid values is 0 to 100. Use the value of `Target` to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the calculation.

If you specify the value in pixels, it must be less than or equal to the value of `MaxWidth`.

- **MaxHeight** (*string*) –

The maximum height of the watermark in one of the following formats:

- number of pixels (px): The minimum value is 16 pixels, and the maximum value is the value of `MaxHeight`.
- integer percentage (%): The range of valid values is 0 to 100. Use the value of `Target` to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the calculation.

If you specify the value in pixels, it must be less than or equal to the value of `MaxHeight`.

- **SizingPolicy** (*string*) –

A value that controls scaling of the watermark:

- **Fit** : Elastic Transcoder scales the watermark so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Stretch** : Elastic Transcoder stretches the watermark to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the watermark and the values of `MaxWidth` and `MaxHeight` are different, the watermark will be distorted.
- **ShrinkToFit** : Elastic Transcoder scales the watermark down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the watermark up.

- **HorizontalAlign** (*string*) –

The horizontal position of the watermark unless you specify a non-zero value for `HorizontalOffset`:

- **Left** : The left edge of the watermark is aligned with the left border of the video.
- **Right** : The right edge of the watermark is aligned with the right border of the video.
- **Center** : The watermark is centered between the left and right

borders.

- **HorizontalOffset** (*string*) –

The amount by which you want the horizontal position of the watermark to be offset from the position specified by `HorizontalAlign`:

- number of pixels (px): The minimum value is 0 pixels, and the maximum value is the value of `MaxWidth`.
- integer percentage (%): The range of valid values is 0 to 100. For example, if you specify `Left` for `HorizontalAlign` and `5px` for `HorizontalOffset`, the left side of the watermark appears 5 pixels from the left border of the output video.

`HorizontalOffset` is only valid when the value of `HorizontalAlign` is `Left` or `Right`. If you specify an offset that causes the watermark to extend beyond the left or right border and Elastic Transcoder has not added black bars, the watermark is cropped. If Elastic Transcoder has added black bars, the watermark extends into the black bars. If the watermark extends beyond the black bars, it is cropped.

Use the value of `Target` to specify whether you want to include the black bars that are added by Elastic Transcoder, if any, in the offset calculation.

- **VerticalAlign** (*string*) –

The vertical position of the watermark unless you specify a non-zero value for `VerticalOffset`:

- **Top**: The top edge of the watermark is aligned with the top border of the video.
- **Bottom**: The bottom edge of the watermark is aligned with the bottom border of the video.
- **Center**: The watermark is centered between the top and bottom borders.
- **VerticalOffset** (*string*) – `VerticalOffset`

The amount by which you want the vertical position of the watermark to be offset from the position specified by `VerticalAlign`:

- number of pixels (px): The minimum value is 0 pixels, and the maximum value is the value of `MaxHeight`.
- integer percentage (%): The range of valid values is 0 to 100. For example, if you specify `Top` for `VerticalAlign` and `5px` for `VerticalOffset`, the top of the watermark appears 5 pixels from the top border of the output video.

`VerticalOffset` is only valid when the value of `VerticalAlign` is `Top` or `Bottom`.

If you specify an offset that causes the watermark to extend beyond the top or bottom border and Elastic Transcoder has not added black bars, the watermark is cropped. If Elastic Transcoder has added black bars, the watermark extends into the black bars. If the watermark extends beyond the black bars, it is cropped.

Use the value of `Target` to specify whether you want Elastic

Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the offset calculation.

- **Opacity** (*string*) –

A percentage that indicates how much you want a watermark to obscure the video in the location where it appears. Valid values are 0 (the watermark is invisible) to 100 (the watermark completely obscures the video in the specified location). The datatype of `Opacity` is float.

Elastic Transcoder supports transparent .png graphics. If you use a transparent .png, the transparent portion of the video appears as if you had specified a value of 0 for `Opacity`. The .jpg file format doesn't support transparency.

- **Target** (*string*) –

A value that determines how Elastic Transcoder interprets values that you specified for `HorizontalOffset`, `VerticalOffset`, `MaxWidth`, and `MaxHeight`:

- **Content**: `HorizontalOffset` and `VerticalOffset` values are calculated based on the borders of the video excluding black bars added by Elastic Transcoder, if any. In addition, `MaxWidth` and `MaxHeight`, if specified as a percentage, are calculated based on the borders of the video excluding black bars added by Elastic Transcoder, if any.
- **Frame**: `HorizontalOffset` and `VerticalOffset` values are calculated based on the borders of the video including black bars added by Elastic Transcoder, if any. In addition, `MaxWidth` and `MaxHeight`, if specified as a percentage, are calculated based on the borders of the video including black bars added by Elastic Transcoder, if any.

- **Audio** (*dict*) – A section of the request body that specifies the audio parameters.

- **Codec** (*string*) –

The audio codec for the output file. Valid values include `aac`, `flac`, `mp2`, `mp3`, `pcm`, and `vorbis`.

- **SampleRate** (*string*) –

The sample rate of the audio stream in the output file, in Hertz. Valid values include:

`auto`, `22050`, `32000`, `44100`, `48000`, `96000`

If you specify `auto`, Elastic Transcoder automatically detects the sample rate.

- **BitRate** (*string*) –

The bit rate of the audio stream in the output file, in kilobits/second. Enter an integer between 64 and 320, inclusive.

- **Channels** (*string*) –

The number of audio channels in the output file. The following values are valid:

`auto`, `0`, `1`, `2`

One channel carries the information played by a single speaker. For example, a stereo track with two channels sends one channel to the left speaker,

and the other channel to the right speaker. The output channels are organized into tracks. If you want Elastic Transcoder to automatically detect the number of audio channels in the input file and use that value for the output file, select `auto`.

The output of a specific channel value and inputs are as follows:

- * `auto` **channel specified, with any input:** Pass through up to eight input channels.
- * `0` **channels specified, with any input:** Audio omitted from the output.
- * `1` **channel specified, with at least one input channel:** Mono sound.
- * `2` **channels specified, with any input:** Two identical mono channels or stereo. For more information about tracks, see `Audio:AudioPackingMode`.

For more information about how Elastic Transcoder organizes channels and tracks, see `Audio:AudioPackingMode`.

– **AudioPackingMode** (*string*) –

The method of organizing audio channels and tracks. Use `Audio:Channels` to specify the number of channels in your output, and `Audio:AudioPackingMode` to specify the number of tracks and their relation to the channels. If you do not specify an `Audio:AudioPackingMode`, Elastic Transcoder uses `SingleTrack`.

The following values are valid:

`SingleTrack`, `OneChannelPerTrack`, and `OneChannelPerTrackWithMosTo8Tracks`

When you specify `SingleTrack`, Elastic Transcoder creates a single track for your output. The track can have up to eight channels. Use `SingleTrack` for all non-mxf containers.

The outputs of `SingleTrack` for a specific channel value and inputs are as follows:

- * `0` **channels with any input:** Audio omitted from the output
- * `1, 2, or auto` **channels with no audio input:** Audio omitted from the output
- * `1` **channel with any input with audio:** One track with one channel, downmixed if necessary
- * `2` **channels with one track with one channel:** One track with two identical channels
- * `2 or auto` **channels with two tracks with one channel each:** One track with two channels
- * `2 or auto` **channels with one track with two channels:** One track with two channels
- * `2` **channels with one track with multiple channels:** One track with two channels
- * `auto` **channels with one track with one channel:** One track with one channel
- * `auto` **channels with one track with multiple channels:** One track with multiple channels

When you specify `OneChannelPerTrack`, Elastic Transcoder creates a new track for every channel in your output. Your output can have up to eight single-channel tracks.

The outputs of `OneChannelPerTrack` for a specific channel value and inputs are as follows:

- * **0 channels with any input:** Audio omitted from the output
- * **1, 2, or auto channels with no audio input:** Audio omitted from the output
- * **1 channel with any input with audio:** One track with one channel, downmixed if necessary
- * **2 channels with one track with one channel:** Two tracks with one identical channel each
- * **2 or auto channels with two tracks with one channel each:** Two tracks with one channel each
- * **2 or auto channels with one track with two channels:** Two tracks with one channel each
- * **2 channels with one track with multiple channels:** Two tracks with one channel each
- * **auto channels with one track with one channel:** One track with one channel
- * **auto channels with one track with multiple channels:** Up to eight tracks with one channel each

When you specify `OneChannelPerTrackWithMosTo8Tracks`, Elastic Transcoder creates eight single-channel tracks for your output. All tracks that do not contain audio data from an input channel are MOS, or Mit Out Sound, tracks.

The outputs of `OneChannelPerTrackWithMosTo8Tracks` for a specific channel value and inputs are as follows:

- * **0 channels with any input:** Audio omitted from the output
- * **1, 2, or auto channels with no audio input:** Audio omitted from the output
- * **1 channel with any input with audio:** One track with one channel, downmixed if necessary, plus six MOS tracks
- * **2 channels with one track with one channel:** Two tracks with one identical channel each, plus six MOS tracks
- * **2 or auto channels with two tracks with one channel each:** Two tracks with one channel each, plus six MOS tracks
- * **2 or auto channels with one track with two channels:** Two tracks with one channel each, plus six MOS tracks
- * **2 channels with one track with multiple channels:** Two tracks with one channel each, plus six MOS tracks
- * **auto channels with one track with one channel:** One track with one channel, plus seven MOS tracks
- * **auto channels with one track with multiple channels:** Up to eight tracks with one channel each, plus MOS tracks until there are eight tracks in all

– **CodecOptions** (*dict*) –

If you specified AAC for `Audio:Codec`, this is the AAC compression profile to use. Valid values include:

`auto`, `AAC-LC`, `HE-AAC`, `HE-AACv2`

If you specify `auto`, Elastic Transcoder chooses a profile based on the bit rate of the output file.

- * **Profile** (*string*) –

You can only choose an audio profile when you specify AAC for the

value of `Audio:Codec`.

Specify the AAC profile for the output file. Elastic Transcoder supports the following profiles:

- `auto` : If you specify `auto`, Elastic Transcoder will select the profile based on the bit rate selected for the output file.
- `AAC-LC` : The most common AAC profile. Use for bit rates larger than 64 kbps.
- `HE-AAC` : Not supported on some older players and devices. Use for bit rates between 40 and 80 kbps.
- `HE-AACv2` : Not supported on some players and devices. Use for bit rates less than 48 kbps.

All outputs in a `Smooth` playlist must have the same value for `Profile`.

Note: If you created any presets before AAC profiles were added, Elastic Transcoder automatically updated your presets to use AAC-LC. You can change the value as required.

* **BitDepth** (*string*) –

You can only choose an audio bit depth when you specify `flac` or `pcm` for the value of `Audio:Codec`.

The bit depth of a sample is how many bits of information are included in the audio samples. The higher the bit depth, the better the audio, but the larger the file.

Valid values are `16` and `24`.

The most common bit depth is `24`.

* **BitOrder** (*string*) –

You can only choose an audio bit order when you specify `pcm` for the value of `Audio:Codec`.

The order the bits of a PCM sample are stored in.

The supported value is `LittleEndian`.

* **Signed** (*string*) –

You can only choose whether an audio sample is signed when you specify `pcm` for the value of `Audio:Codec`.

Whether audio samples are represented with negative and positive numbers (signed) or only positive numbers (unsigned).

The supported value is `Signed`.

- **Thumbnails** (*dict*) – A section of the request body that specifies the thumbnail parameters, if any.

- **Format** (*string*) –

The format of thumbnails, if any. Valid values are `jpg` and `png`.

You specify whether you want Elastic Transcoder to create thumbnails when you create a job.

- **Interval** (*string*) –

The approximate number of seconds between thumbnails. Specify an integer value.

– **Resolution** (*string*) –

Warning: To better control resolution and aspect ratio of thumbnails, we recommend that you use the values `MaxWidth`, `MaxHeight`, `SizingPolicy`, and `PaddingPolicy` instead of `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The width and height of thumbnail files in pixels. Specify a value in the format `*width* x *height*` where both values are even integers. The values cannot exceed the width and height that you specified in the `Video:Resolution` object.

– **AspectRatio** (*string*) –

Warning: To better control resolution and aspect ratio of thumbnails, we recommend that you use the values `MaxWidth`, `MaxHeight`, `SizingPolicy`, and `PaddingPolicy` instead of `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The aspect ratio of thumbnails. Valid values include:

`auto`, `1:1`, `4:3`, `3:2`, `16:9`

If you specify `auto`, Elastic Transcoder tries to preserve the aspect ratio of the video in the output file.

– **MaxWidth** (*string*) –

The maximum width of thumbnails in pixels. If you specify `auto`, Elastic Transcoder uses 1920 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 32 and 4096.

– **MaxHeight** (*string*) –

The maximum height of thumbnails in pixels. If you specify `auto`, Elastic Transcoder uses 1080 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 32 and 3072.

– **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of thumbnails:

- * `Fit`: Elastic Transcoder scales thumbnails so they match the value that you specified in thumbnail `MaxWidth` or `MaxHeight` settings without exceeding the other value.
- * `Fill`: Elastic Transcoder scales thumbnails so they match the value that you specified in thumbnail `MaxWidth` or `MaxHeight` settings and matches or exceeds the other value. Elastic Transcoder centers the image in thumbnails and then crops in the dimension (if any) that exceeds the maximum value.
- * `Stretch`: Elastic Transcoder stretches thumbnails to match the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings. If the relative proportions of the input video and thumbnails are different, the thumbnails will be distorted.
- * `Keep`: Elastic Transcoder does not scale thumbnails. If either dimension of the input video exceeds the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings, Elastic Transcoder crops the thumbnails.

- * `ShrinkToFit` : Elastic Transcoder scales thumbnails down so that their dimensions match the values that you specified for at least one of `thumbnail MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale thumbnails up.
- * `ShrinkToFill` : Elastic Transcoder scales thumbnails down so that their dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale thumbnails up.

– **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder may add black bars to the top and bottom and/or left and right sides of thumbnails to make the total size of the thumbnails match the values that you specified for `thumbnail MaxWidth` and `MaxHeight` settings.

Return type dict

Returns

Response Syntax

```
{
  'Preset': {
    'Id': 'string',
    'Arn': 'string',
    'Name': 'string',
    'Description': 'string',
    'Container': 'string',
    'Audio': {
      'Codec': 'string',
      'SampleRate': 'string',
      'BitRate': 'string',
      'Channels': 'string',
      'AudioPackingMode': 'string',
      'CodecOptions': {
        'Profile': 'string',
        'BitDepth': 'string',
        'BitOrder': 'string',
        'Signed': 'string'
      }
    },
  },
  'Video': {
    'Codec': 'string',
    'CodecOptions': {
      'string': 'string'
    },
    'KeyframesMaxDist': 'string',
    'FixedGOP': 'string',
    'BitRate': 'string',
    'FrameRate': 'string',
    'MaxFrameRate': 'string',
    'Resolution': 'string',
    'AspectRatio': 'string',
    'MaxWidth': 'string',
    'MaxHeight': 'string',
    'DisplayAspectRatio': 'string',
    'SizingPolicy': 'string',
    'PaddingPolicy': 'string',
  },
}
```



```

        'Watermarks': [
            {
                'Id': 'string',
                'MaxWidth': 'string',
                'MaxHeight': 'string',
                'SizingPolicy': 'string',
                'HorizontalAlign': 'string',
                'HorizontalOffset': 'string',
                'VerticalAlign': 'string',
                'VerticalOffset': 'string',
                'Opacity': 'string',
                'Target': 'string'
            },
        ],
    },
    'Thumbnails': {
        'Format': 'string',
        'Interval': 'string',
        'Resolution': 'string',
        'AspectRatio': 'string',
        'MaxWidth': 'string',
        'MaxHeight': 'string',
        'SizingPolicy': 'string',
        'PaddingPolicy': 'string'
    },
    'Type': 'string'
},
'Warning': 'string'
}

```

Response Structure

- (*dict*) –

The CreatePresetResponse structure.

- **Preset** (*dict*) –

A section of the response body that provides information about the preset that is created.

- * **Id** (*string*) –

Identifier for the new preset. You use this value to get settings for the preset or to delete it.

- * **Arn** (*string*) –

The Amazon Resource Name (ARN) for the preset.

- * **Name** (*string*) –

The name of the preset.

- * **Description** (*string*) –

A description of the preset.

- * **Container** (*string*) –

The container type for the output file. Valid values include `flac`, `flv`, `fmp4`, `gif`, `mp3`, `mp4`, `mpg`, `mxf`, `oga`, `ogg`, `ts`, and `webm`.

- * **Audio** (*dict*) –

A section of the response body that provides information about the audio preset values.

- **Codec** (*string*) –

The audio codec for the output file. Valid values include `aac`, `flac`, `mp2`, `mp3`, `pcm`, and `vorbis`.

- **SampleRate** (*string*) –

The sample rate of the audio stream in the output file, in Hertz. Valid values include:

`auto`, `22050`, `32000`, `44100`, `48000`, `96000`

If you specify `auto`, Elastic Transcoder automatically detects the sample rate.

- **BitRate** (*string*) –

The bit rate of the audio stream in the output file, in kilobits/second. Enter an integer between 64 and 320, inclusive.

- **Channels** (*string*) –

The number of audio channels in the output file. The following values are valid:

`auto`, `0`, `1`, `2`

One channel carries the information played by a single speaker. For example, a stereo track with two channels sends one channel to the left speaker, and the other channel to the right speaker. The output channels are organized into tracks. If you want Elastic Transcoder to automatically detect the number of audio channels in the input file and use that value for the output file, select `auto`.

The output of a specific channel value and inputs are as follows:

- `auto` **channel specified, with any input:** Pass through up to eight input channels.
- `0` **channels specified, with any input:** Audio omitted from the output.
- `1` **channel specified, with at least one input channel:** Mono sound.
- `2` **channels specified, with any input:** Two identical mono channels or stereo. For more information about tracks, see `Audio:AudioPackingMode`.

For more information about how Elastic Transcoder organizes channels and tracks, see `Audio:AudioPackingMode`.

- **AudioPackingMode** (*string*) –

The method of organizing audio channels and tracks. Use `Audio:Channels` to specify the number of channels in your output, and `Audio:AudioPackingMode` to specify the number of tracks and their relation to the channels. If you do not specify an `Audio:AudioPackingMode`, Elastic Transcoder uses `SingleTrack`.

The following values are valid:

`SingleTrack`, `OneChannelPerTrack`, and `OneChannelPerTrackWithMostTo8Tracks`

When you specify `SingleTrack`, Elastic Transcoder creates a single track for your output. The track can have up to eight channels. Use `SingleTrack` for all non-mxf containers.

The outputs of `SingleTrack` for a specific channel value and inputs are as follows:

- **0 channels with any input:** Audio omitted from the output
- **1, 2, or auto channels with no audio input:** Audio omitted from the output
- **1 channel with any input with audio:** One track with one channel, downmixed if necessary
- **2 channels with one track with one channel:** One track with two identical channels
- **2 or auto channels with two tracks with one channel each:** One track with two channels
- **2 or auto channels with one track with two channels:** One track with two channels
- **2 channels with one track with multiple channels:** One track with two channels
- **auto channels with one track with one channel:** One track with one channel
- **auto channels with one track with multiple channels:** One track with multiple channels

When you specify `OneChannelPerTrack`, Elastic Transcoder creates a new track for every channel in your output. Your output can have up to eight single-channel tracks.

The outputs of `OneChannelPerTrack` for a specific channel value and inputs are as follows:

- **0 channels with any input:** Audio omitted from the output
- **1, 2, or auto channels with no audio input:** Audio omitted from the output
- **1 channel with any input with audio:** One track with one channel, downmixed if necessary
- **2 channels with one track with one channel:** Two tracks with one identical channel each
- **2 or auto channels with two tracks with one channel each:** Two tracks with one channel each
- **2 or auto channels with one track with two channels:** Two tracks with one channel each
- **2 channels with one track with multiple channels:** Two tracks with one channel each
- **auto channels with one track with one channel:** One track with one channel
- **auto channels with one track with multiple channels:** Up to eight tracks with one channel each

When you specify `OneChannelPerTrackWithMosTo8Tracks`, Elastic Transcoder creates eight single-channel tracks for your output. All tracks that do not contain audio data from an input channel are MOS, or Mit Out Sound, tracks.

The outputs of `OneChannelPerTrackWithMosTo8Tracks` for a specific channel value and inputs are as follows:

- **0 channels with any input:** Audio omitted from the output

- `1`, `2`, or `auto` **channels with no audio input**: Audio omitted from the output
- `1` **channel with any input with audio**: One track with one channel, downmixed if necessary, plus six MOS tracks
- `2` **channels with one track with one channel**: Two tracks with one identical channel each, plus six MOS tracks
- `2` or `auto` **channels with two tracks with one channel each**: Two tracks with one channel each, plus six MOS tracks
- `2` or `auto` **channels with one track with two channels**: Two tracks with one channel each, plus six MOS tracks
- `2` **channels with one track with multiple channels**: Two tracks with one channel each, plus six MOS tracks
- `auto` **channels with one track with one channel**: One track with one channel, plus seven MOS tracks
- `auto` **channels with one track with multiple channels**: Up to eight tracks with one channel each, plus MOS tracks until there are eight tracks in all
- **CodecOptions** (*dict*) –

If you specified AAC for `Audio:Codec`, this is the AAC compression profile to use. Valid values include:

`auto`, `AAC-LC`, `HE-AAC`, `HE-AACv2`

If you specify `auto`, Elastic Transcoder chooses a profile based on the bit rate of the output file.

- **Profile** (*string*) –

You can only choose an audio profile when you specify AAC for the value of `Audio:Codec`.

Specify the AAC profile for the output file. Elastic Transcoder supports the following profiles:

- `auto`: If you specify `auto`, Elastic Transcoder will select the profile based on the bit rate selected for the output file.
- `AAC-LC`: The most common AAC profile. Use for bit rates larger than 64 kbps.
- `HE-AAC`: Not supported on some older players and devices. Use for bit rates between 40 and 80 kbps.
- `HE-AACv2`: Not supported on some players and devices. Use for bit rates less than 48 kbps.

All outputs in a `Smooth` playlist must have the same value for `Profile`.

Note: If you created any presets before AAC profiles were added, Elastic Transcoder automatically updated your presets to use AAC-LC. You can change the value as required.

- **BitDepth** (*string*) –

You can only choose an audio bit depth when you specify `flac` or `pcm` for the value of `Audio:Codec`.

The bit depth of a sample is how many bits of information are included in the audio samples. The higher the bit depth, the better the audio, but the larger the file.

Valid values are `16` and `24`.

The most common bit depth is 24 .

- **BitOrder** (*string*) –

You can only choose an audio bit order when you specify `pcm` for the value of `Audio:Codec`.

The order the bits of a PCM sample are stored in.

The supported value is `LittleEndian` .

- **Signed** (*string*) –

You can only choose whether an audio sample is signed when you specify `pcm` for the value of `Audio:Codec`.

Whether audio samples are represented with negative and positive numbers (signed) or only positive numbers (unsigned).

The supported value is `Signed` .

- * **Video** (*dict*) –

A section of the response body that provides information about the video preset values.

- **Codec** (*string*) –

The video codec for the output file. Valid values include `gif` , `H.264` , `mpeg2` , and `vp8` . You can only specify `vp8` when the container type is `webm` , `gif` when the container type is `gif` , and `mpeg2` when the container type is `mpg` .

- **CodecOptions** (*dict*) –

- Profile (H.264/VP8 Only)**

The H.264 profile that you want to use for the output file. Elastic Transcoder supports the following profiles:

- `baseline` : The profile most commonly used for videoconferencing and for mobile applications.
- `main` : The profile used for standard-definition digital TV broadcasts.
- `high` : The profile used for high-definition digital TV broadcasts and for Blu-ray discs.

- Level (H.264 Only)**

The H.264 level that you want to use for the output file. Elastic Transcoder supports the following levels:

`1` , `1b` , `1.1` , `1.2` , `1.3` , `2` , `2.1` , `2.2` , `3` , `3.1` , `3.2` , `4` , `4.1`

- MaxReferenceFrames (H.264 Only)**

Applicable only when the value of `Video:Codec` is `H.264`. The maximum number of previously decoded frames to use as a reference for decoding future frames. Valid values are integers 0 through 16, but we recommend that you not use a value greater than the following:

```
Min(Floor(Maximum decoded
picture buffer in macroblocks *
256 / (Width in pixels * Height
in pixels)), 16)
```

where *Width in pixels* and *Height in pixels* represent either `MaxWidth` and `MaxHeight`, or `Resolution`. *Maximum de-*

coded picture buffer in macroblocks depends on the value of the `Level` object. See the list below. (A macroblock is a block of pixels measuring 16x16.)

- 1 - 396
- 1b - 396
- 1.1 - 900
- 1.2 - 2376
- 1.3 - 2376
- 2 - 2376
- 2.1 - 4752
- 2.2 - 8100
- 3 - 8100
- 3.1 - 18000
- 3.2 - 20480
- 4 - 32768
- 4.1 - 32768

MaxBitRate (Optional, H.264/MPEG2/VP8 only)

The maximum number of bits per second in a video buffer; the size of the buffer is specified by `BufferSize`. Specify a value between 16 and 62,500. You can reduce the bandwidth required to stream a video by reducing the maximum bit rate, but this also reduces the quality of the video.

BufferSize (Optional, H.264/MPEG2/VP8 only)

The maximum number of bits in any *x* seconds of the output video. This window is commonly 10 seconds, the standard segment duration when you're using FMP4 or MPEG-TS for the container type of the output video. Specify an integer greater than 0. If you specify `MaxBitRate` and omit `BufferSize`, Elastic Transcoder sets `BufferSize` to 10 times the value of `MaxBitRate`.

InterlacedMode (Optional, H.264/MPEG2 Only)

The interlace mode for the output video.

Interlaced video is used to double the perceived frame rate for a video by interlacing two fields (one field on every other line, the other field on the other lines) so that the human eye registers multiple pictures per frame. Interlacing reduces the bandwidth required for transmitting a video, but can result in blurred images and flickering.

Valid values include `Progressive` (no interlacing, top to bottom), `TopFirst` (top field first), `BottomFirst` (bottom field first), and `Auto`.

If `InterlaceMode` is not specified, Elastic Transcoder uses `Progressive` for the output. If `Auto` is specified, Elastic Transcoder interlaces the output.

ColorSpaceConversionMode (Optional, H.264/MPEG2 Only)

The color space conversion Elastic Transcoder applies to the output video. Color spaces are the algorithms used by the

computer to store information about how to render color. Bt.601 is the standard for standard definition video, while Bt.709 is the standard for high definition video.

Valid values include `None`, `Bt709toBt601`, `Bt601toBt709`, and `Auto`.

If you chose `Auto` for `ColorSpaceConversionMode` and your output is interlaced, your frame rate is one of 23.97, 24, 25, 29.97, 50, or 60, your `SegmentDuration` is null, and you are using one of the resolution changes from the list below, Elastic Transcoder applies the following color space conversions:

- *Standard to HD, 720x480 to 1920x1080* - Elastic Transcoder applies `Bt601ToBt709`
- *Standard to HD, 720x576 to 1920x1080* - Elastic Transcoder applies `Bt601ToBt709`
- *HD to Standard, 1920x1080 to 720x480* - Elastic Transcoder applies `Bt709ToBt601`
- *HD to Standard, 1920x1080 to 720x576* - Elastic Transcoder applies `Bt709ToBt601`

Note: Elastic Transcoder may change the behavior of the `ColorspaceConversionMode` `Auto` mode in the future. All outputs in a playlist must use the same `ColorSpaceConversionMode`.

If you do not specify a `ColorSpaceConversionMode`, Elastic Transcoder does not change the color space of a file. If you are unsure what `ColorSpaceConversionMode` was applied to your output file, you can check the `AppliedColorSpaceConversion` parameter included in your job response. If your job does not have an `AppliedColorSpaceConversion` in its response, no `ColorSpaceConversionMode` was applied.

ChromaSubsampling

The sampling pattern for the chroma (color) channels of the output video. Valid values include `yuv420p` and `yuv422p`.

`yuv420p` samples the chroma information of every other horizontal and every other vertical line, `yuv422p` samples the color information of every horizontal line and every other vertical line.

LoopCount (Gif Only)

The number of times you want the output gif to loop. Valid values include `Infinite` and integers between 0 and 100, inclusive.

- *(string)* –
- *(string)* –
- **KeyframesMaxDist** *(string)* –

Applicable only when the value of `Video:Codec` is one of `H.264`, `MPEG2`, or `VP8`.

The maximum number of frames between key frames. Key frames are fully encoded frames; the frames between key

frames are encoded based, in part, on the content of the key frames. The value is an integer formatted as a string; valid values are between 1 (every frame is a key frame) and 100000, inclusive. A higher value results in higher compression but may also discernibly decrease video quality.

For Smooth outputs, the `FrameRate` must have a constant ratio to the `KeyframesMaxDist`. This allows Smooth playlists to switch between different quality levels while the file is being played.

For example, an input file can have a `FrameRate` of 30 with a `KeyframesMaxDist` of 90. The output file then needs to have a ratio of 1:3. Valid outputs would have `FrameRate` of 30, 25, and 10, and `KeyframesMaxDist` of 90, 75, and 30, respectively.

Alternately, this can be achieved by setting `FrameRate` to `auto` and having the same values for `MaxFrameRate` and `KeyframesMaxDist`.

- **FixedGOP** (*string*) –

Applicable only when the value of `Video:Codec` is one of `H.264`, `MPEG2`, or `VP8`.

Whether to use a fixed value for `FixedGOP`. Valid values are `true` and `false`:

- `true`: Elastic Transcoder uses the value of `KeyframesMaxDist` for the distance between key frames (the number of frames in a group of pictures, or GOP).
- `false`: The distance between key frames can vary.

Warning: `FixedGOP` must be set to `true` for `h264` containers.

- **BitRate** (*string*) –

The bit rate of the video stream in the output file, in kilobits/second. Valid values depend on the values of `Level` and `Profile`. If you specify `auto`, Elastic Transcoder uses the detected bit rate of the input source. If you specify a value other than `auto`, we recommend that you specify a value less than or equal to the maximum H.264-compliant value listed for your level and profile:

*Level - Maximum video bit rate in kilobits/second
(baseline and main Profile) : maximum video bit
rate in kilobits/second (high Profile)*

- 1 - 64 : 80
- 1b - 128 : 160
- 1.1 - 192 : 240
- 1.2 - 384 : 480
- 1.3 - 768 : 960
- 2 - 2000 : 2500
- 3 - 10000 : 12500
- 3.1 - 14000 : 17500
- 3.2 - 20000 : 25000
- 4 - 20000 : 25000

- 4.1 - 50000 : 62500
- **FrameRate** (*string*) –

The frames per second for the video stream in the output file. Valid values include:

`auto`, `10`, `15`, `23.97`, `24`, `25`, `29.97`, `30`, `60`

If you specify `auto`, Elastic Transcoder uses the detected frame rate of the input source. If you specify a frame rate, we recommend that you perform the following calculation:

$$\text{Frame rate} = \frac{\text{maximum recommended decoding speed in luma samples/second}}{(\text{width in pixels} * \text{height in pixels})}$$

where:

- *width in pixels* and *height in pixels* represent the Resolution of the output video.
- *maximum recommended decoding speed in Luma samples/second* is less than or equal to the maximum value listed in the following table, based on the value that you specified for Level.

The maximum recommended decoding speed in Luma samples/second for each level is described in the following list (*Level - Decoding speed*):

- 1 - 380160
- 1b - 380160
- 1.1 - 76800
- 1.2 - 1536000
- 1.3 - 3041280
- 2 - 3041280
- 2.1 - 5068800
- 2.2 - 5184000
- 3 - 10368000
- 3.1 - 27648000
- 3.2 - 55296000
- 4 - 62914560
- 4.1 - 62914560
- **MaxFrameRate** (*string*) –

If you specify `auto` for `FrameRate`, Elastic Transcoder uses the frame rate of the input video for the frame rate of the output video. Specify the maximum frame rate that you want Elastic Transcoder to use when the frame rate of the input video is greater than the desired maximum frame rate of the output video. Valid values include: `10`, `15`, `23.97`, `24`, `25`, `29.97`, `30`, `60`.

- **Resolution** (*string*) –

Warning: To better control resolution and aspect ratio of output videos, we recommend that you use the values `MaxWidth`, `MaxHeight`, `SizingPolicy`, `PaddingPolicy`, and `DisplayAspectRatio` instead of `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The width and height of the video in the output file, in pixels. Valid values are `auto` and `width x height` :

- `auto` : Elastic Transcoder attempts to preserve the width and height of the input file, subject to the following rules.
- `*width* x *height*` : The width and height of the output video in pixels.

Note the following about specifying the width and height:

- The width must be an even integer between 128 and 4096, inclusive.
- The height must be an even integer between 96 and 3072, inclusive.
- If you specify a resolution that is less than the resolution of the input file, Elastic Transcoder rescales the output file to the lower resolution.
- If you specify a resolution that is greater than the resolution of the input file, Elastic Transcoder rescales the output to the higher resolution.
- We recommend that you specify a resolution for which the product of width and height is less than or equal to the applicable value in the following list (*List - Max width x height value*) :
- 1 - 25344
- 1b - 25344
- 1.1 - 101376
- 1.2 - 101376
- 1.3 - 101376
- 2 - 101376
- 2.1 - 202752
- 2.2 - 404720
- 3 - 404720
- 3.1 - 921600
- 3.2 - 1310720
- 4 - 2097152
- 4.1 - 2097152
- **AspectRatio** (*string*) –

Warning: To better control resolution and aspect ratio of output videos, we recommend that you use the values `MaxWidth` , `MaxHeight` , `SizingPolicy` , `PaddingPolicy` , and `DisplayAspectRatio` instead of `Resolution` and `AspectRatio` . The two groups of settings are mutually exclusive. Do not use them together.

The display aspect ratio of the video in the output file. Valid values include:

`auto` , `1:1` , `4:3` , `3:2` , `16:9`

If you specify `auto` , Elastic Transcoder tries to preserve the aspect ratio of the input file.

If you specify an aspect ratio for the output file that differs from aspect ratio of the input file, Elastic Transcoder adds

pillarboxing (black bars on the sides) or letterboxing (black bars on the top and bottom) to maintain the aspect ratio of the active region of the video.

- **MaxWidth** (*string*) –

The maximum width of the output video in pixels. If you specify `auto`, Elastic Transcoder uses 1920 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 128 and 4096.

- **MaxHeight** (*string*) –

The maximum height of the output video in pixels. If you specify `auto`, Elastic Transcoder uses 1080 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 96 and 3072.

- **DisplayAspectRatio** (*string*) –

The value that Elastic Transcoder adds to the metadata in the output file.

- **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of the output video:

- **Fit** : Elastic Transcoder scales the output video so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Fill** : Elastic Transcoder scales the output video so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output video and then crops it in the dimension (if any) that exceeds the maximum value.
- **Stretch** : Elastic Transcoder stretches the output video to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input video and the output video are different, the output video will be distorted.
- **Keep** : Elastic Transcoder does not scale the output video. If either dimension of the input video exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output video.
- **ShrinkToFit** : Elastic Transcoder scales the output video down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the video up.
- **ShrinkToFill** : Elastic Transcoder scales the output video down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the video up.
- **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder may add black bars to the top and bottom and/or left and right sides of the output video to make the total size of the output video match the values that you specified for `MaxWidth` and

`MaxHeight`.

- **Watermarks** (*list*) –

Settings for the size, location, and opacity of graphics that you want Elastic Transcoder to overlay over videos that are transcoded using this preset. You can specify settings for up to four watermarks. Watermarks appear in the specified size and location, and with the specified opacity for the duration of the transcoded video.

Watermarks can be in .png or .jpg format. If you want to display a watermark that is not rectangular, use the .png format, which supports transparency.

When you create a job that uses this preset, you specify the .png or .jpg graphics that you want Elastic Transcoder to include in the transcoded videos. You can specify fewer graphics in the job than you specify watermark settings in the preset, which allows you to use the same preset for up to four watermarks that have different dimensions.

- (*dict*) –

Settings for the size, location, and opacity of graphics that you want Elastic Transcoder to overlay over videos that are transcoded using this preset. You can specify settings for up to four watermarks. Watermarks appear in the specified size and location, and with the specified opacity for the duration of the transcoded video.

Watermarks can be in .png or .jpg format. If you want to display a watermark that is not rectangular, use the .png format, which supports transparency.

When you create a job that uses this preset, you specify the .png or .jpg graphics that you want Elastic Transcoder to include in the transcoded videos. You can specify fewer graphics in the job than you specify watermark settings in the preset, which allows you to use the same preset for up to four watermarks that have different dimensions.

- **Id** (*string*) – A unique identifier for the settings for one watermark. The value of `Id` can be up to 40 characters long.

- **MaxWidth** (*string*) –

The maximum width of the watermark in one of the following formats:

- number of pixels (px): The minimum value is 16 pixels, and the maximum value is the value of `MaxWidth`.
- integer percentage (%): The range of valid values is 0 to 100. Use the value of `Target` to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the calculation.

If you specify the value in pixels, it must be less than or equal to the value of `MaxWidth`.

- **MaxHeight** (*string*) –

The maximum height of the watermark in one of the following formats:

- number of pixels (px): The minimum value is 16 pixels, and the maximum value is the value of `MaxHeight` .
- integer percentage (%): The range of valid values is 0 to 100. Use the value of `Target` to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the calculation. If you specify the value in pixels, it must be less than or equal to the value of `MaxHeight` .
- **SizingPolicy** (*string*) –

A value that controls scaling of the watermark:

- **Fit** : Elastic Transcoder scales the watermark so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Stretch** : Elastic Transcoder stretches the watermark to match the values that you specified for `MaxWidth` and `MaxHeight` . If the relative proportions of the watermark and the values of `MaxWidth` and `MaxHeight` are different, the watermark will be distorted.
- **ShrinkToFit** : Elastic Transcoder scales the watermark down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the watermark up.
- **HorizontalAlign** (*string*) –

The horizontal position of the watermark unless you specify a non-zero value for `HorizontalOffset` :

- **Left** : The left edge of the watermark is aligned with the left border of the video.
- **Right** : The right edge of the watermark is aligned with the right border of the video.
- **Center** : The watermark is centered between the left and right borders.
- **HorizontalOffset** (*string*) –

The amount by which you want the horizontal position of the watermark to be offset from the position specified by `HorizontalAlign`:

- number of pixels (px): The minimum value is 0 pixels, and the maximum value is the value of `MaxWidth`.
- integer percentage (%): The range of valid values is 0 to 100. For example, if you specify `Left` for `HorizontalAlign` and 5px for `HorizontalOffset` , the left side of the watermark appears 5 pixels from the left border of the output video.

`HorizontalOffset` is only valid when the value of `HorizontalAlign` is `Left` or `Right` . If you specify an offset that causes the watermark to extend beyond the left or right border and Elastic Transcoder has not added black bars, the watermark is cropped. If Elastic Transcoder has added black bars, the watermark extends into the black bars. If the watermark extends beyond the black bars, it is cropped.

Use the value of `Target` to specify whether you want to in-

clude the black bars that are added by Elastic Transcoder, if any, in the offset calculation.

- **VerticalAlign** (*string*) –

The vertical position of the watermark unless you specify a non-zero value for `VerticalOffset` :

- **Top** : The top edge of the watermark is aligned with the top border of the video.
- **Bottom** : The bottom edge of the watermark is aligned with the bottom border of the video.
- **Center** : The watermark is centered between the top and bottom borders.
- **VerticalOffset** (*string*) – `VerticalOffset`

The amount by which you want the vertical position of the watermark to be offset from the position specified by `VerticalAlign`:

- number of pixels (px): The minimum value is 0 pixels, and the maximum value is the value of `MaxHeight` .
- integer percentage (%): The range of valid values is 0 to 100. For example, if you specify `Top` for `VerticalAlign` and `5px` for `VerticalOffset` , the top of the watermark appears 5 pixels from the top border of the output video.

`VerticalOffset` is only valid when the value of `VerticalAlign` is `Top` or `Bottom`.

If you specify an offset that causes the watermark to extend beyond the top or bottom border and Elastic Transcoder has not added black bars, the watermark is cropped. If Elastic Transcoder has added black bars, the watermark extends into the black bars. If the watermark extends beyond the black bars, it is cropped.

Use the value of `Target` to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the offset calculation.

- **Opacity** (*string*) –

A percentage that indicates how much you want a watermark to obscure the video in the location where it appears. Valid values are 0 (the watermark is invisible) to 100 (the watermark completely obscures the video in the specified location). The datatype of `Opacity` is float.

Elastic Transcoder supports transparent .png graphics. If you use a transparent .png, the transparent portion of the video appears as if you had specified a value of 0 for `Opacity` . The .jpg file format doesn't support transparency.

- **Target** (*string*) –

A value that determines how Elastic Transcoder interprets values that you specified for `HorizontalOffset` , `VerticalOffset` , `MaxWidth` , and `MaxHeight` :

- **Content** : `HorizontalOffset` and `VerticalOffset` values are calculated based on the borders of the video excluding black bars added by Elastic Transcoder, if any. In addition, `MaxWidth` and `MaxHeight` , if specified as a per-

centage, are calculated based on the borders of the video excluding black bars added by Elastic Transcoder, if any.

- **Frame** : `HorizontalOffset` and `VerticalOffset` values are calculated based on the borders of the video including black bars added by Elastic Transcoder, if any.

In addition, `MaxWidth` and `MaxHeight` , if specified as a percentage, are calculated based on the borders of the video including black bars added by Elastic Transcoder, if any.

* **Thumbnails** (*dict*) –

A section of the response body that provides information about the thumbnail preset values, if any.

- **Format** (*string*) –

The format of thumbnails, if any. Valid values are `jpg` and `png` .

You specify whether you want Elastic Transcoder to create thumbnails when you create a job.

- **Interval** (*string*) –

The approximate number of seconds between thumbnails. Specify an integer value.

- **Resolution** (*string*) –

Warning: To better control resolution and aspect ratio of thumbnails, we recommend that you use the values `MaxWidth` , `MaxHeight` , `SizingPolicy` , and `PaddingPolicy` instead of `Resolution` and `AspectRatio` . The two groups of settings are mutually exclusive. Do not use them together.

The width and height of thumbnail files in pixels. Specify a value in the format `*width* x *height*` where both values are even integers. The values cannot exceed the width and height that you specified in the `Video:Resolution` object.

- **AspectRatio** (*string*) –

Warning: To better control resolution and aspect ratio of thumbnails, we recommend that you use the values `MaxWidth` , `MaxHeight` , `SizingPolicy` , and `PaddingPolicy` instead of `Resolution` and `AspectRatio` . The two groups of settings are mutually exclusive. Do not use them together.

The aspect ratio of thumbnails. Valid values include:

`auto` , `1:1` , `4:3` , `3:2` , `16:9`

If you specify `auto` , Elastic Transcoder tries to preserve the aspect ratio of the video in the output file.

- **MaxWidth** (*string*) –

The maximum width of thumbnails in pixels. If you specify `auto`, Elastic Transcoder uses 1920 (Full HD) as the default

value. If you specify a numeric value, enter an even integer between 32 and 4096.

- **MaxHeight** (*string*) –

The maximum height of thumbnails in pixels. If you specify auto, Elastic Transcoder uses 1080 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 32 and 3072.

- **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of thumbnails:

- **Fit** : Elastic Transcoder scales thumbnails so they match the value that you specified in thumbnail `MaxWidth` or `MaxHeight` settings without exceeding the other value.
- **Fill** : Elastic Transcoder scales thumbnails so they match the value that you specified in thumbnail `MaxWidth` or `MaxHeight` settings and matches or exceeds the other value. Elastic Transcoder centers the image in thumbnails and then crops in the dimension (if any) that exceeds the maximum value.
- **Stretch** : Elastic Transcoder stretches thumbnails to match the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings. If the relative proportions of the input video and thumbnails are different, the thumbnails will be distorted.
- **Keep** : Elastic Transcoder does not scale thumbnails. If either dimension of the input video exceeds the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings, Elastic Transcoder crops the thumbnails.
- **ShrinkToFit** : Elastic Transcoder scales thumbnails down so that their dimensions match the values that you specified for at least one of thumbnail `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale thumbnails up.
- **ShrinkToFill** : Elastic Transcoder scales thumbnails down so that their dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale thumbnails up.

- **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder may add black bars to the top and bottom and/or left and right sides of thumbnails to make the total size of the thumbnails match the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings.

- * **Type** (*string*) –

Whether the preset is a default preset provided by Elastic Transcoder (`System`) or a preset that you have defined (`Custom`).

- **Warning** (*string*) –

If the preset settings don't comply with the standards for the video codec but Elastic Transcoder created the preset, this message explains the reason the preset settings don't meet the standard. Elastic Transcoder created the

preset because the settings might produce acceptable output.

delete_pipeline (***kwargs*)

The DeletePipeline operation removes a pipeline.

You can only delete a pipeline that has never been used or that is not currently in use (doesn't contain any active jobs). If the pipeline is currently in use, DeletePipeline returns an error.

Request Syntax

```
response = client.delete_pipeline(
    Id='string'
)
```

Parameters *Id* (*string*) – [REQUIRED]

The identifier of the pipeline that you want to delete.

Return type dict

Returns

Response Syntax

```
{}
```

Response Structure

- (*dict*) –

The DeletePipelineResponse structure.

delete_preset (***kwargs*)

The DeletePreset operation removes a preset that you've added in an AWS region.

Note: You can't delete the default presets that are included with Elastic Transcoder.

Request Syntax

```
response = client.delete_preset(
    Id='string'
)
```

Parameters *Id* (*string*) – [REQUIRED]

The identifier of the preset for which you want to get detailed information.

Return type dict

Returns

Response Syntax

```
{}
```

Response Structure

- (*dict*) –

The DeletePresetResponse structure.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to *ClientMethod*.

- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_waiter (*waiter_name*)

list_jobs_by_pipeline (***kwargs*)

The `ListJobsByPipeline` operation gets a list of the jobs currently in a pipeline.

Elastic Transcoder returns all of the jobs currently in the specified pipeline. The response body contains one element for each job that satisfies the search criteria.

Request Syntax

```
response = client.list_jobs_by_pipeline(
    PipelineId='string',
    Ascending='string',
    PageToken='string'
)
```

Parameters

- **PipelineId** (*string*) – [REQUIRED]

The ID of the pipeline for which you want to get job information.

- **Ascending** (*string*) – To list jobs in chronological order by the date and time that they were submitted, enter `true`. To list jobs in reverse chronological order, enter `false`.
- **PageToken** (*string*) – When Elastic Transcoder returns more than one page of results, use `pageToken` in subsequent GET requests to get each successive page of results.

Return type `dict`

Returns

Response Syntax

```
{
  'Jobs': [
    {
      'Id': 'string',
      'Arn': 'string',
      'PipelineId': 'string',
      'Input': {
        'Key': 'string',
        'FrameRate': 'string',
```

```

'Resolution': 'string',
'AspectRatio': 'string',
'Interlaced': 'string',
'Container': 'string',
'Encryption': {
    'Mode': 'string',
    'Key': 'string',
    'KeyMd5': 'string',
    'InitializationVector': 'string'
},
'DetectedProperties': {
    'Width': 123,
    'Height': 123,
    'FrameRate': 'string',
    'FileSize': 123,
    'DurationMillis': 123
}
},
'Output': {
    'Id': 'string',
    'Key': 'string',
    'ThumbnailPattern': 'string',
    'ThumbnailEncryption': {
        'Mode': 'string',
        'Key': 'string',
        'KeyMd5': 'string',
        'InitializationVector': 'string'
    },
    'Rotate': 'string',
    'PresetId': 'string',
    'SegmentDuration': 'string',
    'Status': 'string',
    'StatusDetail': 'string',
    'Duration': 123,
    'Width': 123,
    'Height': 123,
    'FrameRate': 'string',
    'FileSize': 123,
    'DurationMillis': 123,
    'Watermarks': [
        {
            'PresetWatermarkId': 'string',
            'InputKey': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        }
    ],
    'AlbumArt': {
        'MergePolicy': 'string',
        'Artwork': [
            {
                'InputKey': 'string',
                'MaxWidth': 'string',
                'MaxHeight': 'string',

```

```
        'SizingPolicy': 'string',
        'PaddingPolicy': 'string',
        'AlbumArtFormat': 'string',
        'Encryption': {
            'Mode': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'InitializationVector': 'string'
        }
    },
]
},
'Composition': [
    {
        'TimeSpan': {
            'StartTime': 'string',
            'Duration': 'string'
        }
    },
],
'Captions': {
    'MergePolicy': 'string',
    'CaptionSources': [
        {
            'Key': 'string',
            'Language': 'string',
            'TimeOffset': 'string',
            'Label': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        }
    ],
    'CaptionFormats': [
        {
            'Format': 'string',
            'Pattern': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        }
    ],
    'Encryption': {
        'Mode': 'string',
        'Key': 'string',
        'KeyMd5': 'string',
        'InitializationVector': 'string'
    },
    'AppliedColorSpaceConversion': 'string'
},
'Outputs': [
```

```

{
    'Id': 'string',
    'Key': 'string',
    'ThumbnailPattern': 'string',
    'ThumbnailEncryption': {
        'Mode': 'string',
        'Key': 'string',
        'KeyMd5': 'string',
        'InitializationVector': 'string'
    },
    'Rotate': 'string',
    'PresetId': 'string',
    'SegmentDuration': 'string',
    'Status': 'string',
    'StatusDetail': 'string',
    'Duration': 123,
    'Width': 123,
    'Height': 123,
    'FrameRate': 'string',
    'FileSize': 123,
    'DurationMillis': 123,
    'Watermarks': [
        {
            'PresetWatermarkId': 'string',
            'InputKey': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        },
    ],
    'AlbumArt': {
        'MergePolicy': 'string',
        'Artwork': [
            {
                'InputKey': 'string',
                'MaxWidth': 'string',
                'MaxHeight': 'string',
                'SizingPolicy': 'string',
                'PaddingPolicy': 'string',
                'AlbumArtFormat': 'string',
                'Encryption': {
                    'Mode': 'string',
                    'Key': 'string',
                    'KeyMd5': 'string',
                    'InitializationVector': 'string'
                }
            },
        ],
    },
    'Composition': [
        {
            'TimeSpan': {
                'StartTime': 'string',
                'Duration': 'string'
            }
        }
    ]
}

```

```
    },
  ],
  'Captions': {
    'MergePolicy': 'string',
    'CaptionSources': [
      {
        'Key': 'string',
        'Language': 'string',
        'TimeOffset': 'string',
        'Label': 'string',
        'Encryption': {
          'Mode': 'string',
          'Key': 'string',
          'KeyMd5': 'string',
          'InitializationVector': 'string'
        }
      }
    ],
  },
  'CaptionFormats': [
    {
      'Format': 'string',
      'Pattern': 'string',
      'Encryption': {
        'Mode': 'string',
        'Key': 'string',
        'KeyMd5': 'string',
        'InitializationVector': 'string'
      }
    }
  ],
  'Encryption': {
    'Mode': 'string',
    'Key': 'string',
    'KeyMd5': 'string',
    'InitializationVector': 'string'
  },
  'AppliedColorSpaceConversion': 'string'
},
],
'OutputKeyPrefix': 'string',
'Playlists': [
  {
    'Name': 'string',
    'Format': 'string',
    'OutputKeys': [
      'string',
    ],
  },
  'HlsContentProtection': {
    'Method': 'string',
    'Key': 'string',
    'KeyMd5': 'string',
    'InitializationVector': 'string',
    'LicenseAcquisitionUrl': 'string',
    'KeyStoragePolicy': 'string'
  },
  'PlayReadyDrm': {
    'Format': 'string',
  },
]
```

```

        'Key': 'string',
        'KeyMd5': 'string',
        'KeyId': 'string',
        'InitializationVector': 'string',
        'LicenseAcquisitionUrl': 'string'
    },
    'Status': 'string',
    'StatusDetail': 'string'
},
],
'Status': 'string',
'UserMetadata': {
    'string': 'string'
},
'Timing': {
    'SubmitTimeMillis': 123,
    'StartTimeMillis': 123,
    'FinishTimeMillis': 123
}
},
],
'NextPageToken': 'string'
}

```

Response Structure

- (*dict*) –

The `ListJobsByPipelineResponse` structure.

- **Jobs** (*list*) –

An array of `Job` objects that are in the specified pipeline.

- * (*dict*) –

A section of the response body that provides information about the job that is created.

- **Id** (*string*) –

The identifier that Elastic Transcoder assigned to the job. You use this value to get settings for the job or to delete the job.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) for the job.

- **PipelineId** (*string*) –

The `Id` of the pipeline that you want Elastic Transcoder to use for transcoding. The pipeline determines several settings, including the Amazon S3 bucket from which Elastic Transcoder gets the files to transcode and the bucket into which Elastic Transcoder puts the transcoded files.

- **Input** (*dict*) –

A section of the request or response body that provides information about the file that is being transcoded.

- **Key** (*string*) –

The name of the file to transcode. Elsewhere in the body of the JSON block is the the ID of the pipeline to use for processing the job. The `InputBucket` object in that pipeline tells Elastic Transcoder which Amazon S3 bucket to get the file from.

If the file name includes a prefix, such as `cooking/lasagna.mpg`, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

- **FrameRate** (*string*) –

The frame rate of the input file. If you want Elastic Transcoder to automatically detect the frame rate of the input file, specify `auto`. If you want to specify the frame rate for the input file, enter one of the following values:

`10, 15, 23.97, 24, 25, 29.97, 30, 60`

If you specify a value other than `auto`, Elastic Transcoder disables automatic detection of the frame rate.

- **Resolution** (*string*) –

This value must be `auto`, which causes Elastic Transcoder to automatically detect the resolution of the input file.

- **AspectRatio** (*string*) –

The aspect ratio of the input file. If you want Elastic Transcoder to automatically detect the aspect ratio of the input file, specify `auto`. If you want to specify the aspect ratio for the output file, enter one of the following values:

`1:1, 4:3, 3:2, 16:9`

If you specify a value other than `auto`, Elastic Transcoder disables automatic detection of the aspect ratio.

- **Interlaced** (*string*) –

Whether the input file is interlaced. If you want Elastic Transcoder to automatically detect whether the input file is interlaced, specify `auto`. If you want to specify whether the input file is interlaced, enter one of the following values:

`true, false`

If you specify a value other than `auto`, Elastic Transcoder disables automatic detection of interlacing.

- **Container** (*string*) –

The container type for the input file. If you want Elastic Transcoder to automatically detect the container type of the input file, specify `auto`. If you want to specify the container type for the input file, enter one of the following values:

`3gp, aac, asf, avi, divx, flv, m4a, mkv, mov, mp3, mp4, mpeg, mpeg-ps, mpeg-ts, mxf, ogg, vob, wav, webm`

- **Encryption** (*dict*) –

The encryption settings, if any, that are used for decrypting your input files. If your input file is encrypted, you must specify the mode that Elastic Transcoder will use to decrypt your file.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **DetectedProperties** (*dict*) –

The detected properties of the input file.

- **Width** (*integer*) –

The detected width of the input file, in pixels.

- **Height** (*integer*) –

The detected height of the input file, in pixels.

- **FrameRate** (*string*) –

The detected frame rate of the input file, in frames per second.

- **FileSize** (*integer*) –

The detected file size of the input file, in bytes.

- **DurationMillis** (*integer*) –

The detected duration of the input file, in milliseconds.

- **Output** (*dict*) –

If you specified one output for a job, information about that output. If you specified multiple outputs for a job, the Output object lists information about the first output. This duplicates the information that is listed for the first output in the Outputs object.

Warning: Outputs recommended instead.

A section of the request or response body that provides information about the transcoded (target) file.

- **Id** (*string*) –

A sequential counter, starting with 1, that identifies an output among the outputs from the current job. In the Output syntax, this value is always 1.

- **Key** (*string*) –

The name to assign to the transcoded file. Elastic Transcoder saves the file in the Amazon S3 bucket specified by the OutputBucket object in the pipeline that is specified by the pipeline ID.

- **ThumbnailPattern** (*string*) –

Whether you want Elastic Transcoder to create thumbnails for your videos and, if so, how you want Elastic Transcoder to name the files.

If you don't want Elastic Transcoder to create thumbnails, specify "".

If you do want Elastic Transcoder to create thumbnails, specify the information that you want to include in the file name for each thumbnail. You can specify the following values in any sequence:

- **“{count}” (Required)** : If you want to create thumbnails, you must include {count} in the ThumbnailPattern object. Wherever you specify {count}, Elastic Transcoder adds a five-digit sequence number (beginning with **00001**) to thumbnail file names. The number indicates where a given thumbnail appears in the sequence of thumbnails for a transcoded file. .. warning::If you specify a literal value and/or {resolution} but you omit {count}, Elastic

Transcoder returns a validation error and does not create the job.

- **Literal values (Optional)** : You can specify literal values anywhere in the `ThumbnailPattern` object. For example, you can include them as a file name prefix or as a delimiter between `{resolution}` and `{count}`.
- **“{resolution}” (Optional)** : If you want Elastic Transcoder to include the resolution in the file name, include `{resolution}` in the `ThumbnailPattern` object. When creating thumbnails, Elastic Transcoder automatically saves the files in the format (.jpg or .png) that appears in the preset that you specified in the `PresetID` value of `CreateJobOutput`. Elastic Transcoder also appends the applicable file name extension.
- **ThumbnailEncryption (dict)** –

The encryption settings, if any, that you want Elastic Transcoder to apply to your thumbnail.

- **Mode (string)** –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don’t want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won’t be able to unencrypt your data.

- **Key (string)** –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-

encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Rotate** (*string*) –

The number of degrees clockwise by which you want Elastic Transcoder to rotate the output relative to the input. Enter one of the following values:

auto , 0 , 90 , 180 , 270

The value `auto` generally works only if the file that you're transcoding contains rotation metadata.

- **PresetId** (*string*) –

The value of the `Id` object for the preset that you want to use for this job. The preset determines the audio, video, and thumbnail settings that Elastic Transcoder uses for transcoding. To use a preset that you created, specify the preset ID that Elastic Transcoder returned in the response when you created the preset. You can also use the Elastic Transcoder system presets, which you can get with `ListPresets` .

- **SegmentDuration** (*string*) –

Warning: (Outputs in Fragmented MP4 or MPEG-TS format only.

If you specify a preset in `PresetId` for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `SegmentDuration` is the target maximum duration of each segment in seconds. For `HLSv3` format playlists, each media segment is stored in a separate `.ts` file. For `HLSv4` and `Smooth` playlists, all media segments for an output are stored in a single file. Each segment is approximately the length of the `SegmentDuration` , though individual segments might be shorter or longer.

The range of valid values is 1 to 60 seconds. If the duration of the video is not evenly divisible by `SegmentDuration`

, the duration of the last segment is the remainder of total length/SegmentDuration.

Elastic Transcoder creates an output-specific playlist for each output HLS output that you specify in `OutputKeys`. To add an output to the master playlist for this job, include it in the `OutputKeys` of the associated playlist.

- **Status** (*string*) –

The status of one output in a job. If you specified only one output for the job, `Outputs:Status` is always the same as `Job:Status`. If you specified more than one output:

- `Job:Status` and `Outputs:Status` for all of the outputs is Submitted until Elastic Transcoder starts to process the first output.
 - When Elastic Transcoder starts to process the first output, `Outputs:Status` for that output and `Job:Status` both change to Progressing. For each output, the value of `Outputs:Status` remains Submitted until Elastic Transcoder starts to process the output.
 - `Job:Status` remains Progressing until all of the outputs reach a terminal status, either Complete or Error.
 - When all of the outputs reach a terminal status, `Job:Status` changes to Complete only if `Outputs:Status` for all of the outputs is Complete. If `Outputs:Status` for one or more outputs is Error, the terminal status for `Job:Status` is also Error.
- The value of `Status` is one of the following: Submitted, Progressing, Complete, Canceled, or Error.

- **StatusDetail** (*string*) –

Information that further explains `Status`.

- **Duration** (*integer*) –

Duration of the output file, in seconds.

- **Width** (*integer*) –

Specifies the width of the output file in pixels.

- **Height** (*integer*) –

Height of the output file, in pixels.

- **FrameRate** (*string*) –

Frame rate of the output file, in frames per second.

- **FileSize** (*integer*) –

File size of the output file, in bytes.

- **DurationMillis** (*integer*) –

Duration of the output file, in milliseconds.

- **Watermarks** (*list*) –

Information about the watermarks that you want Elastic Transcoder to add to the video during transcoding. You can specify up to four watermarks for each output. Settings for each watermark must be defined in the preset that you specify in `Preset` for the current output.

Watermarks are added to the output video in the sequence in

which you list them in the job output the first watermark in the list is added to the output video first, the second watermark in the list is added next, and so on. As a result, if the settings in a preset cause Elastic Transcoder to place all watermarks in the same location, the second watermark that you add will cover the first one, the third one will cover the second, and the fourth one will cover the third.

- *(dict)* –

Watermarks can be in .png or .jpg format. If you want to display a watermark that is not rectangular, use the .png format, which supports transparency.

- **PresetWatermarkId** (*string*) –

The ID of the watermark settings that Elastic Transcoder uses to add watermarks to the video during transcoding. The settings are in the preset specified by `Preset` for the current output. In that preset, the value of `Watermarks Id` tells Elastic Transcoder which settings to use.

- **InputKey** (*string*) –

The name of the .png or .jpg file that you want to use for the watermark. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `Pipeline`; the `Input Bucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, **logos/128x64.png**, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your watermarks.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**

- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AlbumArt** (*dict*) –

The album art to be associated with the output file, if any.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder will handle the existence of multiple album artwork files.

- **Replace**: The specified album art will replace any existing album art.
- **Prepend**: The specified album art will be placed in front of any existing album art.
- **Append**: The specified album art will be placed after any existing album art.
- **Fallback**: If the original input file contains artwork, Elastic Transcoder will use that artwork for the output. If the original input does not contain artwork, Elastic Transcoder will use the specified album art file.
- **Artwork** (*list*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20. Valid formats are .jpg and .png

- *(dict)* –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20.

To remove artwork or leave the artwork empty, you can either set `Artwork` to `null`, or set the `Merge Policy` to “Replace” and use an empty `Artwork` array.

To pass through existing artwork unchanged, set the `Merge Policy` to “Prepend”, “Append”, or “Fallback”, and use an empty `Artwork` array.

- **InputKey** (*string*) –

The name of the file to be used as album art. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/pie.jpg`, include the prefix in the key. If the file isn’t in the specified bucket, Elastic Transcoder returns an error.

- **MaxWidth** (*string*) –

The maximum width of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

- **MaxHeight** (*string*) –

The maximum height of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.

- **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of the output album art:

- **Fit**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Fill**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output art and then crops it in the dimension (if any) that exceeds the maximum value.
- **Stretch**: Elastic Transcoder stretches the output art to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input art and the output art are different, the output art will be distorted.
- **Keep**: Elastic Transcoder does not scale the output art. If either dimension of the input art exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output art.
- **ShrinkToFit**: Elastic Transcoder scales the output art down so that its dimensions match the values that you spec-

ified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the art up.

- **ShrinkToFill** Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the art up.

- **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder may add white bars to the top and bottom and/or left and right sides of the output album art to make the total size of the output art match the values that you specified for `MaxWidth` and `MaxHeight`.

- **AlbumArtFormat** (*string*) –

The format of album art, if any. Valid formats are `.jpg` and `.png`.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your artwork.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Composition** (*list*) –

You can create an output file that contains an excerpt from the input file. This excerpt, called a clip, can come from the beginning, middle, or end of the file. The Composition object contains settings for the clips that make up an output file. For the current release, you can only specify settings for a single clip per output file. The Composition object cannot be null.

- (*dict*) –

Settings for one clip in a composition. All jobs in a playlist must have the same clip settings.

- **TimeSpan** (*dict*) –

Settings that determine when a clip begins and how long it lasts.

- **StartTime** (*string*) –

The place in the input file where you want a clip to start. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder starts at the beginning of the input file.

- **Duration** (*string*) –

The duration of the clip. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder creates an output file from StartTime to the end of the file.

If you specify a value longer than the duration of the input file, Elastic Transcoder transcodes the file and returns a warning

message.

- **Captions** (*dict*) –

You can configure Elastic Transcoder to transcode captions, or subtitles, from one format to another. All captions must be in UTF-8. Elastic Transcoder supports two types of captions:

- **Embedded:** Embedded captions are included in the same file as the audio and video. Elastic Transcoder supports only one embedded caption per language, to a maximum of 300 embedded captions per file. Valid input values include: `CEA-608` (EIA-608 , first non-empty channel only), `CEA-708` (EIA-708 , first non-empty channel only), and `mov-text`. Valid outputs include: `mov-text`. Elastic Transcoder supports a maximum of one embedded format per output.
- **Sidecar:** Sidecar captions are kept in a separate metadata file from the audio and video data. Sidecar captions require a player that is capable of understanding the relationship between the video file and the sidecar file. Elastic Transcoder supports only one sidecar caption per language, to a maximum of 20 sidecar captions per file. Valid input values include: `dfxp` (first div element only), `ebu-tt`, `scc`, `smpt`, `srt`, `ttml` (first div element only), and `webvtt`. Valid outputs include: `dfxp` (first div element only), `scc`, `srt`, and `webvtt`.

If you want `ttml` or `smpte-tt` compatible captions, specify `dfxp` as your output format.

Elastic Transcoder does not support OCR (Optical Character Recognition), does not accept pictures as a valid input for captions, and is not available for audio-only transcoding. Elastic Transcoder does not preserve text formatting (for example, italics) during the transcoding process.

To remove captions or leave the captions empty, set `Captions` to `null`. To pass through existing captions unchanged, set the `MergePolicy` to `MergeRetain`, and pass in a null `CaptionSources` array.

For more information on embedded files, see the Subtitles Wikipedia page.

For more information on sidecar files, see the Extensible Metadata Platform and Sidecar file Wikipedia pages.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder handles the existence of multiple captions.

- **MergeOverride:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the sidecar captions and ignores the embedded captions for that language.
- **MergeRetain:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the embedded captions and ignores

the sidecar captions for that language. If `CaptionSources` is empty, Elastic Transcoder omits all sidecar captions from the output files.

- **Override:** Elastic Transcoder transcodes only the sidecar captions that you specify in `CaptionSources`. `MergePolicy` cannot be null.

- **CaptionSources** (*list*) –

Source files for the input sidecar captions used during the transcoding process. To omit all sidecar captions, leave `CaptionSources` blank.

- (*dict*) –

A source file for the input sidecar captions used during the transcoding process.

- **Key** (*string*) –

The name of the sidecar caption file that you want Elastic Transcoder to include in the output file.

- **Language** (*string*) –

A string that specifies the language of the caption. Specify this as one of:

- 2-character ISO 639-1 code
- 3-character ISO 639-2 code

For more information on ISO language codes and language names, see the List of ISO 639-1 codes.

- **TimeOffset** (*string*) –

For clip generation or captions that do not start at the same time as the associated video file, the `TimeOffset` tells Elastic Transcoder how much of the video to encode before including captions.

Specify the `TimeOffset` in the form `[+-]SS.sss` or `[+-]HH:mm:SS.ss`.

- **Label** (*string*) –

The label of the caption shown in the player when choosing a language. We recommend that you put the caption language name here, in the language of the captions.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption sources.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.

- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **CaptionFormats** (*list*) –

The array of file formats for the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- (*dict*) –

The file format of the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- **Format** (*string*) –

The format you specify determines whether Elastic Transcoder generates an embedded or sidecar caption

for this output.

- **Valid Embedded Caption Formats:** * **for FLAC** : None
- **For MP3** : None
- **For MP4** : mov-text
- **For MPEG-TS** : None
- **For ogg** : None
- **For webm** : None
- **Valid Sidecar Caption Formats:** Elastic Transcoder supports dfxp (first div element only), scc, srt, and webvtt. If you want ttml or smpte-tt compatible captions, specify dfxp as your output format. * **For FMP4** : dfxp
- **Non-FMP4 outputs** : All sidecar types
fmp4 captions have an extension of .ismt
- **Pattern** (*string*) –

The prefix for caption filenames, in the form *description* - {language} , where:

- *description* is a description of the video.
- {language} is a literal value that Elastic Transcoder replaces with the two- or three-letter code for the language of the caption in the output file names.

If you don't include {language} in the file name pattern, Elastic Transcoder automatically appends "{language} " to the value that you specify for the description. In addition, Elastic Transcoder automatically appends the count to the end of the segment files.

For example, suppose you're transcoding into srt format. When you enter "Sydney-{language}-sunrise", and the language of the captions is English (en), the name of the first caption file will be Sydney-en-sunrise00000.srt.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption formats.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify S3-AWS-KMS and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following

settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your output files. If you choose to use encryption, you must specify a mode to use. If you choose not to use encryption, Elastic Transcoder will write an unencrypted file to your Amazon S3 bucket.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the

AWS-KMS key that you want to use to your pipeline.

- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AppliedColorSpaceConversion** (*string*) –

If Elastic Transcoder used a preset with a `ColorSpaceConversionMode` to transcode the output file, the `AppliedColorSpaceConversion` parameter shows the conversion used. If no `ColorSpaceConversionMode` was defined in the preset, this parameter will not be included in the job response.

- **Outputs** (*list*) –

Information about the output files. We recommend that you use the `Outputs` syntax for all jobs, even when you want Elastic Transcoder to transcode a file into only one format. Do not use both the `Outputs` and `Output` syntaxes in the same request. You can create a maximum of 30 outputs per job.

If you specify more than one output for a job, Elastic Transcoder creates the files for each output in the order in which you specify them in the job.

· *(dict)* –

Warning: Outputs recommended instead.

If you specified one output for a job, information about that output. If you specified multiple outputs for a job, the `Output` object lists information about the first output. This duplicates the information that is listed for the first output in the `Outputs` object.

· **Id** (*string*) –

A sequential counter, starting with 1, that identifies an output among the outputs from the current job. In the `Output` syntax, this value is always 1.

· **Key** (*string*) –

The name to assign to the transcoded file. Elastic Transcoder saves the file in the Amazon S3 bucket specified by the `OutputBucket` object in the pipeline that is specified by the pipeline ID.

· **ThumbnailPattern** (*string*) –

Whether you want Elastic Transcoder to create thumbnails for your videos and, if so, how you want Elastic Transcoder to name the files.

If you don't want Elastic Transcoder to create thumbnails, specify `""`.

If you do want Elastic Transcoder to create thumbnails, specify the information that you want to include in the file name for each thumbnail. You can specify the following values in any sequence:

- **"{count}" (Required)** : If you want to create thumbnails, you must include `{count}` in the `ThumbnailPattern` object. Wherever you specify `{count}`, Elastic Transcoder adds a five-digit sequence number (beginning with **00001**) to thumbnail file names. The number indicates where a given thumbnail appears in the sequence of thumbnails for a transcoded file. .. warning::If you specify a literal value and/or `{resolution}` but you omit `{count}`, Elastic Transcoder returns a validation error and does not create the job.
- **Literal values (Optional)** : You can specify literal values anywhere in the `ThumbnailPattern` object. For example, you can include them as a file name prefix or as a delimiter

between `{resolution}` and `{count}` .

- “**{resolution}**” (Optional) : If you want Elastic Transcoder to include the resolution in the file name, include `{resolution}` in the `ThumbnailPattern` object.

When creating thumbnails, Elastic Transcoder automatically saves the files in the format (.jpg or .png) that appears in the preset that you specified in the `PresetID` value of `CreateJobOutput` . Elastic Transcoder also appends the applicable file name extension.

- **ThumbnailEncryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your thumbnail.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3**: Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS**: Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don’t want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7**: A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR**: AES Counter Mode.
- **AES-GCM**: AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won’t be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Rotate** (*string*) –

The number of degrees clockwise by which you want Elastic Transcoder to rotate the output relative to the input. Enter one of the following values:

`auto, 0, 90, 180, 270`

The value `auto` generally works only if the file that you're transcoding contains rotation metadata.

- **PresetId** (*string*) –

The value of the `Id` object for the preset that you want to use for this job. The preset determines the audio, video, and thumbnail settings that Elastic Transcoder uses for transcoding. To use a preset that you created, specify the preset ID that Elastic Transcoder returned in the response when you created the preset. You can also use the Elastic Transcoder system presets, which you can get with `ListPresets`.

- **SegmentDuration** (*string*) –

Warning: (Outputs in Fragmented MP4 or MPEG-TS format only.

If you specify a preset in `PresetId` for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `SegmentDuration` is the target maximum duration of each segment in seconds. For `HLSv3` format playlists, each media segment is stored in a separate `.ts` file. For `HLSv4` and `Smooth` playlists, all media segments for an output are stored in a single file. Each segment is approximately the length of the `SegmentDuration`, though individual segments might be shorter or longer.

The range of valid values is 1 to 60 seconds. If the duration of the video is not evenly divisible by `SegmentDuration`, the duration of the last segment is the remainder of total length/`SegmentDuration`.

Elastic Transcoder creates an output-specific playlist for each output `HLS` output that you specify in `OutputKeys`. To add

an output to the master playlist for this job, include it in the `OutputKeys` of the associated playlist.

- **Status** (*string*) –

The status of one output in a job. If you specified only one output for the job, `Outputs:Status` is always the same as `Job:Status`. If you specified more than one output:

- `Job:Status` and `Outputs:Status` for all of the outputs is `Submitted` until Elastic Transcoder starts to process the first output.
- When Elastic Transcoder starts to process the first output, `Outputs:Status` for that output and `Job:Status` both change to `Progressing`. For each output, the value of `Outputs:Status` remains `Submitted` until Elastic Transcoder starts to process the output.
- `Job:Status` remains `Progressing` until all of the outputs reach a terminal status, either `Complete` or `Error`.
- When all of the outputs reach a terminal status, `Job:Status` changes to `Complete` only if `Outputs:Status` for all of the outputs is `Complete`. If `Outputs:Status` for one or more outputs is `Error`, the terminal status for `Job:Status` is also `Error`.

The value of `Status` is one of the following: `Submitted`, `Progressing`, `Complete`, `Canceled`, or `Error`.

- **StatusDetail** (*string*) –

Information that further explains `Status`.

- **Duration** (*integer*) –

Duration of the output file, in seconds.

- **Width** (*integer*) –

Specifies the width of the output file in pixels.

- **Height** (*integer*) –

Height of the output file, in pixels.

- **FrameRate** (*string*) –

Frame rate of the output file, in frames per second.

- **FileSize** (*integer*) –

File size of the output file, in bytes.

- **DurationMillis** (*integer*) –

Duration of the output file, in milliseconds.

- **Watermarks** (*list*) –

Information about the watermarks that you want Elastic Transcoder to add to the video during transcoding. You can specify up to four watermarks for each output. Settings for each watermark must be defined in the preset that you specify in `Preset` for the current output.

Watermarks are added to the output video in the sequence in which you list them in the job output. The first watermark in the list is added to the output video first, the second watermark in the list is added next, and so on. As a result, if the settings in a preset cause Elastic Transcoder to place all watermarks in the same location, the second watermark that you add will

cover the first one, the third one will cover the second, and the fourth one will cover the third.

- *(dict)* –

Watermarks can be in .png or .jpg format. If you want to display a watermark that is not rectangular, use the .png format, which supports transparency.

- **PresetWatermarkId** (*string*) –

The ID of the watermark settings that Elastic Transcoder uses to add watermarks to the video during transcoding. The settings are in the preset specified by `Preset` for the current output. In that preset, the value of `Watermarks Id` tells Elastic Transcoder which settings to use.

- **InputKey** (*string*) –

The name of the .png or .jpg file that you want to use for the watermark. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `Pipeline`; the `Input Bucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, **logos/128x64.png**, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your watermarks.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AlbumArt** (*dict*) –

The album art to be associated with the output file, if any.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder will handle the existence of multiple album artwork files.

- **Replace**: The specified album art will replace any existing album art.
- **Prepend**: The specified album art will be placed in front of any existing album art.
- **Append**: The specified album art will be placed after any existing album art.
- **Fallback**: If the original input file contains artwork, Elastic Transcoder will use that artwork for the output. If the original input does not contain artwork, Elastic Transcoder will use the specified album art file.

- **Artwork** (*list*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20. Valid formats are .jpg and .png

- (*dict*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20.

To remove artwork or leave the artwork empty, you can either set `Artwork` to null, or set the `Merge Policy` to “Replace” and use an empty `Artwork` array.

To pass through existing artwork unchanged, set the `Merge Policy` to “Prepend”, “Append”, or “Fallback”, and use an empty `Artwork` array.

- **InputKey** (*string*) –

The name of the file to be used as album art. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/pie.jpg`, include the prefix in the key. If the file isn’t in the specified bucket, Elastic Transcoder returns an error.

- **MaxWidth** (*string*) –

The maximum width of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

- **MaxHeight** (*string*) –

The maximum height of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.

- **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of the output album art:

- **Fit**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Fill**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output art and then crops it in the dimension (if any) that exceeds the maximum value.
- **Stretch**: Elastic Transcoder stretches the output art to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input art and the output art are different, the output art will be distorted.
- **Keep**: Elastic Transcoder does not scale the output art. If either dimension of the input art exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output art.
- **ShrinkToFit**: Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` with-

out exceeding either value. If you specify this option, Elastic Transcoder does not scale the art up.

- **ShrinkToFill** Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the art up.
- **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder may add white bars to the top and bottom and/or left and right sides of the output album art to make the total size of the output art match the values that you specified for `MaxWidth` and `MaxHeight`.

- **AlbumArtFormat** (*string*) –

The format of album art, if any. Valid formats are `.jpg` and `.png`.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your artwork.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder

to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Composition** (*list*) –

You can create an output file that contains an excerpt from the input file. This excerpt, called a clip, can come from the beginning, middle, or end of the file. The Composition object contains settings for the clips that make up an output file. For the current release, you can only specify settings for a single clip per output file. The Composition object cannot be null.

- (*dict*) –

Settings for one clip in a composition. All jobs in a playlist must have the same clip settings.

- **TimeSpan** (*dict*) –

Settings that determine when a clip begins and how long it lasts.

- **StartTime** (*string*) –

The place in the input file where you want a clip to start. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder starts at the beginning of the input file.

- **Duration** (*string*) –

The duration of the clip. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder creates an output file from StartTime to the end of the file.

If you specify a value longer than the duration of the input file, Elastic Transcoder transcodes the file and returns a warning message.

- **Captions** (*dict*) –

You can configure Elastic Transcoder to transcode captions, or subtitles, from one format to another. All captions must be in UTF-8. Elastic Transcoder supports two types of captions:

- **Embedded:** Embedded captions are included in the same file as the audio and video. Elastic Transcoder supports only one embedded caption per language, to a maximum of 300 embedded captions per file. Valid input values include: `CEA-608` (EIA-608 , first non-empty channel only), `CEA-708` (EIA-708 , first non-empty channel only), and `mov-text`. Valid outputs include: `mov-text`. Elastic Transcoder supports a maximum of one embedded format per output.
- **Sidecar:** Sidecar captions are kept in a separate metadata file from the audio and video data. Sidecar captions require a player that is capable of understanding the relationship between the video file and the sidecar file. Elastic Transcoder supports only one sidecar caption per language, to a maximum of 20 sidecar captions per file. Valid input values include: `dfxp` (first div element only), `ebu-tt` , `scc` , `smpt` , `srt` , `ttml` (first div element only), and `webvtt`. Valid outputs include: `dfxp` (first div element only), `scc` , `srt` , and `webvtt`.

If you want `ttml` or `smpte-tt` compatible captions, specify `dfxp` as your output format.

Elastic Transcoder does not support OCR (Optical Character Recognition), does not accept pictures as a valid input for captions, and is not available for audio-only transcoding. Elastic Transcoder does not preserve text formatting (for example, italics) during the transcoding process.

To remove captions or leave the captions empty, set `Captions` to `null`. To pass through existing captions unchanged, set the `MergePolicy` to `MergeRetain` , and pass in a null `CaptionSources` array.

For more information on embedded files, see the Subtitles Wikipedia page.

For more information on sidecar files, see the Extensible Metadata Platform and Sidecar file Wikipedia pages.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder handles the existence of multiple captions.

- **MergeOverride:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the sidecar captions and ignores the embedded captions for that language.
- **MergeRetain:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the embedded captions and ignores the sidecar captions for that language. If `CaptionSources`

is empty, Elastic Transcoder omits all sidecar captions from the output files.

- **Override:** Elastic Transcoder transcodes only the sidecar captions that you specify in `CaptionSources`. `MergePolicy` cannot be null.

- **CaptionSources** (*list*) –

Source files for the input sidecar captions used during the transcoding process. To omit all sidecar captions, leave `CaptionSources` blank.

- (*dict*) –

A source file for the input sidecar captions used during the transcoding process.

- **Key** (*string*) –

The name of the sidecar caption file that you want Elastic Transcoder to include in the output file.

- **Language** (*string*) –

A string that specifies the language of the caption. Specify this as one of:

- 2-character ISO 639-1 code
- 3-character ISO 639-2 code

For more information on ISO language codes and language names, see the List of ISO 639-1 codes.

- **TimeOffset** (*string*) –

For clip generation or captions that do not start at the same time as the associated video file, the `TimeOffset` tells Elastic Transcoder how much of the video to encode before including captions.

Specify the `TimeOffset` in the form `[+/-]SS.sss` or `[+/-]HH:mm:SS.ss`.

- **Label** (*string*) –

The label of the caption shown in the player when choosing a language. We recommend that you put the caption language name here, in the language of the captions.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption sources.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.

- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **CaptionFormats** (*list*) –

The array of file formats for the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- (*dict*) –

The file format of the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- **Format** (*string*) –

The format you specify determines whether Elastic Transcoder generates an embedded or sidecar caption

for this output.

- **Valid Embedded Caption Formats:** * for **FLAC** : None
- **For MP3** : None
- **For MP4** : mov-text
- **For MPEG-TS** : None
- **For ogg** : None
- **For webm** : None
- **Valid Sidecar Caption Formats:** Elastic Transcoder supports dfxp (first div element only), scc, srt, and webvtt. If you want ttml or smpte-tt compatible captions, specify dfxp as your output format. * **For FMP4** : dfxp
- **Non-FMP4 outputs** : All sidecar types
fmp4 captions have an extension of `.ismt`
- **Pattern** (*string*) –

The prefix for caption filenames, in the form *description* - {*language*} , where:

- *description* is a description of the video.
- {*language*} is a literal value that Elastic Transcoder replaces with the two- or three-letter code for the language of the caption in the output file names.

If you don't include {*language*} in the file name pattern, Elastic Transcoder automatically appends "{*language*}" to the value that you specify for the description. In addition, Elastic Transcoder automatically appends the count to the end of the segment files.

For example, suppose you're transcoding into srt format. When you enter "Sydney-{*language*}-sunrise", and the language of the captions is English (en), the name of the first caption file will be Sydney-en-sunrise00000.srt.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption formats.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following

settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your output files. If you choose to use encryption, you must specify a mode to use. If you choose not to use encryption, Elastic Transcoder will write an unencrypted file to your Amazon S3 bucket.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the

AWS-KMS key that you want to use to your pipeline.

- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AppliedColorSpaceConversion** (*string*) –

If Elastic Transcoder used a preset with a `ColorSpaceConversionMode` to transcode the output file, the `AppliedColorSpaceConversion` parameter shows the conversion used. If no `ColorSpaceConversionMode` was defined in the preset, this parameter will not be included in the job response.

- **OutputKeyPrefix** (*string*) –

The value, if any, that you want Elastic Transcoder to prepend to the names of all files that this job creates, including output files, thumbnails, and playlists. We recommend that you add a / or some other delimiter to the end of the `OutputKeyPrefix`.

- **Playlists** (*list*) –

Warning: Outputs in Fragmented MP4 or MPEG-TS format only.

If you specify a preset in `PresetId` for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `Playlists` contains information about the master playlists that you want Elastic Transcoder to create.

The maximum number of master playlists in a job is 30.

- (*dict*) –

Use Only for Fragmented MP4 or MPEG-TS Outputs. If you specify a preset for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `Playlists` contains information about the master playlists that you want Elastic Transcoder to create. We recommend that you create only one master playlist per output format. The maximum number of master playlists in a job is 30.

- **Name** (*string*) –

The name that you want Elastic Transcoder to assign to the master playlist, for example, `nyc-vacation.m3u8`. If the name includes a / character, the section of the name before the last / must be identical for all `Name` objects. If you create more than one master playlist, the values of all `Name` objects must be unique.

Note : Elastic Transcoder automatically appends the relevant file extension to the file name (`.m3u8` for `HLSv3` and `HLSv4` playlists, and `.ism` and `.ismc` for `Smooth` playlists). If you include a file extension in `Name`, the file name will have two extensions.

- **Format** (*string*) –

The format of the output playlist. Valid formats include `HLSv3`, `HLSv4`, and `Smooth`.

- **OutputKeys** (*list*) –

For each output in this job that you want to include in a master playlist, the value of the `Outputs:Key` object.

- If your output is not `HLS` or does not have a segment duration set, the name of the output file is a concatenation of `OutputKeyPrefix` and `Outputs:Key:OutputKeyPrefix“Outputs:Key“`
- If your output is `HLSv3` and has a segment duration set, or is not included in a playlist, Elastic Transcoder creates an output playlist file with a file extension of `.m3u8`, and a series of `.ts` files that include a five-digit sequential counter beginning with 00000: `OutputKeyPrefix“Outputs:Key“ .m3u8`
`OutputKeyPrefix“Outputs:Key“ 00000.ts`

- If your output is `HLSv4`, has a segment duration set, and is included in an `HLSv4` playlist, Elastic Transcoder creates an output playlist file with a file extension of `_v4.m3u8`. If the output is video, Elastic Transcoder also creates an output file with an extension of `_iframe.m3u8`: `OutputKeyPrefix“Outputs:Key“_v4.m3u8` `OutputKeyPrefix“Outputs:Key“_iframe.m3u8` `OutputKeyPrefix“Outputs:Key“.ts`

Elastic Transcoder automatically appends the relevant file extension to the file name. If you include a file extension in Output Key, the file name will have two extensions.

If you include more than one output in a playlist, any segment duration settings, clip settings, or caption settings must be the same for all outputs in the playlist. For Smooth playlists, the `Audio:Profile`, `Video:Profile`, and `Video:FrameRate` to `Video:KeyframesMaxDist` ratio must be the same for all outputs.

- *(string)* –
- **HlsContentProtection** (*dict*) –

The HLS content protection settings, if any, that you want Elastic Transcoder to apply to the output files associated with this playlist.

- **Method** (*string*) –

The content protection method for your output. The only valid value is: `aes-128`.

This value will be written into the method attribute of the `EXT-X-KEY` metadata tag in the output playlist.

- **Key** (*string*) –

If you want Elastic Transcoder to generate a key for you, leave this field blank.

If you choose to supply your own key, you must encrypt the key by using AWS KMS. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

`128`, `192`, or `256`.

- **KeyMd5** (*string*) –

If Elastic Transcoder is generating your key for you, you must leave this field blank.

The MD5 digest of the key that you want Elastic Transcoder to use to encrypt your output file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

- **InitializationVector** (*string*) –

If Elastic Transcoder is generating your key for you, you must leave this field blank.

The series of random bits created by a random bit generator,

unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

- **LicenseAcquisitionUrl** (*string*) –

The location of the license key required to decrypt your HLS playlist. The URL must be an absolute path, and is referenced in the URI attribute of the EXT-X-KEY metadata tag in the playlist file.

- **KeyStoragePolicy** (*string*) –

Specify whether you want Elastic Transcoder to write your HLS license key to an Amazon S3 bucket. If you choose `WithVariantPlaylists`, `LicenseAcquisitionUrl` must be left blank and Elastic Transcoder writes your data key into the same bucket as the associated playlist.

- **PlayReadyDrm** (*dict*) –

The DRM settings, if any, that you want Elastic Transcoder to apply to the output files associated with this playlist.

- **Format** (*string*) –

The type of DRM, if any, that you want Elastic Transcoder to apply to the output files associated with this playlist.

- **Key** (*string*) –

The DRM key for your file, provided by your DRM license provider. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using AWS KMS.

- **KeyMd5** (*string*) –

The MD5 digest of the key used for DRM on your file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

- **KeyId** (*string*) –

The ID for your DRM key, so that your DRM license provider knows which key to provide.

The key ID must be provided in big endian, and Elastic Transcoder will convert it to little endian before inserting it into the PlayReady DRM headers. If you are unsure whether your license server provides your key ID in big or little endian, check with your DRM provider.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your files. The initialization vector must be base64-encoded, and it must be exactly 8 bytes

long before being base64-encoded. If no initialization vector is provided, Elastic Transcoder generates one for you.

- **LicenseAcquisitionUrl** (*string*) –

The location of the license key required to play DRM content. The URL must be an absolute path, and is referenced by the PlayReady header. The PlayReady header is referenced in the protection header of the client manifest for Smooth Streaming outputs, and in the EXT-X-DXDRM and EXT-XDXDRMINFO metadata tags for HLS playlist outputs. An example URL looks like this: <https://www.example.com/exampleKey/>

- **Status** (*string*) –

The status of the job with which the playlist is associated.

- **StatusDetail** (*string*) –

Information that further explains the status.

- **Status** (*string*) –

The status of the job: Submitted , Progressing , Complete , Canceled , or Error .

- **UserMetadata** (*dict*) –

User-defined metadata that you want to associate with an Elastic Transcoder job. You specify metadata in *key/value* pairs, and you can add up to 10 *key/value* pairs per job. Elastic Transcoder does not guarantee that *key/value* pairs will be returned in the same order in which you specify them.

Metadata *keys* and *values* must use characters from the following list:

- 0–9
- A–Z and a–z
- Space
- The following symbols: `_ . : / = + - % @`
- (*string*) –
- (*string*) –
- **Timing** (*dict*) –

Details about the timing of a job.

- **SubmitTimeMillis** (*integer*) –

The time the job was submitted to Elastic Transcoder, in epoch milliseconds.

- **StartTimeMillis** (*integer*) –

The time the job began transcoding, in epoch milliseconds.

- **FinishTimeMillis** (*integer*) –

The time the job finished transcoding, in epoch milliseconds.

- **NextPageToken** (*string*) –

A value that you use to access the second and subsequent pages of results, if any. When the jobs in the specified pipeline fit on one page or when you've reached the last page of results, the value of `NextPageToken` is `null`.

list_jobs_by_status (***kwargs*)

The `ListJobsByStatus` operation gets a list of jobs that have a specified status. The response body contains

one element for each job that satisfies the search criteria.

Request Syntax

```
response = client.list_jobs_by_status(
    Status='string',
    Ascending='string',
    PageToken='string'
)
```

Parameters

- **Status** (*string*) – [REQUIRED]

To get information about all of the jobs associated with the current AWS account that have a given status, specify the following status: Submitted, Progressing, Complete, Canceled, or Error.

- **Ascending** (*string*) – To list jobs in chronological order by the date and time that they were submitted, enter `true`. To list jobs in reverse chronological order, enter `false`.
- **PageToken** (*string*) – When Elastic Transcoder returns more than one page of results, use `pageToken` in subsequent GET requests to get each successive page of results.

Return type dict

Returns

Response Syntax

```
{
  'Jobs': [
    {
      'Id': 'string',
      'Arn': 'string',
      'PipelineId': 'string',
      'Input': {
        'Key': 'string',
        'FrameRate': 'string',
        'Resolution': 'string',
        'AspectRatio': 'string',
        'Interlaced': 'string',
        'Container': 'string',
        'Encryption': {
          'Mode': 'string',
          'Key': 'string',
          'KeyMd5': 'string',
          'InitializationVector': 'string'
        },
        'DetectedProperties': {
          'Width': 123,
          'Height': 123,
          'FrameRate': 'string',
          'FileSize': 123,
          'DurationMillis': 123
        }
      },
      'Output': {
        'Id': 'string',
        'Key': 'string',
        'ThumbnailPattern': 'string',
```

```

    'ThumbnailEncryption': {
        'Mode': 'string',
        'Key': 'string',
        'KeyMd5': 'string',
        'InitializationVector': 'string'
    },
    'Rotate': 'string',
    'PresetId': 'string',
    'SegmentDuration': 'string',
    'Status': 'string',
    'StatusDetail': 'string',
    'Duration': 123,
    'Width': 123,
    'Height': 123,
    'FrameRate': 'string',
    'FileSize': 123,
    'DurationMillis': 123,
    'Watermarks': [
        {
            'PresetWatermarkId': 'string',
            'InputKey': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        }
    ],
    'AlbumArt': {
        'MergePolicy': 'string',
        'Artwork': [
            {
                'InputKey': 'string',
                'MaxWidth': 'string',
                'MaxHeight': 'string',
                'SizingPolicy': 'string',
                'PaddingPolicy': 'string',
                'AlbumArtFormat': 'string',
                'Encryption': {
                    'Mode': 'string',
                    'Key': 'string',
                    'KeyMd5': 'string',
                    'InitializationVector': 'string'
                }
            }
        ]
    },
    'Composition': [
        {
            'TimeSpan': {
                'StartTime': 'string',
                'Duration': 'string'
            }
        }
    ],
    'Captions': {
        'MergePolicy': 'string',

```

```
    'CaptionSources': [
        {
            'Key': 'string',
            'Language': 'string',
            'TimeOffset': 'string',
            'Label': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        },
    ],
    'CaptionFormats': [
        {
            'Format': 'string',
            'Pattern': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        },
    ],
],
'Encryption': {
    'Mode': 'string',
    'Key': 'string',
    'KeyMd5': 'string',
    'InitializationVector': 'string'
},
'AppliedColorSpaceConversion': 'string'
},
'Outputs': [
    {
        'Id': 'string',
        'Key': 'string',
        'ThumbnailPattern': 'string',
        'ThumbnailEncryption': {
            'Mode': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'InitializationVector': 'string'
        },
        'Rotate': 'string',
        'PresetId': 'string',
        'SegmentDuration': 'string',
        'Status': 'string',
        'StatusDetail': 'string',
        'Duration': 123,
        'Width': 123,
        'Height': 123,
        'FrameRate': 'string',
        'FileSize': 123,
        'DurationMillis': 123,
        'Watermarks': [
```

```

        {
            'PresetWatermarkId': 'string',
            'InputKey': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        },
    ],
    'AlbumArt': {
        'MergePolicy': 'string',
        'Artwork': [
            {
                'InputKey': 'string',
                'MaxWidth': 'string',
                'MaxHeight': 'string',
                'SizingPolicy': 'string',
                'PaddingPolicy': 'string',
                'AlbumArtFormat': 'string',
                'Encryption': {
                    'Mode': 'string',
                    'Key': 'string',
                    'KeyMd5': 'string',
                    'InitializationVector': 'string'
                }
            }
        ],
    },
    'Composition': [
        {
            'TimeSpan': {
                'StartTime': 'string',
                'Duration': 'string'
            }
        },
    ],
    'Captions': {
        'MergePolicy': 'string',
        'CaptionSources': [
            {
                'Key': 'string',
                'Language': 'string',
                'TimeOffset': 'string',
                'Label': 'string',
                'Encryption': {
                    'Mode': 'string',
                    'Key': 'string',
                    'KeyMd5': 'string',
                    'InitializationVector': 'string'
                }
            }
        ],
    },
    'CaptionFormats': [
        {
            'Format': 'string',
            'Pattern': 'string',

```

```
        'Encryption': {
            'Mode': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'InitializationVector': 'string'
        },
    ],
    'Encryption': {
        'Mode': 'string',
        'Key': 'string',
        'KeyMd5': 'string',
        'InitializationVector': 'string'
    },
    'AppliedColorSpaceConversion': 'string'
},
],
'OutputKeyPrefix': 'string',
'Playlists': [
    {
        'Name': 'string',
        'Format': 'string',
        'OutputKeys': [
            'string',
        ],
        'HlsContentProtection': {
            'Method': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'InitializationVector': 'string',
            'LicenseAcquisitionUrl': 'string',
            'KeyStoragePolicy': 'string'
        },
        'PlayReadyDrm': {
            'Format': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'KeyId': 'string',
            'InitializationVector': 'string',
            'LicenseAcquisitionUrl': 'string'
        },
        'Status': 'string',
        'StatusDetail': 'string'
    },
],
'Status': 'string',
'UserMetadata': {
    'string': 'string'
},
'Timing': {
    'SubmitTimeMillis': 123,
    'StartTimeMillis': 123,
    'FinishTimeMillis': 123
}
},
],
'NextPageToken': 'string'
```



```
}
```

Response Structure

- (*dict*) –

The `ListJobsByStatusResponse` structure.

- **Jobs** (*list*) –

An array of `Job` objects that have the specified status.

- * (*dict*) –

A section of the response body that provides information about the job that is created.

- **Id** (*string*) –

The identifier that Elastic Transcoder assigned to the job. You use this value to get settings for the job or to delete the job.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) for the job.

- **PipelineId** (*string*) –

The `Id` of the pipeline that you want Elastic Transcoder to use for transcoding. The pipeline determines several settings, including the Amazon S3 bucket from which Elastic Transcoder gets the files to transcode and the bucket into which Elastic Transcoder puts the transcoded files.

- **Input** (*dict*) –

A section of the request or response body that provides information about the file that is being transcoded.

- **Key** (*string*) –

The name of the file to transcode. Elsewhere in the body of the JSON block is the the ID of the pipeline to use for processing the job. The `InputBucket` object in that pipeline tells Elastic Transcoder which Amazon S3 bucket to get the file from.

If the file name includes a prefix, such as `cooking/lasagna.mpg`, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

- **FrameRate** (*string*) –

The frame rate of the input file. If you want Elastic Transcoder to automatically detect the frame rate of the input file, specify `auto`. If you want to specify the frame rate for the input file, enter one of the following values:

10, 15, 23.97, 24, 25, 29.97, 30, 60

If you specify a value other than `auto`, Elastic Transcoder disables automatic detection of the frame rate.

- **Resolution** (*string*) –

This value must be `auto`, which causes Elastic Transcoder to automatically detect the resolution of the input file.

- **AspectRatio** (*string*) –

The aspect ratio of the input file. If you want Elastic Transcoder to automatically detect the aspect ratio of the in-

put file, specify `auto`. If you want to specify the aspect ratio for the output file, enter one of the following values:

`1:1`, `4:3`, `3:2`, `16:9`

If you specify a value other than `auto`, Elastic Transcoder disables automatic detection of the aspect ratio.

- **Interlaced** (*string*) –

Whether the input file is interlaced. If you want Elastic Transcoder to automatically detect whether the input file is interlaced, specify `auto`. If you want to specify whether the input file is interlaced, enter one of the following values:

`true`, `false`

If you specify a value other than `auto`, Elastic Transcoder disables automatic detection of interlacing.

- **Container** (*string*) –

The container type for the input file. If you want Elastic Transcoder to automatically detect the container type of the input file, specify `auto`. If you want to specify the container type for the input file, enter one of the following values:

`3gp`, `aac`, `asf`, `avi`, `divx`, `flv`, `m4a`,
`mkv`, `mov`, `mp3`, `mp4`, `mpeg`, `mpeg-ps`,
`mpeg-ts`, `mxv`, `ogg`, `vob`, `wav`, `webm`

- **Encryption** (*dict*) –

The encryption settings, if any, that are used for decrypting your input files. If your input file is encrypted, you must specify the mode that Elastic Transcoder will use to decrypt your file.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3**: Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS**: Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7**: A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR**: AES Counter Mode.
- **AES-GCM**: AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **DetectedProperties** (*dict*) –

The detected properties of the input file.

- **Width** (*integer*) –

The detected width of the input file, in pixels.

- **Height** (*integer*) –

The detected height of the input file, in pixels.

- **FrameRate** (*string*) –

The detected frame rate of the input file, in frames per second.

- **FileSize** (*integer*) –

The detected file size of the input file, in bytes.

- **DurationMillis** (*integer*) –

The detected duration of the input file, in milliseconds.

- **Output** (*dict*) –

If you specified one output for a job, information about that output. If you specified multiple outputs for a job, the Output object lists information about the first output. This duplicates the information that is listed for the first output in the Outputs object.

Warning: Outputs recommended instead.

A section of the request or response body that provides information about the transcoded (target) file.

- **Id** (*string*) –

A sequential counter, starting with 1, that identifies an output among the outputs from the current job. In the Output syntax, this value is always 1.

- **Key** (*string*) –

The name to assign to the transcoded file. Elastic Transcoder saves the file in the Amazon S3 bucket specified by the `OutputBucket` object in the pipeline that is specified by the pipeline ID.

- **ThumbnailPattern** (*string*) –

Whether you want Elastic Transcoder to create thumbnails for your videos and, if so, how you want Elastic Transcoder to name the files.

If you don't want Elastic Transcoder to create thumbnails, specify "".

If you do want Elastic Transcoder to create thumbnails, specify the information that you want to include in the file name for each thumbnail. You can specify the following values in any sequence:

- **“{count}” (Required)** : If you want to create thumbnails, you must include `{count}` in the `ThumbnailPattern` object. Wherever you specify `{count}`, Elastic Transcoder adds a five-digit sequence number (beginning with **00001**) to thumbnail file names. The number indicates where a given thumbnail appears in the sequence of thumbnails for a transcoded file. .. warning::If you specify a literal value and/or `{resolution}` but you omit `{count}`, Elastic Transcoder returns a validation error and does not create the job.

- **Literal values (Optional)** : You can specify literal values anywhere in the `ThumbnailPattern` object. For example, you can include them as a file name prefix or as a delimiter between `{resolution}` and `{count}`.

- **“{resolution}” (Optional)** : If you want Elastic Transcoder to include the resolution in the file name, include `{resolution}` in the `ThumbnailPattern` object.

When creating thumbnails, Elastic Transcoder automatically saves the files in the format (.jpg or .png) that appears in the preset that you specified in the `PresetID` value of `CreateJobOutput`. Elastic Transcoder also appends the applicable file name extension.

- **ThumbnailEncryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your thumbnail.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long

before being base64-encoded.

- **Rotate** (*string*) –

The number of degrees clockwise by which you want Elastic Transcoder to rotate the output relative to the input. Enter one of the following values:

`auto`, `0`, `90`, `180`, `270`

The value `auto` generally works only if the file that you're transcoding contains rotation metadata.

- **PresetId** (*string*) –

The value of the `Id` object for the preset that you want to use for this job. The preset determines the audio, video, and thumbnail settings that Elastic Transcoder uses for transcoding. To use a preset that you created, specify the preset ID that Elastic Transcoder returned in the response when you created the preset. You can also use the Elastic Transcoder system presets, which you can get with `ListPresets`.

- **SegmentDuration** (*string*) –

Warning: (Outputs in Fragmented MP4 or MPEG-TS format only.

If you specify a preset in `PresetId` for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `SegmentDuration` is the target maximum duration of each segment in seconds. For `HLSv3` format playlists, each media segment is stored in a separate `.ts` file. For `HLSv4` and `Smooth` playlists, all media segments for an output are stored in a single file. Each segment is approximately the length of the `SegmentDuration`, though individual segments might be shorter or longer.

The range of valid values is 1 to 60 seconds. If the duration of the video is not evenly divisible by `SegmentDuration`, the duration of the last segment is the remainder of total length/`SegmentDuration`.

Elastic Transcoder creates an output-specific playlist for each output `HLS` output that you specify in `OutputKeys`. To add an output to the master playlist for this job, include it in the `OutputKeys` of the associated playlist.

- **Status** (*string*) –

The status of one output in a job. If you specified only one output for the job, `Outputs:Status` is always the same as `Job:Status`. If you specified more than one output:

- `Job:Status` and `Outputs:Status` for all of the outputs is `Submitted` until Elastic Transcoder starts to process the first output.
- When Elastic Transcoder starts to process the first output, `Outputs:Status` for that output and `Job:Status` both change to `Progressing`. For each output, the value of `Outputs:Status` remains `Submitted` until Elastic Transcoder starts to process the output.

- **Job:Status** remains Progressing until all of the outputs reach a terminal status, either Complete or Error.
- When all of the outputs reach a terminal status, **Job:Status** changes to Complete only if **Outputs:Status** for all of the outputs is Complete. If **Outputs:Status** for one or more outputs is Error, the terminal status for **Job:Status** is also Error.
The value of **Status** is one of the following: Submitted, Progressing, Complete, Canceled, or Error.
- **StatusDetail** (*string*) –
Information that further explains **Status**.
- **Duration** (*integer*) –
Duration of the output file, in seconds.
- **Width** (*integer*) –
Specifies the width of the output file in pixels.
- **Height** (*integer*) –
Height of the output file, in pixels.
- **FrameRate** (*string*) –
Frame rate of the output file, in frames per second.
- **FileSize** (*integer*) –
File size of the output file, in bytes.
- **DurationMillis** (*integer*) –
Duration of the output file, in milliseconds.
- **Watermarks** (*list*) –
Information about the watermarks that you want Elastic Transcoder to add to the video during transcoding. You can specify up to four watermarks for each output. Settings for each watermark must be defined in the preset that you specify in **Preset** for the current output.

Watermarks are added to the output video in the sequence in which you list them in the job output: the first watermark in the list is added to the output video first, the second watermark in the list is added next, and so on. As a result, if the settings in a preset cause Elastic Transcoder to place all watermarks in the same location, the second watermark that you add will cover the first one, the third one will cover the second, and the fourth one will cover the third.
- (*dict*) –
Watermarks can be in .png or .jpg format. If you want to display a watermark that is not rectangular, use the .png format, which supports transparency.
- **PresetWatermarkId** (*string*) –
The ID of the watermark settings that Elastic Transcoder uses to add watermarks to the video during transcoding. The settings are in the preset specified by **Preset** for the current output. In that preset, the value of **Watermarks Id** tells Elastic Transcoder which settings to use.
- **InputKey** (*string*) –

The name of the .png or .jpg file that you want to use for the watermark. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `Pipeline`; the `Input Bucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, **logos/128x64.png**, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your watermarks.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3**: Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS**: Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7**: A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR**: AES Counter Mode.
- **AES-GCM**: AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AlbumArt** (*dict*) –

The album art to be associated with the output file, if any.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder will handle the existence of multiple album artwork files.

- **Replace**: The specified album art will replace any existing album art.
- **Prepend**: The specified album art will be placed in front of any existing album art.
- **Append**: The specified album art will be placed after any existing album art.
- **Fallback**: If the original input file contains artwork, Elastic Transcoder will use that artwork for the output. If the original input does not contain artwork, Elastic Transcoder will use the specified album art file.

- **Artwork** (*list*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20. Valid formats are .jpg and .png

- (*dict*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20.

To remove artwork or leave the artwork empty, you can either set `Artwork` to null, or set the `Merge Policy` to “Replace” and use an empty `Artwork` array.

To pass through existing artwork unchanged, set the `Merge Policy` to “Prepend”, “Append”, or “Fallback”, and use an empty `Artwork` array.

- **InputKey** (*string*) –

The name of the file to be used as album art. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/pie.jpg`, include the prefix in the key. If

the file isn't in the specified bucket, Elastic Transcoder returns an error.

- **MaxWidth** (*string*) –

The maximum width of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

- **MaxHeight** (*string*) –

The maximum height of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.

- **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of the output album art:

- **Fit**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Fill**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output art and then crops it in the dimension (if any) that exceeds the maximum value.
- **Stretch**: Elastic Transcoder stretches the output art to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input art and the output art are different, the output art will be distorted.
- **Keep**: Elastic Transcoder does not scale the output art. If either dimension of the input art exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output art.
- **ShrinkToFit**: Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the art up.
- **ShrinkToFill**: Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the art up.
- **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder may add white bars to the top and bottom and/or left and right sides of the output album art to make the total size of the output art match the values that you specified for `MaxWidth` and `MaxHeight`.

- **AlbumArtFormat** (*string*) –

The format of album art, if any. Valid formats are `.jpg` and `.png`.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your artwork.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to

encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Composition** (*list*) –

You can create an output file that contains an excerpt from the input file. This excerpt, called a clip, can come from the beginning, middle, or end of the file. The Composition object contains settings for the clips that make up an output file. For the current release, you can only specify settings for a single clip per output file. The Composition object cannot be null.

- (*dict*) –

Settings for one clip in a composition. All jobs in a playlist must have the same clip settings.

- **TimeSpan** (*dict*) –

Settings that determine when a clip begins and how long it lasts.

- **StartTime** (*string*) –

The place in the input file where you want a clip to start. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder starts at the beginning of the input file.

- **Duration** (*string*) –

The duration of the clip. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder creates an output file from StartTime to the end of the file.

If you specify a value longer than the duration of the input file, Elastic Transcoder transcodes the file and returns a warning message.

- **Captions** (*dict*) –

You can configure Elastic Transcoder to transcode captions, or subtitles, from one format to another. All captions must be in UTF-8. Elastic Transcoder supports two types of captions:

- **Embedded:** Embedded captions are included in the same file as the audio and video. Elastic Transcoder supports only one embedded caption per language, to a maximum of 300 embedded captions per file. Valid input values include: CEA-608 (EIA-608 , first non-empty channel only), CEA-708 (EIA-708 , first non-empty channel only), and mov-text. Valid outputs include: mov-text. Elastic Transcoder supports a maximum of one embedded format per output.
- **Sidecar:** Sidecar captions are kept in a separate metadata file from the audio and video data. Sidecar captions require a player that is capable of understanding the relationship between the video file and the sidecar file. Elastic Transcoder supports only one sidecar caption per language, to a maxi-

mum of 20 sidecar captions per file. Valid input values include: `dfxp` (first div element only), `ebu-tt`, `scc`, `smpt`, `srt`, `ttml` (first div element only), and `webvtt`. Valid outputs include: `dfxp` (first div element only), `scc`, `srt`, and `webvtt`.

If you want `ttml` or `smp-te-tt` compatible captions, specify `dfxp` as your output format.

Elastic Transcoder does not support OCR (Optical Character Recognition), does not accept pictures as a valid input for captions, and is not available for audio-only transcoding. Elastic Transcoder does not preserve text formatting (for example, italics) during the transcoding process.

To remove captions or leave the captions empty, set `Captions` to `null`. To pass through existing captions unchanged, set the `MergePolicy` to `MergeRetain`, and pass in a `null` `CaptionSources` array.

For more information on embedded files, see the Subtitles Wikipedia page.

For more information on sidecar files, see the Extensible Metadata Platform and Sidecar file Wikipedia pages.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder handles the existence of multiple captions.

- **MergeOverride:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the sidecar captions and ignores the embedded captions for that language.

- **MergeRetain:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the embedded captions and ignores the sidecar captions for that language. If `CaptionSources` is empty, Elastic Transcoder omits all sidecar captions from the output files.

- **Override:** Elastic Transcoder transcodes only the sidecar captions that you specify in `CaptionSources`.

`MergePolicy` cannot be `null`.

- **CaptionSources** (*list*) –

Source files for the input sidecar captions used during the transcoding process. To omit all sidecar captions, leave `CaptionSources` blank.

- (*dict*) –

A source file for the input sidecar captions used during the transcoding process.

- **Key** (*string*) –

The name of the sidecar caption file that you want Elastic Transcoder to include in the output file.

- **Language** (*string*) –

A string that specifies the language of the caption. Specify this as one of:

- 2-character ISO 639-1 code
- 3-character ISO 639-2 code

For more information on ISO language codes and language names, see the List of ISO 639-1 codes.

- **TimeOffset** (*string*) –

For clip generation or captions that do not start at the same time as the associated video file, the `TimeOffset` tells Elastic Transcoder how much of the video to encode before including captions.

Specify the `TimeOffset` in the form `[+-]SS.sss` or `[+-]HH:mm:SS.ss`.

- **Label** (*string*) –

The label of the caption shown in the player when choosing a language. We recommend that you put the caption language name here, in the language of the captions.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption sources.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3**: Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS**: Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7**: A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR**: AES Counter Mode.
- **AES-GCM**: AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **CaptionFormats** (*list*) –

The array of file formats for the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- (*dict*) –

The file format of the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- **Format** (*string*) –

The format you specify determines whether Elastic Transcoder generates an embedded or sidecar caption for this output.

- **Valid Embedded Caption Formats:** * **for FLAC** : None

- **For MP3** : None

- **For MP4** : mov-text

- **For MPEG-TS** : None

- **For ogg** : None

- **For webm** : None

- **Valid Sidecar Caption Formats:** Elastic Transcoder supports dfxp (first div element only), scc, srt, and webvtt. If you want ttml or smpte-tt compatible captions, specify dfxp as your output format. * **For FMP4** : dfxp

- **Non-FMP4 outputs** : All sidecar types

fmp4 captions have an extension of `.ismt`

- **Pattern** (*string*) –

The prefix for caption filenames, in the form *description* - {language} , where:

- *description* is a description of the video.

- {language} is a literal value that Elastic Transcoder replaces with the two- or three-letter code for the language of

the caption in the output file names.

If you don't include `{language}` in the file name pattern, Elastic Transcoder automatically appends "`{language}`" to the value that you specify for the description. In addition, Elastic Transcoder automatically appends the count to the end of the segment files.

For example, suppose you're transcoding into srt format. When you enter "Sydney-`{language}`-sunrise", and the language of the captions is English (en), the name of the first caption file will be Sydney-en-sunrise00000.srt.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption formats.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key

Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your output files. If you choose to use encryption, you must specify a mode to use. If you choose not to use encryption, Elastic Transcoder will write an unencrypted file to your Amazon S3 bucket.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AppliedColorSpaceConversion** (*string*) –

If Elastic Transcoder used a preset with a `ColorSpaceConversionMode` to transcode the output file, the `AppliedColorSpaceConversion` parameter shows the conversion used. If no `ColorSpaceConversionMode` was defined in the preset, this parameter will not be included in the job response.

- **Outputs** (*list*) –

Information about the output files. We recommend that you use the `Outputs` syntax for all jobs, even when you want Elastic Transcoder to transcode a file into only one format. Do not use both the `Outputs` and `Output` syntaxes in the same request. You can create a maximum of 30 outputs per job.

If you specify more than one output for a job, Elastic Transcoder creates the files for each output in the order in which you specify them in the job.

- (*dict*) –

Warning: Outputs recommended instead.

If you specified one output for a job, information about that output. If you specified multiple outputs for a job, the `Output` object lists information about the first output. This duplicates the information that is listed for the first output in the `Outputs` object.

- **Id** (*string*) –

A sequential counter, starting with 1, that identifies an output among the outputs from the current job. In the Output syntax, this value is always 1.

- **Key** (*string*) –

The name to assign to the transcoded file. Elastic Transcoder saves the file in the Amazon S3 bucket specified by the `OutputBucket` object in the pipeline that is specified by the pipeline ID.

- **ThumbnailPattern** (*string*) –

Whether you want Elastic Transcoder to create thumbnails for your videos and, if so, how you want Elastic Transcoder to name the files.

If you don't want Elastic Transcoder to create thumbnails, specify `""`.

If you do want Elastic Transcoder to create thumbnails, specify the information that you want to include in the file name for each thumbnail. You can specify the following values in any sequence:

- **“{count}” (Required)** : If you want to create thumbnails, you must include `{count}` in the `ThumbnailPattern` object. Wherever you specify `{count}`, Elastic Transcoder adds a five-digit sequence number (beginning with **00001**) to thumbnail file names. The number indicates where a given thumbnail appears in the sequence of thumbnails for a transcoded file. .. warning::If you specify a literal value and/or `{resolution}` but you omit `{count}`, Elastic Transcoder returns a validation error and does not create the job.
- **Literal values (Optional)** : You can specify literal values anywhere in the `ThumbnailPattern` object. For example, you can include them as a file name prefix or as a delimiter between `{resolution}` and `{count}`.
- **“{resolution}” (Optional)** : If you want Elastic Transcoder to include the resolution in the file name, include `{resolution}` in the `ThumbnailPattern` object. When creating thumbnails, Elastic Transcoder automatically saves the files in the format (.jpg or .png) that appears in the preset that you specified in the `PresetID` value of `CreateJobOutput`. Elastic Transcoder also appends the applicable file name extension.

- **ThumbnailEncryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your thumbnail.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3**: Amazon S3 creates and manages the keys used for encrypting your files.

- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Rotate** (*string*) –

The number of degrees clockwise by which you want Elastic Transcoder to rotate the output relative to the input. Enter one of the following values:

`auto, 0, 90, 180, 270`

The value `auto` generally works only if the file that you're transcoding contains rotation metadata.

- **PresetId** (*string*) –

The value of the `Id` object for the preset that you want to use for this job. The preset determines the audio, video, and thumbnail settings that Elastic Transcoder uses for transcoding. To use a preset that you created, specify the preset ID that Elastic Transcoder returned in the response when you created the preset. You can also use the Elastic Transcoder system presets, which you can get with `ListPresets`.

- **SegmentDuration** (*string*) –

Warning: (Outputs in Fragmented MP4 or MPEG-TS format only.

If you specify a preset in `PresetId` for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `SegmentDuration` is the target maximum duration of each segment in seconds. For `HLSv3` format playlists, each media segment is stored in a separate `.ts` file. For `HLSv4` and `Smooth` playlists, all media segments for an output are stored in a single file. Each segment is approximately the length of the `SegmentDuration`, though individual segments might be shorter or longer.

The range of valid values is 1 to 60 seconds. If the duration of the video is not evenly divisible by `SegmentDuration`, the duration of the last segment is the remainder of `total length/SegmentDuration`.

Elastic Transcoder creates an output-specific playlist for each output `HLS` output that you specify in `OutputKeys`. To add an output to the master playlist for this job, include it in the `OutputKeys` of the associated playlist.

- **Status** (*string*) –

The status of one output in a job. If you specified only one output for the job, `Outputs:Status` is always the same as `Job:Status`. If you specified more than one output:

- `Job:Status` and `Outputs:Status` for all of the outputs is `Submitted` until Elastic Transcoder starts to process the first output.
- When Elastic Transcoder starts to process the first output, `Outputs:Status` for that output and `Job:Status` both change to `Progressing`. For each output, the value of `Outputs:Status` remains `Submitted` until Elastic Transcoder starts to process the output.
- `Job:Status` remains `Progressing` until all of the outputs reach a terminal status, either `Complete` or `Error`.
- When all of the outputs reach a terminal status, `Job:Status` changes to `Complete` only if `Outputs:Status` for all of the outputs is `Complete`. If `Outputs:Status` for one or more outputs is `Error`, the

terminal status for `Job:Status` is also `Error`.

The value of `Status` is one of the following: `Submitted`, `Progressing`, `Complete`, `Canceled`, or `Error`.

- **StatusDetail** (*string*) –

Information that further explains `Status`.

- **Duration** (*integer*) –

Duration of the output file, in seconds.

- **Width** (*integer*) –

Specifies the width of the output file in pixels.

- **Height** (*integer*) –

Height of the output file, in pixels.

- **FrameRate** (*string*) –

Frame rate of the output file, in frames per second.

- **FileSize** (*integer*) –

File size of the output file, in bytes.

- **DurationMillis** (*integer*) –

Duration of the output file, in milliseconds.

- **Watermarks** (*list*) –

Information about the watermarks that you want Elastic Transcoder to add to the video during transcoding. You can specify up to four watermarks for each output. Settings for each watermark must be defined in the preset that you specify in `Preset` for the current output.

Watermarks are added to the output video in the sequence in which you list them in the job output: the first watermark in the list is added to the output video first, the second watermark in the list is added next, and so on. As a result, if the settings in a preset cause Elastic Transcoder to place all watermarks in the same location, the second watermark that you add will cover the first one, the third one will cover the second, and the fourth one will cover the third.

- (*dict*) –

Watermarks can be in `.png` or `.jpg` format. If you want to display a watermark that is not rectangular, use the `.png` format, which supports transparency.

- **PresetWatermarkId** (*string*) –

The ID of the watermark settings that Elastic Transcoder uses to add watermarks to the video during transcoding. The settings are in the preset specified by `Preset` for the current output. In that preset, the value of `Watermarks Id` tells Elastic Transcoder which settings to use.

- **InputKey** (*string*) –

The name of the `.png` or `.jpg` file that you want to use for the watermark. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `Pipeline`; the `Input Bucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, **logos/128x64.png**, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your watermarks.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly

16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AlbumArt** (*dict*) –

The album art to be associated with the output file, if any.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder will handle the existence of multiple album artwork files.

- **Replace**: The specified album art will replace any existing album art.
- **Prepend**: The specified album art will be placed in front of any existing album art.
- **Append**: The specified album art will be placed after any existing album art.
- **Fallback**: If the original input file contains artwork, Elastic Transcoder will use that artwork for the output. If the original input does not contain artwork, Elastic Transcoder will use the specified album art file.

- **Artwork** (*list*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20. Valid formats are .jpg and .png

- (*dict*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20.

To remove artwork or leave the artwork empty, you can either set `Artwork` to null, or set the `Merge Policy` to “Replace” and use an empty `Artwork` array.

To pass through existing artwork unchanged, set the `Merge Policy` to “Prepend”, “Append”, or “Fallback”, and use an empty `Artwork` array.

- **InputKey** (*string*) –

The name of the file to be used as album art. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/pie.jpg`, include the prefix in the key. If the file isn’t in the specified bucket, Elastic Transcoder returns an error.

- **MaxWidth** (*string*) –

The maximum width of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default

value. If you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

- **MaxHeight** (*string*) –

The maximum height of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.

- **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of the output album art:

- **Fit**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Fill**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output art and then crops it in the dimension (if any) that exceeds the maximum value.
- **Stretch**: Elastic Transcoder stretches the output art to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input art and the output art are different, the output art will be distorted.
- **Keep**: Elastic Transcoder does not scale the output art. If either dimension of the input art exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output art.
- **ShrinkToFit**: Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the art up.
- **ShrinkToFill**: Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the art up.

- **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder may add white bars to the top and bottom and/or left and right sides of the output album art to make the total size of the output art match the values that you specified for `MaxWidth` and `MaxHeight`.

- **AlbumArtFormat** (*string*) –

The format of album art, if any. Valid formats are `.jpg` and `.png`.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your artwork.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or en-

crypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Composition** (*list*) –

You can create an output file that contains an excerpt from the input file. This excerpt, called a clip, can come from the beginning, middle, or end of the file. The Composition object contains settings for the clips that make up an output file. For the current release, you can only specify settings for a single clip per output file. The Composition object cannot be null.

- **(dict)** –

Settings for one clip in a composition. All jobs in a playlist must have the same clip settings.

- **TimeSpan** (*dict*) –

Settings that determine when a clip begins and how long it lasts.

- **StartTime** (*string*) –

The place in the input file where you want a clip to start. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder starts at the beginning of the input file.

- **Duration** (*string*) –

The duration of the clip. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder creates an output file from StartTime to the end of the file.

If you specify a value longer than the duration of the input file, Elastic Transcoder transcodes the file and returns a warning message.

- **Captions** (*dict*) –

You can configure Elastic Transcoder to transcode captions, or subtitles, from one format to another. All captions must be in UTF-8. Elastic Transcoder supports two types of captions:

- **Embedded:** Embedded captions are included in the same file as the audio and video. Elastic Transcoder supports only one embedded caption per language, to a maximum of 300 embedded captions per file. Valid input values include: CEA-608 (EIA-608 , first non-empty channel only), CEA-708 (EIA-708 , first non-empty channel only), and mov-text. Valid outputs include: mov-text. Elastic Transcoder supports a maximum of one embedded format per output.
- **Sidecar:** Sidecar captions are kept in a separate metadata file from the audio and video data. Sidecar captions require a player that is capable of understanding the relationship between the video file and the sidecar file. Elastic Transcoder supports only one sidecar caption per language, to a maximum of 20 sidecar captions per file. Valid input values include: dfxp (first div element only), ebu-tt , scc , smpt , srt , ttml (first div element only), and webvtt. Valid outputs include: dfxp (first div element only), scc , srt , and webvtt .

If you want `ttml` or `smpte-tt` compatible captions, specify `dfxp` as your output format.

Elastic Transcoder does not support OCR (Optical Character Recognition), does not accept pictures as a valid input for captions, and is not available for audio-only transcoding. Elastic Transcoder does not preserve text formatting (for example, italics) during the transcoding process.

To remove captions or leave the captions empty, set `Captions` to `null`. To pass through existing captions unchanged, set the `MergePolicy` to `MergeRetain`, and pass in a `null` `CaptionSources` array.

For more information on embedded files, see the Subtitles Wikipedia page.

For more information on sidecar files, see the Extensible Metadata Platform and Sidecar file Wikipedia pages.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder handles the existence of multiple captions.

- **MergeOverride:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the sidecar captions and ignores the embedded captions for that language.

- **MergeRetain:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the embedded captions and ignores the sidecar captions for that language. If `CaptionSources` is empty, Elastic Transcoder omits all sidecar captions from the output files.

- **Override:** Elastic Transcoder transcodes only the sidecar captions that you specify in `CaptionSources`. `MergePolicy` cannot be `null`.

- **CaptionSources** (*list*) –

Source files for the input sidecar captions used during the transcoding process. To omit all sidecar captions, leave `CaptionSources` blank.

- (*dict*) –

A source file for the input sidecar captions used during the transcoding process.

- **Key** (*string*) –

The name of the sidecar caption file that you want Elastic Transcoder to include in the output file.

- **Language** (*string*) –

A string that specifies the language of the caption. Specify this as one of:

- 2-character ISO 639-1 code
- 3-character ISO 639-2 code

For more information on ISO language codes and language names, see the List of ISO 639-1 codes.

- **TimeOffset** (*string*) –

For clip generation or captions that do not start at the same time as the associated video file, the `TimeOffset` tells Elastic Transcoder how much of the video to encode before including captions.

Specify the `TimeOffset` in the form `[+-]SS.sss` or `[+-]HH:mm:SS.ss`.

- **Label** (*string*) –

The label of the caption shown in the player when choosing a language. We recommend that you put the caption language name here, in the language of the captions.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption sources.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it

must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **CaptionFormats** (*list*) –

The array of file formats for the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- (*dict*) –

The file format of the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- **Format** (*string*) –

The format you specify determines whether Elastic Transcoder generates an embedded or sidecar caption for this output.

- **Valid Embedded Caption Formats:** * **for FLAC** : None

- **For MP3** : None

- **For MP4** : mov-text

- **For MPEG-TS** : None

- **For ogg** : None

- **For webm** : None

- **Valid Sidecar Caption Formats:** Elastic Transcoder supports dfxp (first div element only), scc, srt, and webvtt. If you want ttml or smpte-tt compatible captions, specify dfxp as your output format. * **For FMP4** : dfxp

- **Non-FMP4 outputs** : All sidecar types

fm4 captions have an extension of .ismt

- **Pattern** (*string*) –

The prefix for caption filenames, in the form *description* - {language} , where:

- *description* is a description of the video.

- {language} is a literal value that Elastic Transcoder replaces with the two- or three-letter code for the language of the caption in the output file names.

If you don't include {language} in the file name pattern, Elastic Transcoder automatically appends "{language} "

to the value that you specify for the description. In addition, Elastic Transcoder automatically appends the count to the end of the segment files.

For example, suppose you're transcoding into srt format. When you enter "Sydney-{language}-sunrise", and the language of the captions is English (en), the name of the first caption file will be Sydney-en-sunrise00000.srt.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption formats.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify S3-AWS-KMS and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your output files. If you choose to use encryption, you must specify a mode to use. If you choose not to use encryption, Elastic Transcoder will write an unencrypted file to your Amazon S3 bucket.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to en-

crypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AppliedColorSpaceConversion** (*string*) –

If Elastic Transcoder used a preset with a `ColorSpaceConversionMode` to transcode the output file, the `AppliedColorSpaceConversion` parameter shows the conversion used. If no `ColorSpaceConversionMode` was defined in the preset, this parameter will not be included in the job response.

- **OutputKeyPrefix** (*string*) –

The value, if any, that you want Elastic Transcoder to prepend to the names of all files that this job creates, including output files, thumbnails, and playlists. We recommend that you add a / or some other delimiter to the end of the `OutputKeyPrefix`.

- **Playlists** (*list*) –

Warning: Outputs in Fragmented MP4 or MPEG-TS format only.
--

If you specify a preset in `PresetId` for which the value of `Container` is `fmp4` (Fragmented MP4) or `ts` (MPEG-TS), `Playlists` contains information about the master playlists that you want Elastic Transcoder to create.

The maximum number of master playlists in a job is 30.

- (*dict*) –

Use Only for Fragmented MP4 or MPEG-TS Outputs. If you specify a preset for which the value of `Container` is `fmp4` (Fragmented MP4) or `ts` (MPEG-TS), `Playlists` contains information about the master playlists that you want Elastic Transcoder to create. We recommend that you create only

one master playlist per output format. The maximum number of master playlists in a job is 30.

- **Name** (*string*) –

The name that you want Elastic Transcoder to assign to the master playlist, for example, nyc-vacation.m3u8. If the name includes a / character, the section of the name before the last / must be identical for all Name objects. If you create more than one master playlist, the values of all Name objects must be unique.

Note : Elastic Transcoder automatically appends the relevant file extension to the file name (.m3u8 for HLSv3 and HLSv4 playlists, and .ism and .ismc for Smooth playlists). If you include a file extension in Name, the file name will have two extensions.

- **Format** (*string*) –

The format of the output playlist. Valid formats include HLSv3, HLSv4, and Smooth.

- **OutputKeys** (*list*) –

For each output in this job that you want to include in a master playlist, the value of the Outputs:Key object.

- If your output is not HLS or does not have a segment duration set, the name of the output file is a concatenation of OutputKeyPrefix and Outputs:Key: OutputKeyPrefix“Outputs:Key“
- If your output is HLSv3 and has a segment duration set, or is not included in a playlist, Elastic Transcoder creates an output playlist file with a file extension of .m3u8, and a series of .ts files that include a five-digit sequential counter beginning with 00000: OutputKeyPrefix“Outputs:Key“ .m3u8 OutputKeyPrefix“Outputs:Key“ 00000.ts
- If your output is HLSv4, has a segment duration set, and is included in an HLSv4 playlist, Elastic Transcoder creates an output playlist file with a file extension of _v4.m3u8. If the output is video, Elastic Transcoder also creates an output file with an extension of _iframe.m3u8: OutputKeyPrefix“Outputs:Key“ _v4.m3u8 OutputKeyPrefix“Outputs:Key“ _iframe.m3u8 OutputKeyPrefix“Outputs:Key“ .ts

Elastic Transcoder automatically appends the relevant file extension to the file name. If you include a file extension in Output Key, the file name will have two extensions.

If you include more than one output in a playlist, any segment duration settings, clip settings, or caption settings must be the same for all outputs in the playlist. For Smooth playlists, the Audio:Profile, Video:Profile, and Video:FrameRate to Video:KeyframesMaxDist ratio must be the same for all outputs.

- (*string*) –

- **HlsContentProtection** (*dict*) –

The HLS content protection settings, if any, that you want Elastic Transcoder to apply to the output files associated with

this playlist.

- **Method** (*string*) –

The content protection method for your output. The only valid value is: `aes-128`.

This value will be written into the method attribute of the `EXT-X-KEY` metadata tag in the output playlist.

- **Key** (*string*) –

If you want Elastic Transcoder to generate a key for you, leave this field blank.

If you choose to supply your own key, you must encrypt the key by using AWS KMS. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

`128`, `192`, or `256`.

- **KeyMd5** (*string*) –

If Elastic Transcoder is generating your key for you, you must leave this field blank.

The MD5 digest of the key that you want Elastic Transcoder to use to encrypt your output file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

- **InitializationVector** (*string*) –

If Elastic Transcoder is generating your key for you, you must leave this field blank.

The series of random bits created by a random bit generator, unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

- **LicenseAcquisitionUrl** (*string*) –

The location of the license key required to decrypt your HLS playlist. The URL must be an absolute path, and is referenced in the URI attribute of the `EXT-X-KEY` metadata tag in the playlist file.

- **KeyStoragePolicy** (*string*) –

Specify whether you want Elastic Transcoder to write your HLS license key to an Amazon S3 bucket. If you choose `WithVariantPlaylists`, `LicenseAcquisitionUrl` must be left blank and Elastic Transcoder writes your data key into the same bucket as the associated playlist.

- **PlayReadyDrm** (*dict*) –

The DRM settings, if any, that you want Elastic Transcoder to apply to the output files associated with this playlist.

- **Format** (*string*) –

The type of DRM, if any, that you want Elastic Transcoder to apply to the output files associated with this playlist.

- **Key** (*string*) –

The DRM key for your file, provided by your DRM license provider. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using AWS KMS.

- **KeyMd5** (*string*) –

The MD5 digest of the key used for DRM on your file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

- **KeyId** (*string*) –

The ID for your DRM key, so that your DRM license provider knows which key to provide.

The key ID must be provided in big endian, and Elastic Transcoder will convert it to little endian before inserting it into the PlayReady DRM headers. If you are unsure whether your license server provides your key ID in big or little endian, check with your DRM provider.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your files. The initialization vector must be base64-encoded, and it must be exactly 8 bytes long before being base64-encoded. If no initialization vector is provided, Elastic Transcoder generates one for you.

- **LicenseAcquisitionUrl** (*string*) –

The location of the license key required to play DRM content. The URL must be an absolute path, and is referenced by the PlayReady header. The PlayReady header is referenced in the protection header of the client manifest for Smooth Streaming outputs, and in the EXT-X-DXDRM and EXT-XDXDRMINFO metadata tags for HLS playlist outputs. An example URL looks like this: <https://www.example.com/exampleKey/>

- **Status** (*string*) –

The status of the job with which the playlist is associated.

- **StatusDetail** (*string*) –

Information that further explains the status.

- **Status** (*string*) –

The status of the job: Submitted , Progressing , Complete , Canceled , or Error .

- **UserMetadata** (*dict*) –

User-defined metadata that you want to associate with an Elastic Transcoder job. You specify metadata in key/value

pairs, and you can add up to 10 `key/value` pairs per job. Elastic Transcoder does not guarantee that `key/value` pairs will be returned in the same order in which you specify them.

Metadata keys and values must use characters from the following list:

- 0–9
- A–Z and a–z
- Space
- The following symbols: `_ . : / = + - % @`
- *(string)* –
- *(string)* –
- **Timing** *(dict)* –

Details about the timing of a job.

- **SubmitTimeMillis** *(integer)* –

The time the job was submitted to Elastic Transcoder, in epoch milliseconds.

- **StartTimeMillis** *(integer)* –

The time the job began transcoding, in epoch milliseconds.

- **FinishTimeMillis** *(integer)* –

The time the job finished transcoding, in epoch milliseconds.

- **NextPageToken** *(string)* –

A value that you use to access the second and subsequent pages of results, if any. When the jobs in the specified pipeline fit on one page or when you've reached the last page of results, the value of `NextPageToken` is `null`.

list_pipelines (***kwargs*)

The `ListPipelines` operation gets a list of the pipelines associated with the current AWS account.

Request Syntax

```
response = client.list_pipelines(
    Ascending='string',
    PageToken='string'
)
```

Parameters

- **Ascending** *(string)* – To list pipelines in chronological order by the date and time that they were created, enter `true`. To list pipelines in reverse chronological order, enter `false`.
- **PageToken** *(string)* – When Elastic Transcoder returns more than one page of results, use `pageToken` in subsequent GET requests to get each successive page of results.

Return type dict

Returns

Response Syntax

```
{
  'Pipelines': [
    {
      'Id': 'string',
      'Arn': 'string',
```

```
    'Name': 'string',
    'Status': 'string',
    'InputBucket': 'string',
    'OutputBucket': 'string',
    'Role': 'string',
    'AwsKmsKeyArn': 'string',
    'Notifications': {
        'Progressing': 'string',
        'Completed': 'string',
        'Warning': 'string',
        'Error': 'string'
    },
    'ContentConfig': {
        'Bucket': 'string',
        'StorageClass': 'string',
        'Permissions': [
            {
                'GranteeType': 'string',
                'Grantee': 'string',
                'Access': [
                    'string',
                ]
            },
        ]
    },
    'ThumbnailConfig': {
        'Bucket': 'string',
        'StorageClass': 'string',
        'Permissions': [
            {
                'GranteeType': 'string',
                'Grantee': 'string',
                'Access': [
                    'string',
                ]
            },
        ]
    },
    ],
    'NextPageToken': 'string'
}
```

Response Structure

- (*dict*) –

A list of the pipelines associated with the current AWS account.

- **Pipelines** (*list*) –

An array of Pipeline objects.

- * (*dict*) –

The pipeline (queue) that is used to manage jobs.

- **Id** (*string*) –

The identifier for the pipeline. You use this value to identify the pipeline in which you want to perform a variety of operations, such as creating a job or a preset.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) for the pipeline.

- **Name** (*string*) –

The name of the pipeline. We recommend that the name be unique within the AWS account, but uniqueness is not enforced.

Constraints: Maximum 40 characters

- **Status** (*string*) –

The current status of the pipeline:

- **Active** : The pipeline is processing jobs.
- **Paused** : The pipeline is not currently processing jobs.
- **InputBucket** (*string*) –

The Amazon S3 bucket from which Elastic Transcoder gets media files for transcoding and the graphics files, if any, that you want to use for watermarks.

- **OutputBucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files, thumbnails, and playlists. Either you specify this value, or you specify both `ContentConfig` and `ThumbnailConfig`.

- **Role** (*string*) –

The IAM Amazon Resource Name (ARN) for the role that Elastic Transcoder uses to transcode jobs for this pipeline.

- **AwsKmsKeyArn** (*string*) –

The AWS Key Management Service (AWS KMS) key that you want to use with this pipeline.

If you use either `S3` or `S3-AWS-KMS` as your `Encryption:Mode`, you don't need to provide a key with your job because a default key, known as an AWS-KMS key, is created for you automatically. You need to provide an AWS-KMS key only if you want to use a non-default AWS-KMS key, or if you are using an `Encryption:Mode` of `AES-PKCS7`, `AES-CTR`, or `AES-GCM`.

- **Notifications** (*dict*) –

The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify to report job status.

Warning: To receive notifications, you must also subscribe to the new topic in the Amazon SNS console.

- **Progressing** (optional): The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process the job.
- **Completed** (optional): The Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing the job.
- **Warning** (optional): The Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition.
- **Error** (optional): The Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition.

- **Progressing** (*string*) –

The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process the job.

- **Completed** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing the job.

- **Warning** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition.

- **Error** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition.

- **ContentConfig** (*dict*) –

Information about the Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists. Either you specify both `ContentConfig` and `ThumbnailConfig`, or you specify `OutputBucket`.

- **Bucket** : The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists.

- **Permissions** : A list of the users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access that you want them to have. * **GranteeType**: The type of value that appears in the `Grantee` object: * **Canonical** : Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution.

- **Email** : The registered email address of an AWS account.

- **Group** : One of the following predefined Amazon S3 groups: `AllUsers` , `AuthenticatedUsers` , or `LogDelivery` .

- **Grantee** : The AWS user or group that you want to have access to transcoded files and playlists.

- **Access** : The permission that you want to give to the AWS user that is listed in `Grantee` . Valid values include: * **READ** : The grantee can read the objects and metadata for objects that Elastic Transcoder adds to the Amazon S3 bucket.

- **READ_ACP** : The grantee can read the object ACL for objects that Elastic Transcoder adds to the Amazon S3 bucket.

- **WRITE_ACP** : The grantee can write the ACL for the objects that Elastic Transcoder adds to the Amazon S3 bucket.

- **FULL_CONTROL** : The grantee has `READ` , `READ_ACP` , and `WRITE_ACP` permissions for the objects that Elastic Transcoder adds to the Amazon S3 bucket.

- **StorageClass** : The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

- **Bucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. Specify this value when all of the

following are true:

- You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.
- You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- You do not want to specify the permissions that Elastic Transcoder grants to the files.
- You want to associate the transcoded files and thumbnails with the Amazon S3 Standard storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit `OutputBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

- **StorageClass** (*string*) –

The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

- **Permissions** (*list*) –

Optional. The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions`, Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role`. If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions`, Elastic Transcoder grants full control over the transcoded files and playlists to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

- (*dict*) –

The `Permission` structure.

- **GranteeType** (*string*) –

The type of value that appears in the `Grantee` object:

- `Canonical`: Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.
- `Email`: The registered email address of an AWS account.
- `Group`: One of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.
- **Grantee** (*string*) –

The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group,

you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group.

- **Access** (*list*) –

The permission that you want to give to the AWS user that is listed in Grantee. Valid values include:

- **READ** : The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **READ_ACP** : The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **WRITE_ACP** : The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **FULL_CONTROL** : The grantee has **READ**, **READ_ACP**, and **WRITE_ACP** permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.

- (*string*) –

- **ThumbnailConfig** (*dict*) –

Information about the Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files. Either you specify both `ContentConfig` and `ThumbnailConfig`, or you specify `OutputBucket`.

- **Bucket** : The Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files.
- **Permissions** : A list of the users and/or predefined Amazon S3 groups you want to have access to thumbnail files, and the type of access that you want them to have. * **GranteeType**: The type of value that appears in the Grantee object: * **Canonical** : Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.
- **Email** : The registered email address of an AWS account.
- **Group** : One of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.
- **Grantee** : The AWS user or group that you want to have access to thumbnail files.
- **Access**: The permission that you want to give to the AWS user that is listed in Grantee. Valid values include: * **READ** : The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **READ_ACP** : The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **WRITE_ACP** : The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **FULL_CONTROL** : The grantee has **READ**, **READ_ACP**, and **WRITE_ACP** permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **StorageClass** : The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the thumbnails that it stores in your

Amazon S3 bucket.

- **Bucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. Specify this value when all of the following are true:

- You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.
- You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- You do not want to specify the permissions that Elastic Transcoder grants to the files.
- You want to associate the transcoded files and thumbnails with the Amazon S3 Standard storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit `OutputBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

- **StorageClass** (*string*) –

The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

- **Permissions** (*list*) –

Optional. The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions`, Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role`. If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions`, Elastic Transcoder grants full control over the transcoded files and playlists to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

- (*dict*) –

The `Permission` structure.

- **GranteeType** (*string*) –

The type of value that appears in the `Grantee` object:

- `Canonical`: Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning:: A canonical user ID is not the same as an AWS account number.
- `Email`: The registered email address of an AWS account.
- `Group`: One of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or

LogDelivery.

- **Grantee** (*string*) –

The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group.

- **Access** (*list*) –

The permission that you want to give to the AWS user that is listed in Grantee. Valid values include:

- **READ** : The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **READ_ACP** : The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **WRITE_ACP** : The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **FULL_CONTROL** : The grantee has **READ**, **READ_ACP**, and **WRITE_ACP** permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- (*string*) –

- **NextPageToken** (*string*) –

A value that you use to access the second and subsequent pages of results, if any. When the pipelines fit on one page or when you’ve reached the last page of results, the value of `NextPageToken` is `null`.

list_presets (***kwargs*)

The `ListPresets` operation gets a list of the default presets included with Elastic Transcoder and the presets that you’ve added in an AWS region.

Request Syntax

```
response = client.list_presets(
    Ascending='string',
    PageToken='string'
)
```

Parameters

- **Ascending** (*string*) – To list presets in chronological order by the date and time that they were created, enter `true`. To list presets in reverse chronological order, enter `false`.
- **PageToken** (*string*) – When Elastic Transcoder returns more than one page of results, use `pageToken` in subsequent `GET` requests to get each successive page of results.

Return type dict

Returns

Response Syntax

```
{
    'Presets': [
        {
            'Id': 'string',
```

```

'Arn': 'string',
'Name': 'string',
'Description': 'string',
'Container': 'string',
'Audio': {
    'Codec': 'string',
    'SampleRate': 'string',
    'BitRate': 'string',
    'Channels': 'string',
    'AudioPackingMode': 'string',
    'CodecOptions': {
        'Profile': 'string',
        'BitDepth': 'string',
        'BitOrder': 'string',
        'Signed': 'string'
    }
},
'Video': {
    'Codec': 'string',
    'CodecOptions': {
        'string': 'string'
    },
    'KeyframesMaxDist': 'string',
    'FixedGOP': 'string',
    'BitRate': 'string',
    'FrameRate': 'string',
    'MaxFrameRate': 'string',
    'Resolution': 'string',
    'AspectRatio': 'string',
    'MaxWidth': 'string',
    'MaxHeight': 'string',
    'DisplayAspectRatio': 'string',
    'SizingPolicy': 'string',
    'PaddingPolicy': 'string',
    'Watermarks': [
        {
            'Id': 'string',
            'MaxWidth': 'string',
            'MaxHeight': 'string',
            'SizingPolicy': 'string',
            'HorizontalAlign': 'string',
            'HorizontalOffset': 'string',
            'VerticalAlign': 'string',
            'VerticalOffset': 'string',
            'Opacity': 'string',
            'Target': 'string'
        }
    ],
},
'Thumbnails': {
    'Format': 'string',
    'Interval': 'string',
    'Resolution': 'string',
    'AspectRatio': 'string',
    'MaxWidth': 'string',
    'MaxHeight': 'string',
    'SizingPolicy': 'string',
    'PaddingPolicy': 'string'
}

```

```
        },
        'Type': 'string'
    },
    ],
    'NextPageToken': 'string'
}
```

Response Structure

- *(dict)* –

The `ListPresetsResponse` structure.

- **Presets** (*list*) –

An array of `Preset` objects.

- * *(dict)* –

Presets are templates that contain most of the settings for transcoding media files from one format to another. Elastic Transcoder includes some default presets for common formats, for example, several iPod and iPhone versions. You can also create your own presets for formats that aren't included among the default presets. You specify which preset you want to use when you create a job.

- **Id** (*string*) –

Identifier for the new preset. You use this value to get settings for the preset or to delete it.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) for the preset.

- **Name** (*string*) –

The name of the preset.

- **Description** (*string*) –

A description of the preset.

- **Container** (*string*) –

The container type for the output file. Valid values include `flac`, `flv`, `fmp4`, `gif`, `mp3`, `mp4`, `mpg`, `mxf`, `oga`, `ogg`, `ts`, and `webm`.

- **Audio** (*dict*) –

A section of the response body that provides information about the audio preset values.

- **Codec** (*string*) –

The audio codec for the output file. Valid values include `aac`, `flac`, `mp2`, `mp3`, `pcm`, and `vorbis`.

- **SampleRate** (*string*) –

The sample rate of the audio stream in the output file, in Hertz. Valid values include:

`auto`, `22050`, `32000`, `44100`, `48000`, `96000`

If you specify `auto`, Elastic Transcoder automatically detects the sample rate.

- **BitRate** (*string*) –

The bit rate of the audio stream in the output file, in kilobits/second. Enter an integer between 64 and 320, inclusive.

- **Channels** (*string*) –

The number of audio channels in the output file. The following values are valid:

`auto`, `0`, `1`, `2`

One channel carries the information played by a single speaker. For example, a stereo track with two channels sends one channel to the left speaker, and the other channel to the right speaker. The output channels are organized into tracks. If you want Elastic Transcoder to automatically detect the number of audio channels in the input file and use that value for the output file, select `auto`.

The output of a specific channel value and inputs are as follows:

- **auto channel specified, with any input:** Pass through up to eight input channels.
- **0 channels specified, with any input:** Audio omitted from the output.
- **1 channel specified, with at least one input channel:** Mono sound.
- **2 channels specified, with any input:** Two identical mono channels or stereo. For more information about tracks, see `Audio:AudioPackingMode`.

For more information about how Elastic Transcoder organizes channels and tracks, see `Audio:AudioPackingMode`.

- **AudioPackingMode** (*string*) –

The method of organizing audio channels and tracks. Use `Audio:Channels` to specify the number of channels in your output, and `Audio:AudioPackingMode` to specify the number of tracks and their relation to the channels. If you do not specify an `Audio:AudioPackingMode`, Elastic Transcoder uses `SingleTrack`.

The following values are valid:

`SingleTrack`, `OneChannelPerTrack`, and `OneChannelPerTrackWithMostTo8Tracks`

When you specify `SingleTrack`, Elastic Transcoder creates a single track for your output. The track can have up to eight channels. Use `SingleTrack` for all non-mxf containers.

The outputs of `SingleTrack` for a specific channel value and inputs are as follows:

- **0 channels with any input:** Audio omitted from the output
- **1, 2, or auto channels with no audio input:** Audio omitted from the output
- **1 channel with any input with audio:** One track with one channel, downmixed if necessary
- **2 channels with one track with one channel:** One track with two identical channels
- **2 or auto channels with two tracks with one channel each:** One track with two channels

- **2 or auto channels with one track with two channels:** One track with two channels
- **2 channels with one track with multiple channels:** One track with two channels
- **auto channels with one track with one channel:** One track with one channel

- **auto channels with one track with multiple channels:** One track with multiple channels

When you specify `OneChannelPerTrack`, Elastic Transcoder creates a new track for every channel in your output. Your output can have up to eight single-channel tracks.

The outputs of `OneChannelPerTrack` for a specific channel value and inputs are as follows:

- **0 channels with any input:** Audio omitted from the output
- **1, 2, or auto channels with no audio input:** Audio omitted from the output
- **1 channel with any input with audio:** One track with one channel, downmixed if necessary
- **2 channels with one track with one channel:** Two tracks with one identical channel each
- **2 or auto channels with two tracks with one channel each:** Two tracks with one channel each
- **2 or auto channels with one track with two channels:** Two tracks with one channel each
- **2 channels with one track with multiple channels:** Two tracks with one channel each
- **auto channels with one track with one channel:** One track with one channel
- **auto channels with one track with multiple channels:** Up to eight tracks with one channel each

When you specify `OneChannelPerTrackWithMosTo8Tracks`, Elastic Transcoder creates eight single-channel tracks for your output. All tracks that do not contain audio data from an input channel are MOS, or Mit Out Sound, tracks.

The outputs of `OneChannelPerTrackWithMosTo8Tracks` for a specific channel value and inputs are as follows:

- **0 channels with any input:** Audio omitted from the output
- **1, 2, or auto channels with no audio input:** Audio omitted from the output
- **1 channel with any input with audio:** One track with one channel, downmixed if necessary, plus six MOS tracks
- **2 channels with one track with one channel:** Two tracks with one identical channel each, plus six MOS tracks
- **2 or auto channels with two tracks with one channel each:** Two tracks with one channel each, plus six MOS tracks
- **2 or auto channels with one track with two channels:** Two tracks with one channel each, plus six MOS tracks
- **2 channels with one track with multiple channels:** Two tracks with one channel each, plus six MOS tracks
- **auto channels with one track with one channel:** One track with one channel, plus seven MOS tracks
- **auto channels with one track with multiple channels:** Up to eight tracks with one channel each, plus MOS tracks until

there are eight tracks in all

- **CodecOptions** (*dict*) –

If you specified AAC for `Audio:Codec`, this is the AAC compression profile to use. Valid values include:

`auto`, `AAC-LC`, `HE-AAC`, `HE-AACv2`

If you specify `auto`, Elastic Transcoder chooses a profile based on the bit rate of the output file.

- **Profile** (*string*) –

You can only choose an audio profile when you specify AAC for the value of `Audio:Codec`.

Specify the AAC profile for the output file. Elastic Transcoder supports the following profiles:

- `auto` : If you specify `auto`, Elastic Transcoder will select the profile based on the bit rate selected for the output file.
- `AAC-LC` : The most common AAC profile. Use for bit rates larger than 64 kbps.
- `HE-AAC` : Not supported on some older players and devices. Use for bit rates between 40 and 80 kbps.
- `HE-AACv2` : Not supported on some players and devices. Use for bit rates less than 48 kbps.

All outputs in a `Smooth` playlist must have the same value for `Profile`.

Note: If you created any presets before AAC profiles were added, Elastic Transcoder automatically updated your presets to use AAC-LC. You can change the value as required.

- **BitDepth** (*string*) –

You can only choose an audio bit depth when you specify `flac` or `pcm` for the value of `Audio:Codec`.

The bit depth of a sample is how many bits of information are included in the audio samples. The higher the bit depth, the better the audio, but the larger the file.

Valid values are `16` and `24`.

The most common bit depth is `24`.

- **BitOrder** (*string*) –

You can only choose an audio bit order when you specify `pcm` for the value of `Audio:Codec`.

The order the bits of a PCM sample are stored in.

The supported value is `LittleEndian`.

- **Signed** (*string*) –

You can only choose whether an audio sample is signed when you specify `pcm` for the value of `Audio:Codec`.

Whether audio samples are represented with negative and positive numbers (signed) or only positive numbers (unsigned).

The supported value is `Signed`.

- **Video** (*dict*) –

A section of the response body that provides information about the video preset values.

- **Codec** (*string*) –

The video codec for the output file. Valid values include `gif`, `H.264`, `mpeg2`, and `vp8`. You can only specify `vp8` when the container type is `webm`, `gif` when the container type is `gif`, and `mpeg2` when the container type is `mpg`.

- **CodecOptions** (*dict*) –

- Profile (H.264/VP8 Only)**

The H.264 profile that you want to use for the output file. Elastic Transcoder supports the following profiles:

- `baseline`: The profile most commonly used for videoconferencing and for mobile applications.
- `main`: The profile used for standard-definition digital TV broadcasts.
- `high`: The profile used for high-definition digital TV broadcasts and for Blu-ray discs.

- Level (H.264 Only)**

The H.264 level that you want to use for the output file. Elastic Transcoder supports the following levels:

`1`, `1b`, `1.1`, `1.2`, `1.3`, `2`, `2.1`, `2.2`, `3`, `3.1`, `3.2`, `4`, `4.1`

- MaxReferenceFrames (H.264 Only)**

Applicable only when the value of `Video:Codec` is `H.264`. The maximum number of previously decoded frames to use as a reference for decoding future frames. Valid values are integers 0 through 16, but we recommend that you not use a value greater than the following:

```
Min(Floor(Maximum decoded
picture buffer in macroblocks *
256 / (Width in pixels * Height
in pixels)), 16)
```

where *Width in pixels* and *Height in pixels* represent either `MaxWidth` and `MaxHeight`, or `Resolution`. *Maximum decoded picture buffer in macroblocks* depends on the value of the `Level` object. See the list below. (A macroblock is a block of pixels measuring 16x16.)

- `1` - 396
- `1b` - 396
- `1.1` - 900
- `1.2` - 2376
- `1.3` - 2376
- `2` - 2376
- `2.1` - 4752
- `2.2` - 8100
- `3` - 8100
- `3.1` - 18000
- `3.2` - 20480
- `4` - 32768
- `4.1` - 32768

MaxBitRate (Optional, H.264/MPEG2/VP8 only)

The maximum number of bits per second in a video buffer; the size of the buffer is specified by `BufferSize`. Specify a value between 16 and 62,500. You can reduce the bandwidth required to stream a video by reducing the maximum bit rate, but this also reduces the quality of the video.

BufferSize (Optional, H.264/MPEG2/VP8 only)

The maximum number of bits in any x seconds of the output video. This window is commonly 10 seconds, the standard segment duration when you're using FMP4 or MPEG-TS for the container type of the output video. Specify an integer greater than 0. If you specify `MaxBitRate` and omit `BufferSize`, Elastic Transcoder sets `BufferSize` to 10 times the value of `MaxBitRate`.

InterlacedMode (Optional, H.264/MPEG2 Only)

The interlace mode for the output video.

Interlaced video is used to double the perceived frame rate for a video by interlacing two fields (one field on every other line, the other field on the other lines) so that the human eye registers multiple pictures per frame. Interlacing reduces the bandwidth required for transmitting a video, but can result in blurred images and flickering.

Valid values include `Progressive` (no interlacing, top to bottom), `TopFirst` (top field first), `BottomFirst` (bottom field first), and `Auto`.

If `InterlaceMode` is not specified, Elastic Transcoder uses `Progressive` for the output. If `Auto` is specified, Elastic Transcoder interlaces the output.

ColorSpaceConversionMode (Optional, H.264/MPEG2 Only)

The color space conversion Elastic Transcoder applies to the output video. Color spaces are the algorithms used by the computer to store information about how to render color. `Bt.601` is the standard for standard definition video, while `Bt.709` is the standard for high definition video.

Valid values include `None`, `Bt709toBt601`, `Bt601toBt709`, and `Auto`.

If you chose `Auto` for `ColorSpaceConversionMode` and your output is interlaced, your frame rate is one of 23.97, 24, 25, 29.97, 50, or 60, your `SegmentDuration` is null, and you are using one of the resolution changes from the list below, Elastic Transcoder applies the following color space conversions:

- *Standard to HD, 720x480 to 1920x1080* - Elastic Transcoder applies `Bt601ToBt709`
- *Standard to HD, 720x576 to 1920x1080* - Elastic Transcoder applies `Bt601ToBt709`

- *HD to Standard, 1920x1080 to 720x480* - Elastic Transcoder applies Bt709ToBt601
- *HD to Standard, 1920x1080 to 720x576* - Elastic Transcoder applies Bt709ToBt601

Note: Elastic Transcoder may change the behavior of the `ColorSpaceConversionMode Auto` mode in the future. All outputs in a playlist must use the same `ColorSpaceConversionMode`.

If you do not specify a `ColorSpaceConversionMode`, Elastic Transcoder does not change the color space of a file. If you are unsure what `ColorSpaceConversionMode` was applied to your output file, you can check the `AppliedColorSpaceConversion` parameter included in your job response. If your job does not have an `AppliedColorSpaceConversion` in its response, no `ColorSpaceConversionMode` was applied.

ChromaSubsampling

The sampling pattern for the chroma (color) channels of the output video. Valid values include `yuv420p` and `yuv422p`.

`yuv420p` samples the chroma information of every other horizontal and every other vertical line, `yuv422p` samples the color information of every horizontal line and every other vertical line.

LoopCount (Gif Only)

The number of times you want the output gif to loop. Valid values include `Infinite` and integers between 0 and 100, inclusive.

- *(string)* –
- *(string)* –
- **KeyframesMaxDist** *(string)* –

Applicable only when the value of `Video:Codec` is one of `H.264`, `MPEG2`, or `VP8`.

The maximum number of frames between key frames. Key frames are fully encoded frames; the frames between key frames are encoded based, in part, on the content of the key frames. The value is an integer formatted as a string; valid values are between 1 (every frame is a key frame) and 100000, inclusive. A higher value results in higher compression but may also discernibly decrease video quality.

For Smooth outputs, the `FrameRate` must have a constant ratio to the `KeyframesMaxDist`. This allows Smooth playlists to switch between different quality levels while the file is being played.

For example, an input file can have a `FrameRate` of 30 with a `KeyframesMaxDist` of 90. The output file then needs to have a ratio of 1:3. Valid outputs would have `FrameRate` of 30, 25, and 10, and `KeyframesMaxDist` of 90, 75, and 30, respectively.

Alternately, this can be achieved by setting `FrameRate` to `auto` and having the same values for `MaxFrameRate` and `KeyframesMaxDist`.

- **FixedGOP** (*string*) –

Applicable only when the value of `Video:Codec` is one of `H.264`, `MPEG2`, or `VP8`.

Whether to use a fixed value for `FixedGOP`. Valid values are `true` and `false`:

- `true` : Elastic Transcoder uses the value of `KeyframesMaxDist` for the distance between key frames (the number of frames in a group of pictures, or GOP).
- `false` : The distance between key frames can vary.

Warning: `FixedGOP` must be set to `true` for `h264` containers.

- **BitRate** (*string*) –

The bit rate of the video stream in the output file, in kilobits/second. Valid values depend on the values of `Level` and `Profile`. If you specify `auto`, Elastic Transcoder uses the detected bit rate of the input source. If you specify a value other than `auto`, we recommend that you specify a value less than or equal to the maximum H.264-compliant value listed for your level and profile:

*Level - Maximum video bit rate in kilobits/second
(baseline and main Profile) : maximum video bit
rate in kilobits/second (high Profile)*

- 1 - 64 : 80
- 1b - 128 : 160
- 1.1 - 192 : 240
- 1.2 - 384 : 480
- 1.3 - 768 : 960
- 2 - 2000 : 2500
- 3 - 10000 : 12500
- 3.1 - 14000 : 17500
- 3.2 - 20000 : 25000
- 4 - 20000 : 25000
- 4.1 - 50000 : 62500
- **FrameRate** (*string*) –

The frames per second for the video stream in the output file. Valid values include:

`auto`, `10`, `15`, `23.97`, `24`, `25`, `29.97`, `30`, `60`

If you specify `auto`, Elastic Transcoder uses the detected frame rate of the input source. If you specify a frame rate, we recommend that you perform the following calculation:

Frame rate = maximum recommended
decoding speed in luma
samples/second / (width in
pixels * height in pixels)

where:

- *width in pixels* and *height in pixels* represent the Resolution of the output video.
- *maximum recommended decoding speed in Luma samples/second* is less than or equal to the maximum value listed in the following table, based on the value that you specified for Level.

The maximum recommended decoding speed in Luma samples/second for each level is described in the following list (*Level - Decoding speed*):

- 1 - 380160
- 1b - 380160
- 1.1 - 76800
- 1.2 - 1536000
- 1.3 - 3041280
- 2 - 3041280
- 2.1 - 5068800
- 2.2 - 5184000
- 3 - 10368000
- 3.1 - 27648000
- 3.2 - 55296000
- 4 - 62914560
- 4.1 - 62914560
- **MaxFrameRate** (*string*) –

If you specify `auto` for `FrameRate`, Elastic Transcoder uses the frame rate of the input video for the frame rate of the output video. Specify the maximum frame rate that you want Elastic Transcoder to use when the frame rate of the input video is greater than the desired maximum frame rate of the output video. Valid values include: 10, 15, 23.97, 24, 25, 29.97, 30, 60.

- **Resolution** (*string*) –

Warning: To better control resolution and aspect ratio of output videos, we recommend that you use the values `MaxWidth`, `MaxHeight`, `SizingPolicy`, `PaddingPolicy`, and `DisplayAspectRatio` instead of `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The width and height of the video in the output file, in pixels. Valid values are `auto` and *width x height*:

- `auto`: Elastic Transcoder attempts to preserve the width and height of the input file, subject to the following rules.
- **width* x *height**: The width and height of the output video in pixels.

Note the following about specifying the width and height:

- The width must be an even integer between 128 and 4096, inclusive.
- The height must be an even integer between 96 and 3072, inclusive.
- If you specify a resolution that is less than the resolution of the input file, Elastic Transcoder rescales the output file to the

lower resolution.

- If you specify a resolution that is greater than the resolution of the input file, Elastic Transcoder rescales the output to the higher resolution.
- We recommend that you specify a resolution for which the product of width and height is less than or equal to the applicable value in the following list (*List - Max width x height value*):
- 1 - 25344
- 1b - 25344
- 1.1 - 101376
- 1.2 - 101376
- 1.3 - 101376
- 2 - 101376
- 2.1 - 202752
- 2.2 - 404720
- 3 - 404720
- 3.1 - 921600
- 3.2 - 1310720
- 4 - 2097152
- 4.1 - 2097152
- **AspectRatio** (*string*) –

Warning: To better control resolution and aspect ratio of output videos, we recommend that you use the values `MaxWidth`, `MaxHeight`, `SizingPolicy`, `PaddingPolicy`, and `DisplayAspectRatio` instead of `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The display aspect ratio of the video in the output file. Valid values include:

`auto`, `1:1`, `4:3`, `3:2`, `16:9`

If you specify `auto`, Elastic Transcoder tries to preserve the aspect ratio of the input file.

If you specify an aspect ratio for the output file that differs from aspect ratio of the input file, Elastic Transcoder adds pillarboxing (black bars on the sides) or letterboxing (black bars on the top and bottom) to maintain the aspect ratio of the active region of the video.

- **MaxWidth** (*string*) –

The maximum width of the output video in pixels. If you specify `auto`, Elastic Transcoder uses 1920 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 128 and 4096.

- **MaxHeight** (*string*) –

The maximum height of the output video in pixels. If you specify `auto`, Elastic Transcoder uses 1080 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 96 and 3072.

- **DisplayAspectRatio** (*string*) –

The value that Elastic Transcoder adds to the metadata in the output file.

- **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of the output video:

- **Fit** : Elastic Transcoder scales the output video so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Fill** : Elastic Transcoder scales the output video so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output video and then crops it in the dimension (if any) that exceeds the maximum value.
- **Stretch** : Elastic Transcoder stretches the output video to match the values that you specified for `MaxWidth` and `MaxHeight` . If the relative proportions of the input video and the output video are different, the output video will be distorted.
- **Keep** : Elastic Transcoder does not scale the output video. If either dimension of the input video exceeds the values that you specified for `MaxWidth` and `MaxHeight` , Elastic Transcoder crops the output video.
- **ShrinkToFit** : Elastic Transcoder scales the output video down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the video up.
- **ShrinkToFill** : Elastic Transcoder scales the output video down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the video up.

- **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad` , Elastic Transcoder may add black bars to the top and bottom and/or left and right sides of the output video to make the total size of the output video match the values that you specified for `MaxWidth` and `MaxHeight` .

- **Watermarks** (*list*) –

Settings for the size, location, and opacity of graphics that you want Elastic Transcoder to overlay over videos that are transcoded using this preset. You can specify settings for up to four watermarks. Watermarks appear in the specified size and location, and with the specified opacity for the duration of the transcoded video.

Watermarks can be in `.png` or `.jpg` format. If you want to display a watermark that is not rectangular, use the `.png` format, which supports transparency.

When you create a job that uses this preset, you specify the `.png` or `.jpg` graphics that you want Elastic Transcoder to in-

clude in the transcoded videos. You can specify fewer graphics in the job than you specify watermark settings in the preset, which allows you to use the same preset for up to four watermarks that have different dimensions.

- *(dict)* –

Settings for the size, location, and opacity of graphics that you want Elastic Transcoder to overlay over videos that are transcoded using this preset. You can specify settings for up to four watermarks. Watermarks appear in the specified size and location, and with the specified opacity for the duration of the transcoded video.

Watermarks can be in .png or .jpg format. If you want to display a watermark that is not rectangular, use the .png format, which supports transparency.

When you create a job that uses this preset, you specify the .png or .jpg graphics that you want Elastic Transcoder to include in the transcoded videos. You can specify fewer graphics in the job than you specify watermark settings in the preset, which allows you to use the same preset for up to four watermarks that have different dimensions.

- **Id** (*string*) – A unique identifier for the settings for one watermark. The value of `Id` can be up to 40 characters long.
- **MaxWidth** (*string*) –

The maximum width of the watermark in one of the following formats:

- number of pixels (px): The minimum value is 16 pixels, and the maximum value is the value of `MaxWidth`.
 - integer percentage (%): The range of valid values is 0 to 100. Use the value of `Target` to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the calculation.
- If you specify the value in pixels, it must be less than or equal to the value of `MaxWidth`.

- **MaxHeight** (*string*) –

The maximum height of the watermark in one of the following formats:

- number of pixels (px): The minimum value is 16 pixels, and the maximum value is the value of `MaxHeight`.
- integer percentage (%): The range of valid values is 0 to 100. Use the value of `Target` to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the calculation.

If you specify the value in pixels, it must be less than or equal to the value of `MaxHeight`.

- **SizingPolicy** (*string*) –

A value that controls scaling of the watermark:

- **Fit** : Elastic Transcoder scales the watermark so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Stretch** : Elastic Transcoder stretches the watermark to match the values that you specified for `MaxWidth` and

`MaxHeight` . If the relative proportions of the watermark and the values of `MaxWidth` and `MaxHeight` are different, the watermark will be distorted.

- **ShrinkToFit** : Elastic Transcoder scales the watermark down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the watermark up.

- **HorizontalAlign** (*string*) –

The horizontal position of the watermark unless you specify a non-zero value for `HorizontalOffset` :

- **Left** : The left edge of the watermark is aligned with the left border of the video.
- **Right** : The right edge of the watermark is aligned with the right border of the video.
- **Center** : The watermark is centered between the left and right borders.
- **HorizontalOffset** (*string*) –

The amount by which you want the horizontal position of the watermark to be offset from the position specified by `HorizontalAlign`:

- number of pixels (px): The minimum value is 0 pixels, and the maximum value is the value of `MaxWidth`.
- integer percentage (%): The range of valid values is 0 to 100. For example, if you specify `Left` for `HorizontalAlign` and `5px` for `HorizontalOffset` , the left side of the watermark appears 5 pixels from the left border of the output video.

`HorizontalOffset` is only valid when the value of `HorizontalAlign` is `Left` or `Right` . If you specify an offset that causes the watermark to extend beyond the left or right border and Elastic Transcoder has not added black bars, the watermark is cropped. If Elastic Transcoder has added black bars, the watermark extends into the black bars. If the watermark extends beyond the black bars, it is cropped.

Use the value of `Target` to specify whether you want to include the black bars that are added by Elastic Transcoder, if any, in the offset calculation.

- **VerticalAlign** (*string*) –

The vertical position of the watermark unless you specify a non-zero value for `VerticalOffset` :

- **Top** : The top edge of the watermark is aligned with the top border of the video.
- **Bottom** : The bottom edge of the watermark is aligned with the bottom border of the video.
- **Center** : The watermark is centered between the top and bottom borders.
- **VerticalOffset** (*string*) – `VerticalOffset`

The amount by which you want the vertical position of the watermark to be offset from the position specified by `VerticalAlign`:

- number of pixels (px): The minimum value is 0 pixels, and the maximum value is the value of `MaxHeight`.
- integer percentage (%): The range of valid values is 0 to 100. For example, if you specify `Top` for `VerticalAlign` and `5px` for `VerticalOffset`, the top of the watermark appears 5 pixels from the top border of the output video.

`VerticalOffset` is only valid when the value of `VerticalAlign` is `Top` or `Bottom`.

If you specify an offset that causes the watermark to extend beyond the top or bottom border and Elastic Transcoder has not added black bars, the watermark is cropped. If Elastic Transcoder has added black bars, the watermark extends into the black bars. If the watermark extends beyond the black bars, it is cropped.

Use the value of `Target` to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the offset calculation.

- **Opacity** (*string*) –

A percentage that indicates how much you want a watermark to obscure the video in the location where it appears. Valid values are 0 (the watermark is invisible) to 100 (the watermark completely obscures the video in the specified location). The datatype of `Opacity` is float.

Elastic Transcoder supports transparent .png graphics. If you use a transparent .png, the transparent portion of the video appears as if you had specified a value of 0 for `Opacity`. The .jpg file format doesn't support transparency.

- **Target** (*string*) –

A value that determines how Elastic Transcoder interprets values that you specified for `HorizontalOffset`, `VerticalOffset`, `MaxWidth`, and `MaxHeight`:

- **Content**: `HorizontalOffset` and `VerticalOffset` values are calculated based on the borders of the video excluding black bars added by Elastic Transcoder, if any. In addition, `MaxWidth` and `MaxHeight`, if specified as a percentage, are calculated based on the borders of the video excluding black bars added by Elastic Transcoder, if any.
- **Frame**: `HorizontalOffset` and `VerticalOffset` values are calculated based on the borders of the video including black bars added by Elastic Transcoder, if any. In addition, `MaxWidth` and `MaxHeight`, if specified as a percentage, are calculated based on the borders of the video including black bars added by Elastic Transcoder, if any.

- **Thumbnails** (*dict*) –

A section of the response body that provides information about the thumbnail preset values, if any.

- **Format** (*string*) –

The format of thumbnails, if any. Valid values are `jpg` and `png`.

You specify whether you want Elastic Transcoder to create thumbnails when you create a job.

- **Interval** (*string*) –

The approximate number of seconds between thumbnails. Specify an integer value.

- **Resolution** (*string*) –

Warning: To better control resolution and aspect ratio of thumbnails, we recommend that you use the values `MaxWidth`, `MaxHeight`, `SizingPolicy`, and `PaddingPolicy` instead of `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The width and height of thumbnail files in pixels. Specify a value in the format `*width* x *height*` where both values are even integers. The values cannot exceed the width and height that you specified in the `Video:Resolution` object.

- **AspectRatio** (*string*) –

Warning: To better control resolution and aspect ratio of thumbnails, we recommend that you use the values `MaxWidth`, `MaxHeight`, `SizingPolicy`, and `PaddingPolicy` instead of `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The aspect ratio of thumbnails. Valid values include:

`auto`, `1:1`, `4:3`, `3:2`, `16:9`

If you specify `auto`, Elastic Transcoder tries to preserve the aspect ratio of the video in the output file.

- **MaxWidth** (*string*) –

The maximum width of thumbnails in pixels. If you specify `auto`, Elastic Transcoder uses 1920 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 32 and 4096.

- **MaxHeight** (*string*) –

The maximum height of thumbnails in pixels. If you specify `auto`, Elastic Transcoder uses 1080 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 32 and 3072.

- **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of thumbnails:

- `Fit`: Elastic Transcoder scales thumbnails so they match the value that you specified in thumbnail `MaxWidth` or `MaxHeight` settings without exceeding the other value.
- `Fill`: Elastic Transcoder scales thumbnails so they match the value that you specified in thumbnail `MaxWidth` or

`MaxHeight` settings and matches or exceeds the other value. Elastic Transcoder centers the image in thumbnails and then crops in the dimension (if any) that exceeds the maximum value.

- `Stretch` : Elastic Transcoder stretches thumbnails to match the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings. If the relative proportions of the input video and thumbnails are different, the thumbnails will be distorted.
- `Keep` : Elastic Transcoder does not scale thumbnails. If either dimension of the input video exceeds the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings, Elastic Transcoder crops the thumbnails.
- `ShrinkToFit` : Elastic Transcoder scales thumbnails down so that their dimensions match the values that you specified for at least one of thumbnail `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale thumbnails up.
- `ShrinkToFill` : Elastic Transcoder scales thumbnails down so that their dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale thumbnails up.
- **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder may add black bars to the top and bottom and/or left and right sides of thumbnails to make the total size of the thumbnails match the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings.

- **Type** (*string*) –

Whether the preset is a default preset provided by Elastic Transcoder (`System`) or a preset that you have defined (`Custom`).

- **NextPageToken** (*string*) –

A value that you use to access the second and subsequent pages of results, if any. When the presets fit on one page or when you've reached the last page of results, the value of `NextPageToken` is `null`.

read_job (***kwargs*)

The `ReadJob` operation returns detailed information about a job.

Request Syntax

```
response = client.read_job(
    Id='string'
)
```

Parameters `Id` (*string*) – [REQUIRED]

The identifier of the job for which you want to get detailed information.

Return type dict

Returns

Response Syntax

```
{
  'Job': {
    'Id': 'string',
    'Arn': 'string',
    'PipelineId': 'string',
    'Input': {
      'Key': 'string',
      'FrameRate': 'string',
      'Resolution': 'string',
      'AspectRatio': 'string',
      'Interlaced': 'string',
      'Container': 'string',
      'Encryption': {
        'Mode': 'string',
        'Key': 'string',
        'KeyMd5': 'string',
        'InitializationVector': 'string'
      },
      'DetectedProperties': {
        'Width': 123,
        'Height': 123,
        'FrameRate': 'string',
        'FileSize': 123,
        'DurationMillis': 123
      }
    },
    'Output': {
      'Id': 'string',
      'Key': 'string',
      'ThumbnailPattern': 'string',
      'ThumbnailEncryption': {
        'Mode': 'string',
        'Key': 'string',
        'KeyMd5': 'string',
        'InitializationVector': 'string'
      },
      'Rotate': 'string',
      'PresetId': 'string',
      'SegmentDuration': 'string',
      'Status': 'string',
      'StatusDetail': 'string',
      'Duration': 123,
      'Width': 123,
      'Height': 123,
      'FrameRate': 'string',
      'FileSize': 123,
      'DurationMillis': 123,
      'Watermarks': [
        {
          'PresetWatermarkId': 'string',
          'InputKey': 'string',
          'Encryption': {
            'Mode': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'InitializationVector': 'string'
          }
        }
      ]
    }
  },
}
```

```

],
'AlbumArt': {
    'MergePolicy': 'string',
    'Artwork': [
        {
            'InputKey': 'string',
            'MaxWidth': 'string',
            'MaxHeight': 'string',
            'SizingPolicy': 'string',
            'PaddingPolicy': 'string',
            'AlbumArtFormat': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        }
    ],
},
'Composition': [
    {
        'TimeSpan': {
            'StartTime': 'string',
            'Duration': 'string'
        }
    }
],
'Captions': {
    'MergePolicy': 'string',
    'CaptionSources': [
        {
            'Key': 'string',
            'Language': 'string',
            'TimeOffset': 'string',
            'Label': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        }
    ],
    'CaptionFormats': [
        {
            'Format': 'string',
            'Pattern': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        }
    ]
},
'Encryption': {

```

```
        'Mode': 'string',
        'Key': 'string',
        'KeyMd5': 'string',
        'InitializationVector': 'string'
    },
    'AppliedColorSpaceConversion': 'string'
},
'Outputs': [
    {
        'Id': 'string',
        'Key': 'string',
        'ThumbnailPattern': 'string',
        'ThumbnailEncryption': {
            'Mode': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'InitializationVector': 'string'
        },
        'Rotate': 'string',
        'PresetId': 'string',
        'SegmentDuration': 'string',
        'Status': 'string',
        'StatusDetail': 'string',
        'Duration': 123,
        'Width': 123,
        'Height': 123,
        'FrameRate': 'string',
        'FileSize': 123,
        'DurationMillis': 123,
        'Watermarks': [
            {
                'PresetWatermarkId': 'string',
                'InputKey': 'string',
                'Encryption': {
                    'Mode': 'string',
                    'Key': 'string',
                    'KeyMd5': 'string',
                    'InitializationVector': 'string'
                }
            }
        ],
        'AlbumArt': {
            'MergePolicy': 'string',
            'Artwork': [
                {
                    'InputKey': 'string',
                    'MaxWidth': 'string',
                    'MaxHeight': 'string',
                    'SizingPolicy': 'string',
                    'PaddingPolicy': 'string',
                    'AlbumArtFormat': 'string',
                    'Encryption': {
                        'Mode': 'string',
                        'Key': 'string',
                        'KeyMd5': 'string',
                        'InitializationVector': 'string'
                    }
                }
            ]
        }
    }
],
```



```

    ],
    'Composition': [
        {
            'TimeSpan': {
                'StartTime': 'string',
                'Duration': 'string'
            }
        },
    ],
    'Captions': {
        'MergePolicy': 'string',
        'CaptionSources': [
            {
                'Key': 'string',
                'Language': 'string',
                'TimeOffset': 'string',
                'Label': 'string',
                'Encryption': {
                    'Mode': 'string',
                    'Key': 'string',
                    'KeyMd5': 'string',
                    'InitializationVector': 'string'
                }
            },
        ],
        'CaptionFormats': [
            {
                'Format': 'string',
                'Pattern': 'string',
                'Encryption': {
                    'Mode': 'string',
                    'Key': 'string',
                    'KeyMd5': 'string',
                    'InitializationVector': 'string'
                }
            },
        ],
    },
    'Encryption': {
        'Mode': 'string',
        'Key': 'string',
        'KeyMd5': 'string',
        'InitializationVector': 'string'
    },
    'AppliedColorSpaceConversion': 'string'
},
],
'OutputKeyPrefix': 'string',
'Playlists': [
    {
        'Name': 'string',
        'Format': 'string',
        'OutputKeys': [
            'string',
        ],
        'HlsContentProtection': {
            'Method': 'string',

```

```
        'Key': 'string',
        'KeyMd5': 'string',
        'InitializationVector': 'string',
        'LicenseAcquisitionUrl': 'string',
        'KeyStoragePolicy': 'string'
    },
    'PlayReadyDrm': {
        'Format': 'string',
        'Key': 'string',
        'KeyMd5': 'string',
        'KeyId': 'string',
        'InitializationVector': 'string',
        'LicenseAcquisitionUrl': 'string'
    },
    'Status': 'string',
    'StatusDetail': 'string'
},
],
'Status': 'string',
'UserMetadata': {
    'string': 'string'
},
'Timing': {
    'SubmitTimeMillis': 123,
    'StartTimeMillis': 123,
    'FinishTimeMillis': 123
}
}
```

Response Structure

- (*dict*) –

The ReadJobResponse structure.

- **Job** (*dict*) –

A section of the response body that provides information about the job.

- * **Id** (*string*) –

The identifier that Elastic Transcoder assigned to the job. You use this value to get settings for the job or to delete the job.

- * **Arn** (*string*) –

The Amazon Resource Name (ARN) for the job.

- * **PipelineId** (*string*) –

The `Id` of the pipeline that you want Elastic Transcoder to use for transcoding. The pipeline determines several settings, including the Amazon S3 bucket from which Elastic Transcoder gets the files to transcode and the bucket into which Elastic Transcoder puts the transcoded files.

- * **Input** (*dict*) –

A section of the request or response body that provides information about the file that is being transcoded.

- **Key** (*string*) –

The name of the file to transcode. Elsewhere in the body of the JSON block is the the ID of the pipeline to use for processing the job. The `InputBucket` object in that pipeline

tells Elastic Transcoder which Amazon S3 bucket to get the file from.

If the file name includes a prefix, such as `cooking/lasagna.mpg`, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

- **FrameRate** (*string*) –

The frame rate of the input file. If you want Elastic Transcoder to automatically detect the frame rate of the input file, specify `auto`. If you want to specify the frame rate for the input file, enter one of the following values:

`10, 15, 23.97, 24, 25, 29.97, 30, 60`

If you specify a value other than `auto`, Elastic Transcoder disables automatic detection of the frame rate.

- **Resolution** (*string*) –

This value must be `auto`, which causes Elastic Transcoder to automatically detect the resolution of the input file.

- **AspectRatio** (*string*) –

The aspect ratio of the input file. If you want Elastic Transcoder to automatically detect the aspect ratio of the input file, specify `auto`. If you want to specify the aspect ratio for the output file, enter one of the following values:

`1:1, 4:3, 3:2, 16:9`

If you specify a value other than `auto`, Elastic Transcoder disables automatic detection of the aspect ratio.

- **Interlaced** (*string*) –

Whether the input file is interlaced. If you want Elastic Transcoder to automatically detect whether the input file is interlaced, specify `auto`. If you want to specify whether the input file is interlaced, enter one of the following values:

`true, false`

If you specify a value other than `auto`, Elastic Transcoder disables automatic detection of interlacing.

- **Container** (*string*) –

The container type for the input file. If you want Elastic Transcoder to automatically detect the container type of the input file, specify `auto`. If you want to specify the container type for the input file, enter one of the following values:

`3gp, aac, asf, avi, divx, flv, m4a, mkv, mov, mp3, mp4, mpeg, mpeg-ps, mpeg-ts, mxf, ogg, vob, wav, webm`

- **Encryption** (*dict*) –

The encryption settings, if any, that are used for decrypting your input files. If your input file is encrypted, you must specify the mode that Elastic Transcoder will use to decrypt your file.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or en-

crypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **DetectedProperties** (*dict*) –

The detected properties of the input file.

- **Width** (*integer*) –

The detected width of the input file, in pixels.

- **Height** (*integer*) –

The detected height of the input file, in pixels.

- **FrameRate** (*string*) –

The detected frame rate of the input file, in frames per second.

- **FileSize** (*integer*) –

The detected file size of the input file, in bytes.

- **DurationMillis** (*integer*) –

The detected duration of the input file, in milliseconds.

- * **Output** (*dict*) –

If you specified one output for a job, information about that output. If you specified multiple outputs for a job, the Output object lists information about the first output. This duplicates the information that is listed for the first output in the Outputs object.

Warning: Outputs recommended instead.
--

A section of the request or response body that provides information about the transcoded (target) file.

- **Id** (*string*) –

A sequential counter, starting with 1, that identifies an output among the outputs from the current job. In the Output syntax, this value is always 1.

- **Key** (*string*) –

The name to assign to the transcoded file. Elastic Transcoder saves the file in the Amazon S3 bucket specified by the `OutputBucket` object in the pipeline that is specified by the pipeline ID.

- **ThumbnailPattern** (*string*) –

Whether you want Elastic Transcoder to create thumbnails for your videos and, if so, how you want Elastic Transcoder to name the files.

If you don't want Elastic Transcoder to create thumbnails, specify "".

If you do want Elastic Transcoder to create thumbnails, specify the information that you want to include in the file name for each thumbnail. You can specify the following values in any sequence:

- **“{count}” (Required)** : If you want to create thumbnails, you must include `{count}` in the `ThumbnailPattern` object. Wherever you specify `{count}`, Elastic Transcoder adds a five-digit sequence number (beginning with **00001**) to thumbnail file names. The number indicates where a given thumbnail appears in the sequence of thumbnails for a transcoded file. .. warning::If you specify a literal value and/or `{resolution}` but you omit `{count}`, Elastic

Transcoder returns a validation error and does not create the job.

- **Literal values (Optional)** : You can specify literal values anywhere in the `ThumbnailPattern` object. For example, you can include them as a file name prefix or as a delimiter between `{resolution}` and `{count}`.
- **“{resolution}” (Optional)** : If you want Elastic Transcoder to include the resolution in the file name, include `{resolution}` in the `ThumbnailPattern` object. When creating thumbnails, Elastic Transcoder automatically saves the files in the format (.jpg or .png) that appears in the preset that you specified in the `PresetID` value of `CreateJobOutput`. Elastic Transcoder also appends the applicable file name extension.
- **ThumbnailEncryption (dict)** –

The encryption settings, if any, that you want Elastic Transcoder to apply to your thumbnail.

- **Mode (string)** –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3**: Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS**: Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7**: A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR**: AES Counter Mode.
- **AES-GCM**: AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key (string)** –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-

encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Rotate** (*string*) –

The number of degrees clockwise by which you want Elastic Transcoder to rotate the output relative to the input. Enter one of the following values:

`auto` , `0` , `90` , `180` , `270`

The value `auto` generally works only if the file that you're transcoding contains rotation metadata.

- **PresetId** (*string*) –

The value of the `Id` object for the preset that you want to use for this job. The preset determines the audio, video, and thumbnail settings that Elastic Transcoder uses for transcoding. To use a preset that you created, specify the preset ID that Elastic Transcoder returned in the response when you created the preset. You can also use the Elastic Transcoder system presets, which you can get with `ListPresets` .

- **SegmentDuration** (*string*) –

Warning: (Outputs in Fragmented MP4 or MPEG-TS format only.

If you specify a preset in `PresetId` for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `SegmentDuration` is the target maximum duration of each segment in seconds. For `HLSv3` format playlists, each media segment is stored in a separate `.ts` file. For `HLSv4` and `Smooth` playlists, all media segments for an output are stored in a single file. Each segment is approximately the length of the `SegmentDuration` , though individual segments might be shorter or longer.

The range of valid values is 1 to 60 seconds. If the duration of the video is not evenly divisible by `SegmentDuration`

, the duration of the last segment is the remainder of total length/SegmentDuration.

Elastic Transcoder creates an output-specific playlist for each output HLS output that you specify in `OutputKeys`. To add an output to the master playlist for this job, include it in the `OutputKeys` of the associated playlist.

- **Status** (*string*) –

The status of one output in a job. If you specified only one output for the job, `Outputs:Status` is always the same as `Job:Status`. If you specified more than one output:

- `Job:Status` and `Outputs:Status` for all of the outputs is `Submitted` until Elastic Transcoder starts to process the first output.
- When Elastic Transcoder starts to process the first output, `Outputs:Status` for that output and `Job:Status` both change to `Progressing`. For each output, the value of `Outputs:Status` remains `Submitted` until Elastic Transcoder starts to process the output.
- `Job:Status` remains `Progressing` until all of the outputs reach a terminal status, either `Complete` or `Error`.
- When all of the outputs reach a terminal status, `Job:Status` changes to `Complete` only if `Outputs:Status` for all of the outputs is `Complete`. If `Outputs:Status` for one or more outputs is `Error`, the terminal status for `Job:Status` is also `Error`.
The value of `Status` is one of the following: `Submitted`, `Progressing`, `Complete`, `Canceled`, or `Error`.

- **StatusDetail** (*string*) –

Information that further explains `Status`.

- **Duration** (*integer*) –

Duration of the output file, in seconds.

- **Width** (*integer*) –

Specifies the width of the output file in pixels.

- **Height** (*integer*) –

Height of the output file, in pixels.

- **FrameRate** (*string*) –

Frame rate of the output file, in frames per second.

- **FileSize** (*integer*) –

File size of the output file, in bytes.

- **DurationMillis** (*integer*) –

Duration of the output file, in milliseconds.

- **Watermarks** (*list*) –

Information about the watermarks that you want Elastic Transcoder to add to the video during transcoding. You can specify up to four watermarks for each output. Settings for each watermark must be defined in the preset that you specify in `Preset` for the current output.

Watermarks are added to the output video in the sequence in

which you list them in the job output the first watermark in the list is added to the output video first, the second watermark in the list is added next, and so on. As a result, if the settings in a preset cause Elastic Transcoder to place all watermarks in the same location, the second watermark that you add will cover the first one, the third one will cover the second, and the fourth one will cover the third.

- *(dict)* –

Watermarks can be in .png or .jpg format. If you want to display a watermark that is not rectangular, use the .png format, which supports transparency.

- **PresetWatermarkId** (*string*) –

The ID of the watermark settings that Elastic Transcoder uses to add watermarks to the video during transcoding. The settings are in the preset specified by `Preset` for the current output. In that preset, the value of `Watermarks Id` tells Elastic Transcoder which settings to use.

- **InputKey** (*string*) –

The name of the .png or .jpg file that you want to use for the watermark. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `Pipeline`; the `Input Bucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, **logos/128x64.png**, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your watermarks.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**

- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AlbumArt** (*dict*) –

The album art to be associated with the output file, if any.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder will handle the existence of multiple album artwork files.

- **Replace**: The specified album art will replace any existing album art.
- **Prepend**: The specified album art will be placed in front of any existing album art.
- **Append**: The specified album art will be placed after any existing album art.
- **Fallback**: If the original input file contains artwork, Elastic Transcoder will use that artwork for the output. If the original input does not contain artwork, Elastic Transcoder will use the specified album art file.
- **Artwork** (*list*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20. Valid formats are .jpg and .png

- *(dict)* –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20.

To remove artwork or leave the artwork empty, you can either set `Artwork` to `null`, or set the `Merge Policy` to “Replace” and use an empty `Artwork` array.

To pass through existing artwork unchanged, set the `Merge Policy` to “Prepend”, “Append”, or “Fallback”, and use an empty `Artwork` array.

- **InputKey** (*string*) –

The name of the file to be used as album art. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/pie.jpg`, include the prefix in the key. If the file isn’t in the specified bucket, Elastic Transcoder returns an error.

- **MaxWidth** (*string*) –

The maximum width of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

- **MaxHeight** (*string*) –

The maximum height of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.

- **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of the output album art:

- **Fit**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Fill**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output art and then crops it in the dimension (if any) that exceeds the maximum value.
- **Stretch**: Elastic Transcoder stretches the output art to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input art and the output art are different, the output art will be distorted.
- **Keep**: Elastic Transcoder does not scale the output art. If either dimension of the input art exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output art.
- **ShrinkToFit**: Elastic Transcoder scales the output art down so that its dimensions match the values that you spec-

ified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the art up.

- **ShrinkToFill** Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the art up.

- **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder may add white bars to the top and bottom and/or left and right sides of the output album art to make the total size of the output art match the values that you specified for `MaxWidth` and `MaxHeight`.

- **AlbumArtFormat** (*string*) –

The format of album art, if any. Valid formats are `.jpg` and `.png`.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your artwork.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Composition** (*list*) –

You can create an output file that contains an excerpt from the input file. This excerpt, called a clip, can come from the beginning, middle, or end of the file. The Composition object contains settings for the clips that make up an output file. For the current release, you can only specify settings for a single clip per output file. The Composition object cannot be null.

- (*dict*) –

Settings for one clip in a composition. All jobs in a playlist must have the same clip settings.

- **TimeSpan** (*dict*) –

Settings that determine when a clip begins and how long it lasts.

- **StartTime** (*string*) –

The place in the input file where you want a clip to start. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder starts at the beginning of the input file.

- **Duration** (*string*) –

The duration of the clip. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder creates an output file from StartTime to the end of the file.

If you specify a value longer than the duration of the input file, Elastic Transcoder transcodes the file and returns a warning

message.

- **Captions** (*dict*) –

You can configure Elastic Transcoder to transcode captions, or subtitles, from one format to another. All captions must be in UTF-8. Elastic Transcoder supports two types of captions:

- **Embedded:** Embedded captions are included in the same file as the audio and video. Elastic Transcoder supports only one embedded caption per language, to a maximum of 300 embedded captions per file. Valid input values include: `CEA-608` (EIA-608 , first non-empty channel only), `CEA-708` (EIA-708 , first non-empty channel only), and `mov-text`. Valid outputs include: `mov-text`. Elastic Transcoder supports a maximum of one embedded format per output.
- **Sidecar:** Sidecar captions are kept in a separate metadata file from the audio and video data. Sidecar captions require a player that is capable of understanding the relationship between the video file and the sidecar file. Elastic Transcoder supports only one sidecar caption per language, to a maximum of 20 sidecar captions per file. Valid input values include: `dfxp` (first div element only), `ebu-tt`, `scc`, `smpt`, `srt`, `ttml` (first div element only), and `webvtt`. Valid outputs include: `dfxp` (first div element only), `scc`, `srt`, and `webvtt`.

If you want `ttml` or `smpte-tt` compatible captions, specify `dfxp` as your output format.

Elastic Transcoder does not support OCR (Optical Character Recognition), does not accept pictures as a valid input for captions, and is not available for audio-only transcoding. Elastic Transcoder does not preserve text formatting (for example, italics) during the transcoding process.

To remove captions or leave the captions empty, set `Captions` to `null`. To pass through existing captions unchanged, set the `MergePolicy` to `MergeRetain`, and pass in a null `CaptionSources` array.

For more information on embedded files, see the Subtitles Wikipedia page.

For more information on sidecar files, see the Extensible Metadata Platform and Sidecar file Wikipedia pages.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder handles the existence of multiple captions.

- **MergeOverride:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the sidecar captions and ignores the embedded captions for that language.
- **MergeRetain:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the embedded captions and ignores

the sidecar captions for that language. If `CaptionSources` is empty, Elastic Transcoder omits all sidecar captions from the output files.

- **Override:** Elastic Transcoder transcodes only the sidecar captions that you specify in `CaptionSources`. `MergePolicy` cannot be null.

- **CaptionSources** (*list*) –

Source files for the input sidecar captions used during the transcoding process. To omit all sidecar captions, leave `CaptionSources` blank.

- (*dict*) –

A source file for the input sidecar captions used during the transcoding process.

- **Key** (*string*) –

The name of the sidecar caption file that you want Elastic Transcoder to include in the output file.

- **Language** (*string*) –

A string that specifies the language of the caption. Specify this as one of:

- 2-character ISO 639-1 code
- 3-character ISO 639-2 code

For more information on ISO language codes and language names, see the List of ISO 639-1 codes.

- **TimeOffset** (*string*) –

For clip generation or captions that do not start at the same time as the associated video file, the `TimeOffset` tells Elastic Transcoder how much of the video to encode before including captions.

Specify the `TimeOffset` in the form `[+/-]SS.sss` or `[+/-]HH:mm:SS.ss`.

- **Label** (*string*) –

The label of the caption shown in the player when choosing a language. We recommend that you put the caption language name here, in the language of the captions.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption sources.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.

- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **CaptionFormats** (*list*) –

The array of file formats for the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- (*dict*) –

The file format of the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- **Format** (*string*) –

The format you specify determines whether Elastic Transcoder generates an embedded or sidecar caption

for this output.

- **Valid Embedded Caption Formats:** * for **FLAC** : None
- **For MP3** : None
- **For MP4** : mov-text
- **For MPEG-TS** : None
- **For ogg** : None
- **For webm** : None
- **Valid Sidecar Caption Formats:** Elastic Transcoder supports dfxp (first div element only), scc, srt, and webvtt. If you want ttml or smpte-tt compatible captions, specify dfxp as your output format. * **For FMP4** : dfxp
- **Non-FMP4 outputs** : All sidecar types
fmp4 captions have an extension of .ismt
- **Pattern** (*string*) –

The prefix for caption filenames, in the form *description* - {language} , where:

- *description* is a description of the video.
- {language} is a literal value that Elastic Transcoder replaces with the two- or three-letter code for the language of the caption in the output file names.

If you don't include {language} in the file name pattern, Elastic Transcoder automatically appends "{language}" to the value that you specify for the description. In addition, Elastic Transcoder automatically appends the count to the end of the segment files.

For example, suppose you're transcoding into srt format. When you enter "Sydney-{language}-sunrise", and the language of the captions is English (en), the name of the first caption file will be Sydney-en-sunrise00000.srt.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption formats.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify S3-AWS-KMS and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following

settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your output files. If you choose to use encryption, you must specify a mode to use. If you choose not to use encryption, Elastic Transcoder will write an unencrypted file to your Amazon S3 bucket.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the

AWS-KMS key that you want to use to your pipeline.

- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AppliedColorSpaceConversion** (*string*) –

If Elastic Transcoder used a preset with a `ColorSpaceConversionMode` to transcode the output file, the `AppliedColorSpaceConversion` parameter shows the conversion used. If no `ColorSpaceConversionMode` was defined in the preset, this parameter will not be included in the job response.

- * **Outputs** (*list*) –

Information about the output files. We recommend that you use the `Outputs` syntax for all jobs, even when you want Elastic Transcoder to transcode a file into only one format. Do not use both the `Outputs` and `Output` syntaxes in the same request. You can create a maximum of 30 outputs per job.

If you specify more than one output for a job, Elastic Transcoder creates the files for each output in the order in which you specify them in the job.

- *(dict)* –

Warning: Outputs recommended instead.

If you specified one output for a job, information about that output. If you specified multiple outputs for a job, the `Output` object lists information about the first output. This duplicates the information that is listed for the first output in the `Outputs` object.

- **Id** (*string*) –

A sequential counter, starting with 1, that identifies an output among the outputs from the current job. In the `Output` syntax, this value is always 1.

- **Key** (*string*) –

The name to assign to the transcoded file. Elastic Transcoder saves the file in the Amazon S3 bucket specified by the `OutputBucket` object in the pipeline that is specified by the pipeline ID.

- **ThumbnailPattern** (*string*) –

Whether you want Elastic Transcoder to create thumbnails for your videos and, if so, how you want Elastic Transcoder to name the files.

If you don't want Elastic Transcoder to create thumbnails, specify `""`.

If you do want Elastic Transcoder to create thumbnails, specify the information that you want to include in the file name for each thumbnail. You can specify the following values in any sequence:

- **“{count}” (Required)** : If you want to create thumbnails, you must include `{count}` in the `ThumbnailPattern` object. Wherever you specify `{count}`, Elastic Transcoder adds a five-digit sequence number (beginning with **00001**) to thumbnail file names. The number indicates where a given thumbnail appears in the sequence of thumbnails for a transcoded file. .. warning::If you specify a literal value and/or `{resolution}` but you omit `{count}`, Elastic Transcoder returns a validation error and does not create the job.
- **Literal values (Optional)** : You can specify literal values anywhere in the `ThumbnailPattern` object. For example, you can include them as a file name prefix or as a delimiter between `{resolution}` and `{count}`.

- “{resolution}” (Optional) : If you want Elastic Transcoder to include the resolution in the file name, include {resolution} in the ThumbnailPattern object.

When creating thumbnails, Elastic Transcoder automatically saves the files in the format (.jpg or .png) that appears in the preset that you specified in the PresetID value of CreateJobOutput . Elastic Transcoder also appends the applicable file name extension.

- **ThumbnailEncryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your thumbnail.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3**: Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS**: Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify S3-AWS-KMS and you don’t want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7**: A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR**: AES Counter Mode.
- **AES-GCM**: AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won’t be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Rotate** (*string*) –

The number of degrees clockwise by which you want Elastic Transcoder to rotate the output relative to the input. Enter one of the following values:

`auto, 0, 90, 180, 270`

The value `auto` generally works only if the file that you're transcoding contains rotation metadata.

- **PresetId** (*string*) –

The value of the `Id` object for the preset that you want to use for this job. The preset determines the audio, video, and thumbnail settings that Elastic Transcoder uses for transcoding. To use a preset that you created, specify the preset ID that Elastic Transcoder returned in the response when you created the preset. You can also use the Elastic Transcoder system presets, which you can get with `ListPresets`.

- **SegmentDuration** (*string*) –

Warning: (Outputs in Fragmented MP4 or MPEG-TS format only.

If you specify a preset in `PresetId` for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `SegmentDuration` is the target maximum duration of each segment in seconds. For `HLSv3` format playlists, each media segment is stored in a separate `.ts` file. For `HLSv4` and `Smooth` playlists, all media segments for an output are stored in a single file. Each segment is approximately the length of the `SegmentDuration`, though individual segments might be shorter or longer.

The range of valid values is 1 to 60 seconds. If the duration of the video is not evenly divisible by `SegmentDuration`, the duration of the last segment is the remainder of total length/`SegmentDuration`.

Elastic Transcoder creates an output-specific playlist for each output `HLS` output that you specify in `OutputKeys`. To add an output to the master playlist for this job, include it in the `OutputKeys` of the associated playlist.

- **Status** (*string*) –

The status of one output in a job. If you specified only one output for the job, `Outputs:Status` is always the same as `Job:Status`. If you specified more than one output:

- `Job:Status` and `Outputs:Status` for all of the outputs is `Submitted` until Elastic Transcoder starts to process the first output.
 - When Elastic Transcoder starts to process the first output, `Outputs:Status` for that output and `Job:Status` both change to `Progressing`. For each output, the value of `Outputs:Status` remains `Submitted` until Elastic Transcoder starts to process the output.
 - `Job:Status` remains `Progressing` until all of the outputs reach a terminal status, either `Complete` or `Error`.
 - When all of the outputs reach a terminal status, `Job:Status` changes to `Complete` only if `Outputs:Status` for all of the outputs is `Complete`. If `Outputs:Status` for one or more outputs is `Error`, the terminal status for `Job:Status` is also `Error`.
- The value of `Status` is one of the following: `Submitted`, `Progressing`, `Complete`, `Canceled`, or `Error`.

- **StatusDetail** (*string*) –

Information that further explains `Status`.

- **Duration** (*integer*) –

Duration of the output file, in seconds.

- **Width** (*integer*) –

Specifies the width of the output file in pixels.

- **Height** (*integer*) –

Height of the output file, in pixels.

- **FrameRate** (*string*) –

Frame rate of the output file, in frames per second.

- **FileSize** (*integer*) –

File size of the output file, in bytes.

- **DurationMillis** (*integer*) –

Duration of the output file, in milliseconds.

- **Watermarks** (*list*) –

Information about the watermarks that you want Elastic Transcoder to add to the video during transcoding. You can specify up to four watermarks for each output. Settings for each watermark must be defined in the preset that you specify in `Preset` for the current output.

Watermarks are added to the output video in the sequence in which you list them in the job output. The first watermark in the list is added to the output video first, the second watermark in the list is added next, and so on. As a result, if the settings in a preset cause Elastic Transcoder to place all watermarks in the same location, the second watermark that you add will cover the first one, the third one will cover the second, and the fourth one will cover the third.

- *(dict)* –

Watermarks can be in .png or .jpg format. If you want to display a watermark that is not rectangular, use the .png format, which supports transparency.

- **PresetWatermarkId** (*string*) –

The ID of the watermark settings that Elastic Transcoder uses to add watermarks to the video during transcoding. The settings are in the preset specified by `Preset` for the current output. In that preset, the value of `Watermarks Id` tells Elastic Transcoder which settings to use.

- **InputKey** (*string*) –

The name of the .png or .jpg file that you want to use for the watermark. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `Pipeline`; the `Input Bucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, **logos/128x64.png**, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your watermarks.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

· **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

· **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

· **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

· **AlbumArt** (*dict*) –

The album art to be associated with the output file, if any.

· **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder will handle the existence of multiple album artwork files.

- **Replace**: The specified album art will replace any existing album art.
- **Prepend**: The specified album art will be placed in front of any existing album art.
- **Append**: The specified album art will be placed after any existing album art.
- **Fallback**: If the original input file contains artwork, Elastic Transcoder will use that artwork for the output. If the original input does not contain artwork, Elastic Transcoder will use the specified album art file.

· **Artwork** (*list*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20. Valid formats are .jpg and .png

· (*dict*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20.

To remove artwork or leave the artwork empty, you can either set `Artwork` to null, or set the `Merge Policy` to “Replace” and use an empty `Artwork` array.

To pass through existing artwork unchanged, set the `Merge Policy` to “Prepend”, “Append”, or “Fallback”, and use an empty `Artwork` array.

- **InputKey** (*string*) –

The name of the file to be used as album art. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/pie.jpg`, include the prefix in the key. If the file isn’t in the specified bucket, Elastic Transcoder returns an error.

- **MaxWidth** (*string*) –

The maximum width of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

- **MaxHeight** (*string*) –

The maximum height of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.

- **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of the output album art:

- **Fit**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Fill**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output art and then crops it in the dimension (if any) that exceeds the maximum value.
- **Stretch**: Elastic Transcoder stretches the output art to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input art and the output art are different, the output art will be distorted.
- **Keep**: Elastic Transcoder does not scale the output art. If either dimension of the input art exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output art.
- **ShrinkToFit**: Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` with-

out exceeding either value. If you specify this option, Elastic Transcoder does not scale the art up.

- **ShrinkToFill** Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the art up.
- **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder may add white bars to the top and bottom and/or left and right sides of the output album art to make the total size of the output art match the values that you specified for `MaxWidth` and `MaxHeight`.

- **AlbumArtFormat** (*string*) –

The format of album art, if any. Valid formats are `.jpg` and `.png`.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your artwork.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder

to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Composition** (*list*) –

You can create an output file that contains an excerpt from the input file. This excerpt, called a clip, can come from the beginning, middle, or end of the file. The Composition object contains settings for the clips that make up an output file. For the current release, you can only specify settings for a single clip per output file. The Composition object cannot be null.

- (*dict*) –

Settings for one clip in a composition. All jobs in a playlist must have the same clip settings.

- **TimeSpan** (*dict*) –

Settings that determine when a clip begins and how long it lasts.

- **StartTime** (*string*) –

The place in the input file where you want a clip to start. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder starts at the beginning of the input file.

- **Duration** (*string*) –

The duration of the clip. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder creates an output file from StartTime to the end of the file.

If you specify a value longer than the duration of the input file, Elastic Transcoder transcodes the file and returns a warning message.

- **Captions** (*dict*) –

You can configure Elastic Transcoder to transcode captions, or subtitles, from one format to another. All captions must be in UTF-8. Elastic Transcoder supports two types of captions:

- **Embedded:** Embedded captions are included in the same file as the audio and video. Elastic Transcoder supports only one embedded caption per language, to a maximum of 300 embedded captions per file. Valid input values include: `CEA-608` (EIA-608 , first non-empty channel only), `CEA-708` (EIA-708 , first non-empty channel only), and `mov-text`. Valid outputs include: `mov-text`. Elastic Transcoder supports a maximum of one embedded format per output.
- **Sidecar:** Sidecar captions are kept in a separate metadata file from the audio and video data. Sidecar captions require a player that is capable of understanding the relationship between the video file and the sidecar file. Elastic Transcoder supports only one sidecar caption per language, to a maximum of 20 sidecar captions per file. Valid input values include: `dfxp` (first div element only), `ebu-tt` , `scc` , `smpt` , `srt` , `ttml` (first div element only), and `webvtt`. Valid outputs include: `dfxp` (first div element only), `scc` , `srt` , and `webvtt`.

If you want `ttml` or `smpte-tt` compatible captions, specify `dfxp` as your output format.

Elastic Transcoder does not support OCR (Optical Character Recognition), does not accept pictures as a valid input for captions, and is not available for audio-only transcoding. Elastic Transcoder does not preserve text formatting (for example, italics) during the transcoding process.

To remove captions or leave the captions empty, set `Captions` to `null`. To pass through existing captions unchanged, set the `MergePolicy` to `MergeRetain` , and pass in a null `CaptionSources` array.

For more information on embedded files, see the Subtitles Wikipedia page.

For more information on sidecar files, see the Extensible Metadata Platform and Sidecar file Wikipedia pages.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder handles the existence of multiple captions.

- **MergeOverride:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the sidecar captions and ignores the embedded captions for that language.
- **MergeRetain:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the embedded captions and ignores the sidecar captions for that language. If `CaptionSources`

is empty, Elastic Transcoder omits all sidecar captions from the output files.

- **Override:** Elastic Transcoder transcodes only the sidecar captions that you specify in `CaptionSources`. `MergePolicy` cannot be null.

- **CaptionSources** (*list*) –

Source files for the input sidecar captions used during the transcoding process. To omit all sidecar captions, leave `CaptionSources` blank.

- (*dict*) –

A source file for the input sidecar captions used during the transcoding process.

- **Key** (*string*) –

The name of the sidecar caption file that you want Elastic Transcoder to include in the output file.

- **Language** (*string*) –

A string that specifies the language of the caption. Specify this as one of:

- 2-character ISO 639-1 code
- 3-character ISO 639-2 code

For more information on ISO language codes and language names, see the List of ISO 639-1 codes.

- **TimeOffset** (*string*) –

For clip generation or captions that do not start at the same time as the associated video file, the `TimeOffset` tells Elastic Transcoder how much of the video to encode before including captions.

Specify the `TimeOffset` in the form `[+-]SS.sss` or `[+-]HH:mm:SS.ss`.

- **Label** (*string*) –

The label of the caption shown in the player when choosing a language. We recommend that you put the caption language name here, in the language of the captions.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption sources.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.

- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **CaptionFormats** (*list*) –

The array of file formats for the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- (*dict*) –

The file format of the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- **Format** (*string*) –

The format you specify determines whether Elastic Transcoder generates an embedded or sidcar caption

for this output.

- **Valid Embedded Caption Formats:** * **for FLAC** : None
- **For MP3** : None
- **For MP4** : mov-text
- **For MPEG-TS** : None
- **For ogg** : None
- **For webm** : None
- **Valid Sidecar Caption Formats:** Elastic Transcoder supports dfxp (first div element only), scc, srt, and webvtt. If you want ttml or smpte-tt compatible captions, specify dfxp as your output format. * **For FMP4** : dfxp
- **Non-FMP4 outputs** : All sidecar types
fmp4 captions have an extension of .ismt
- **Pattern** (*string*) –

The prefix for caption filenames, in the form *description* - {language}, where:

- *description* is a description of the video.
- {language} is a literal value that Elastic Transcoder replaces with the two- or three-letter code for the language of the caption in the output file names.

If you don't include {language} in the file name pattern, Elastic Transcoder automatically appends "{language}" to the value that you specify for the description. In addition, Elastic Transcoder automatically appends the count to the end of the segment files.

For example, suppose you're transcoding into srt format. When you enter "Sydney-{language}-sunrise", and the language of the captions is English (en), the name of the first caption file will be Sydney-en-sunrise00000.srt.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption formats.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify S3-AWS-KMS and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following

settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your output files. If you choose to use encryption, you must specify a mode to use. If you choose not to use encryption, Elastic Transcoder will write an unencrypted file to your Amazon S3 bucket.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the

AWS-KMS key that you want to use to your pipeline.

- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AppliedColorSpaceConversion** (*string*) –

If Elastic Transcoder used a preset with a `ColorSpaceConversionMode` to transcode the output file, the `AppliedColorSpaceConversion` parameter shows the conversion used. If no `ColorSpaceConversionMode` was defined in the preset, this parameter will not be included in the job response.

- * **OutputKeyPrefix** (*string*) –

The value, if any, that you want Elastic Transcoder to prepend to the names of all files that this job creates, including output files, thumbnails, and playlists. We recommend that you add a / or some other delimiter to the end of the `OutputKeyPrefix`.

* **Playlists** (*list*) –

Warning: Outputs in Fragmented MP4 or MPEG-TS format only.

If you specify a preset in `PresetId` for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `Playlists` contains information about the master playlists that you want Elastic Transcoder to create.

The maximum number of master playlists in a job is 30.

· (*dict*) –

Use Only for Fragmented MP4 or MPEG-TS Outputs. If you specify a preset for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `Playlists` contains information about the master playlists that you want Elastic Transcoder to create. We recommend that you create only one master playlist per output format. The maximum number of master playlists in a job is 30.

· **Name** (*string*) –

The name that you want Elastic Transcoder to assign to the master playlist, for example, `nyc-vacation.m3u8`. If the name includes a / character, the section of the name before the last / must be identical for all `Name` objects. If you create more than one master playlist, the values of all `Name` objects must be unique.

Note : Elastic Transcoder automatically appends the relevant file extension to the file name (`.m3u8` for `HLSv3` and `HLSv4` playlists, and `.ism` and `.ismc` for `Smooth` playlists). If you include a file extension in `Name`, the file name will have two extensions.

· **Format** (*string*) –

The format of the output playlist. Valid formats include `HLSv3`, `HLSv4`, and `Smooth`.

· **OutputKeys** (*list*) –

For each output in this job that you want to include in a master playlist, the value of the `Outputs:Key` object.

- If your output is not `HLS` or does not have a segment duration set, the name of the output file is a concatenation of `OutputKeyPrefix` and `Outputs:Key:OutputKeyPrefix“Outputs:Key“`
- If your output is `HLSv3` and has a segment duration set, or is not included in a playlist, Elastic Transcoder creates an output playlist file with a file extension of `.m3u8`, and a series of `.ts` files that include a five-digit sequential counter beginning with 00000: `OutputKeyPrefix“Outputs:Key“ .m3u8`
`OutputKeyPrefix“Outputs:Key“ 00000.ts`

- If your output is `HLSv4` , has a segment duration set, and is included in an `HLSv4` playlist, Elastic Transcoder creates an output playlist file with a file extension of `_v4.m3u8` . If the output is video, Elastic Transcoder also creates an output file with an extension of `_iframe.m3u8` : `OutputKeyPrefix“Outputs:Key“_v4.m3u8` `OutputKeyPrefix“Outputs:Key“_iframe.m3u8` `OutputKeyPrefix“Outputs:Key“ .ts`

Elastic Transcoder automatically appends the relevant file extension to the file name. If you include a file extension in Output Key, the file name will have two extensions.

If you include more than one output in a playlist, any segment duration settings, clip settings, or caption settings must be the same for all outputs in the playlist. For Smooth playlists, the `Audio:Profile` , `Video:Profile` , and `Video:FrameRate` to `Video:KeyframesMaxDist` ratio must be the same for all outputs.

- *(string)* –
- **HlsContentProtection** (*dict*) –

The HLS content protection settings, if any, that you want Elastic Transcoder to apply to the output files associated with this playlist.

- **Method** (*string*) –

The content protection method for your output. The only valid value is: `aes-128` .

This value will be written into the method attribute of the `EXT-X-KEY` metadata tag in the output playlist.

- **Key** (*string*) –

If you want Elastic Transcoder to generate a key for you, leave this field blank.

If you choose to supply your own key, you must encrypt the key by using AWS KMS. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

`128` , `192` , or `256` .

- **KeyMd5** (*string*) –

If Elastic Transcoder is generating your key for you, you must leave this field blank.

The MD5 digest of the key that you want Elastic Transcoder to use to encrypt your output file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

- **InitializationVector** (*string*) –

If Elastic Transcoder is generating your key for you, you must leave this field blank.

The series of random bits created by a random bit generator,

unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

- **LicenseAcquisitionUrl** (*string*) –

The location of the license key required to decrypt your HLS playlist. The URL must be an absolute path, and is referenced in the URI attribute of the EXT-X-KEY metadata tag in the playlist file.

- **KeyStoragePolicy** (*string*) –

Specify whether you want Elastic Transcoder to write your HLS license key to an Amazon S3 bucket. If you choose `WithVariantPlaylists`, `LicenseAcquisitionUrl` must be left blank and Elastic Transcoder writes your data key into the same bucket as the associated playlist.

- **PlayReadyDrm** (*dict*) –

The DRM settings, if any, that you want Elastic Transcoder to apply to the output files associated with this playlist.

- **Format** (*string*) –

The type of DRM, if any, that you want Elastic Transcoder to apply to the output files associated with this playlist.

- **Key** (*string*) –

The DRM key for your file, provided by your DRM license provider. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using AWS KMS.

- **KeyMd5** (*string*) –

The MD5 digest of the key used for DRM on your file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

- **KeyId** (*string*) –

The ID for your DRM key, so that your DRM license provider knows which key to provide.

The key ID must be provided in big endian, and Elastic Transcoder will convert it to little endian before inserting it into the PlayReady DRM headers. If you are unsure whether your license server provides your key ID in big or little endian, check with your DRM provider.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your files. The initialization vector must be base64-encoded, and it must be exactly 8 bytes

long before being base64-encoded. If no initialization vector is provided, Elastic Transcoder generates one for you.

- **LicenseAcquisitionUrl** (*string*) –

The location of the license key required to play DRM content. The URL must be an absolute path, and is referenced by the PlayReady header. The PlayReady header is referenced in the protection header of the client manifest for Smooth Streaming outputs, and in the EXT-X-DXDRM and EXT-XDXDRMINFO metadata tags for HLS playlist outputs. An example URL looks like this: <https://www.example.com/exampleKey/>

- **Status** (*string*) –

The status of the job with which the playlist is associated.

- **StatusDetail** (*string*) –

Information that further explains the status.

- * **Status** (*string*) –

The status of the job: Submitted, Progressing, Complete, Canceled, or Error.

- * **UserMetadata** (*dict*) –

User-defined metadata that you want to associate with an Elastic Transcoder job. You specify metadata in *key/value* pairs, and you can add up to 10 *key/value* pairs per job. Elastic Transcoder does not guarantee that *key/value* pairs will be returned in the same order in which you specify them.

Metadata *keys* and *values* must use characters from the following list:

- 0–9
- A–Z and a–z
- Space
- The following symbols: `_ . : / = + - % @`
- (*string*) –
- (*string*) –

- * **Timing** (*dict*) –

Details about the timing of a job.

- **SubmitTimeMillis** (*integer*) –

The time the job was submitted to Elastic Transcoder, in epoch milliseconds.

- **StartTimeMillis** (*integer*) –

The time the job began transcoding, in epoch milliseconds.

- **FinishTimeMillis** (*integer*) –

The time the job finished transcoding, in epoch milliseconds.

read_pipeline (***kwargs*)

The ReadPipeline operation gets detailed information about a pipeline.

Request Syntax

```
response = client.read_pipeline(
    Id='string'
)
```

Parameters *Id* (*string*) – **[REQUIRED]**

The identifier of the pipeline to read.

Return type dict

Returns

Response Syntax

```
{
  'Pipeline': {
    'Id': 'string',
    'Arn': 'string',
    'Name': 'string',
    'Status': 'string',
    'InputBucket': 'string',
    'OutputBucket': 'string',
    'Role': 'string',
    'AwsKmsKeyArn': 'string',
    'Notifications': {
      'Progressing': 'string',
      'Completed': 'string',
      'Warning': 'string',
      'Error': 'string'
    },
    'ContentConfig': {
      'Bucket': 'string',
      'StorageClass': 'string',
      'Permissions': [
        {
          'GranteeType': 'string',
          'Grantee': 'string',
          'Access': [
            'string',
          ]
        }
      ],
    },
    'ThumbnailConfig': {
      'Bucket': 'string',
      'StorageClass': 'string',
      'Permissions': [
        {
          'GranteeType': 'string',
          'Grantee': 'string',
          'Access': [
            'string',
          ]
        }
      ],
    },
  ],
  'Warnings': [
    {
      'Code': 'string',
```

```
        'Message': 'string'
    },
]
}
```

Response Structure

- (*dict*) –

The `ReadPipelineResponse` structure.

- **Pipeline** (*dict*) –

A section of the response body that provides information about the pipeline.

- * **Id** (*string*) –

The identifier for the pipeline. You use this value to identify the pipeline in which you want to perform a variety of operations, such as creating a job or a preset.

- * **Arn** (*string*) –

The Amazon Resource Name (ARN) for the pipeline.

- * **Name** (*string*) –

The name of the pipeline. We recommend that the name be unique within the AWS account, but uniqueness is not enforced.

Constraints: Maximum 40 characters

- * **Status** (*string*) –

The current status of the pipeline:

- `Active`: The pipeline is processing jobs.
- `Paused`: The pipeline is not currently processing jobs.

- * **InputBucket** (*string*) –

The Amazon S3 bucket from which Elastic Transcoder gets media files for transcoding and the graphics files, if any, that you want to use for watermarks.

- * **OutputBucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files, thumbnails, and playlists. Either you specify this value, or you specify both `ContentConfig` and `ThumbnailConfig`.

- * **Role** (*string*) –

The IAM Amazon Resource Name (ARN) for the role that Elastic Transcoder uses to transcode jobs for this pipeline.

- * **AwsKmsKeyArn** (*string*) –

The AWS Key Management Service (AWS KMS) key that you want to use with this pipeline.

If you use either `S3` or `S3-AWS-KMS` as your `Encryption:Mode`, you don't need to provide a key with your job because a default key, known as an AWS-KMS key, is created for you automatically. You need to provide an AWS-KMS key only if you want to use a non-default AWS-KMS key, or if you are using an `Encryption:Mode` of `AES-PKCS7`, `AES-CTR`, or `AES-GCM`.

* **Notifications** (*dict*) –

The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify to report job status.

Warning: To receive notifications, you must also subscribe to the new topic in the Amazon SNS console.

- **Progressing** (optional): The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process the job.
- **Completed** (optional): The Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing the job.
- **Warning** (optional): The Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition.
- **Error** (optional): The Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition.

· **Progressing** (*string*) –

The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process the job.

· **Completed** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing the job.

· **Warning** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition.

· **Error** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition.

* **ContentConfig** (*dict*) –

Information about the Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists. Either you specify both `ContentConfig` and `ThumbnailConfig`, or you specify `OutputBucket`.

- **Bucket** : The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists.
- **Permissions** : A list of the users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access that you want them to have. * **GranteeType**: The type of value that appears in the `Grantee` object: * **Canonical** : Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution.
- **Email** : The registered email address of an AWS account.
- **Group** : One of the following predefined Amazon S3 groups: `AllUsers` , `AuthenticatedUsers` , or `LogDelivery` .
- **Grantee** : The AWS user or group that you want to have access to transcoded files and playlists.

- **Access** : The permission that you want to give to the AWS user that is listed in `Grantee` . Valid values include: * `READ` : The grantee can read the objects and metadata for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- `READ_ACP` : The grantee can read the object ACL for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- `WRITE_ACP` : The grantee can write the ACL for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
- `FULL_CONTROL` : The grantee has `READ` , `READ_ACP` , and `WRITE_ACP` permissions for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **StorageClass** : The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

- **Bucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. Specify this value when all of the following are true:

- You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.
- You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- You do not want to specify the permissions that Elastic Transcoder grants to the files.
- You want to associate the transcoded files and thumbnails with the Amazon S3 `Standard` storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit `OutputBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

- **StorageClass** (*string*) –

The Amazon S3 storage class, `Standard` or `ReducedRedundancy` , that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

- **Permissions** (*list*) –

Optional. The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions` , Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role` . If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions` , Elastic Transcoder grants full

control over the transcoded files and playlists to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

- *(dict)* –

The `Permission` structure.

- **GranteeType** (*string*) –

The type of value that appears in the `Grantee` object:

- `Canonical` : Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.
- `Email` : The registered email address of an AWS account.
- `Group` : One of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.
- **Grantee** (*string*) –

The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group.

- **Access** (*list*) –

The permission that you want to give to the AWS user that is listed in `Grantee`. Valid values include:

- `READ` : The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `READ_ACP` : The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `WRITE_ACP` : The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `FULL_CONTROL` : The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- (*string*) –

- * **ThumbnailConfig** (*dict*) –

Information about the Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files. Either you specify both `ContentConfig` and `ThumbnailConfig`, or you specify `OutputBucket`.

- `Bucket` : The Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files.
- `Permissions` : A list of the users and/or predefined Amazon S3 groups you want to have access to thumbnail files, and the type of access that you want them to have. * **GranteeType**: The type of value that appears in the `Grantee` object: * `Canonical` : Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.

- **Email** : The registered email address of an AWS account.
- **Group** : One of the following predefined Amazon S3 groups: `AllUsers` , `AuthenticatedUsers` , or `LogDelivery` .
- **Grantee** : The AWS user or group that you want to have access to thumbnail files.
- **Access**: The permission that you want to give to the AWS user that is listed in **Grantee**. Valid values include: * `READ` : The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `READ_ACP` : The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `WRITE_ACP` : The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `FULL_CONTROL` : The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **StorageClass** : The Amazon S3 storage class, `Standard` or `ReducedRedundancy` , that you want Elastic Transcoder to assign to the thumbnails that it stores in your Amazon S3 bucket.

- **Bucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. Specify this value when all of the following are true:

- You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.
- You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- You do not want to specify the permissions that Elastic Transcoder grants to the files.
- You want to associate the transcoded files and thumbnails with the Amazon S3 Standard storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit `OutputBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

- **StorageClass** (*string*) –

The Amazon S3 storage class, `Standard` or `ReducedRedundancy` , that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

- **Permissions** (*list*) –

Optional. The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions` , Elastic Transcoder grants

only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role`. If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions`, Elastic Transcoder grants full control over the transcoded files and playlists to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

- *(dict)* –

The `Permission` structure.

- **GranteeType** (*string*) –

The type of value that appears in the Grantee object:

- `Canonical`: Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.
- `Email`: The registered email address of an AWS account.
- `Group`: One of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.
- **Grantee** (*string*) –

The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group.

- **Access** (*list*) –

The permission that you want to give to the AWS user that is listed in Grantee. Valid values include:

- `READ`: The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `READ_ACP`: The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `WRITE_ACP`: The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `FULL_CONTROL`: The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- (*string*) –

- **Warnings** (*list*) –

Elastic Transcoder returns a warning if the resources used by your pipeline are not in the same region as the pipeline.

Using resources in the same region, such as your Amazon S3 buckets, Amazon SNS notification topics, and AWS KMS key, reduces processing time and prevents cross-regional charges.

- *(dict)* –

Elastic Transcoder returns a warning if the resources used by your pipeline are not in the same region as the pipeline.

Using resources in the same region, such as your Amazon S3 buckets, Amazon SNS notification topics, and AWS KMS key, reduces processing time and prevents cross-regional charges.

- **Code** (*string*) –

The code of the cross-regional warning.

- **Message** (*string*) –

The message explaining what resources are in a different region from the pipeline.

Note: AWS KMS keys must be in the same region as the pipeline.

read_preset (***kwargs*)

The ReadPreset operation gets detailed information about a preset.

Request Syntax

```
response = client.read_preset(
    Id='string'
)
```

Parameters **Id** (*string*) – [REQUIRED]

The identifier of the preset for which you want to get detailed information.

Return type dict

Returns

Response Syntax

```
{
  'Preset': {
    'Id': 'string',
    'Arn': 'string',
    'Name': 'string',
    'Description': 'string',
    'Container': 'string',
    'Audio': {
      'Codec': 'string',
      'SampleRate': 'string',
      'BitRate': 'string',
      'Channels': 'string',
      'AudioPackingMode': 'string',
      'CodecOptions': {
        'Profile': 'string',
        'BitDepth': 'string',
        'BitOrder': 'string',
        'Signed': 'string'
      }
    },
  },
  'Video': {
    'Codec': 'string',
    'CodecOptions': {
      'string': 'string'
    },
    'KeyframesMaxDist': 'string',
    'FixedGOP': 'string',
    'BitRate': 'string',
    'FrameRate': 'string',
  },
}
```

```

        'MaxFrameRate': 'string',
        'Resolution': 'string',
        'AspectRatio': 'string',
        'MaxWidth': 'string',
        'MaxHeight': 'string',
        'DisplayAspectRatio': 'string',
        'SizingPolicy': 'string',
        'PaddingPolicy': 'string',
        'Watermarks': [
            {
                'Id': 'string',
                'MaxWidth': 'string',
                'MaxHeight': 'string',
                'SizingPolicy': 'string',
                'HorizontalAlign': 'string',
                'HorizontalOffset': 'string',
                'VerticalAlign': 'string',
                'VerticalOffset': 'string',
                'Opacity': 'string',
                'Target': 'string'
            },
        ]
    },
    'Thumbnails': {
        'Format': 'string',
        'Interval': 'string',
        'Resolution': 'string',
        'AspectRatio': 'string',
        'MaxWidth': 'string',
        'MaxHeight': 'string',
        'SizingPolicy': 'string',
        'PaddingPolicy': 'string'
    },
    'Type': 'string'
}

```

Response Structure

- *(dict)* –

The ReadPresetResponse structure.

- **Preset** (*dict*) –

A section of the response body that provides information about the preset.

- * **Id** (*string*) –

Identifier for the new preset. You use this value to get settings for the preset or to delete it.

- * **Arn** (*string*) –

The Amazon Resource Name (ARN) for the preset.

- * **Name** (*string*) –

The name of the preset.

- * **Description** (*string*) –

A description of the preset.

- * **Container** (*string*) –

The container type for the output file. Valid values include `flac`,

flv , fmp4 , gif , mp3 , mp4 , mpg , mxf , oga , ogg , ts , and webm .

* **Audio** (*dict*) –

A section of the response body that provides information about the audio preset values.

- **Codec** (*string*) –

The audio codec for the output file. Valid values include `aac` , `flac` , `mp2` , `mp3` , `pcm` , and `vorbis` .

- **SampleRate** (*string*) –

The sample rate of the audio stream in the output file, in Hertz. Valid values include:

`auto` , `22050` , `32000` , `44100` , `48000` , `96000`

If you specify `auto` , Elastic Transcoder automatically detects the sample rate.

- **BitRate** (*string*) –

The bit rate of the audio stream in the output file, in kilobits/second. Enter an integer between 64 and 320, inclusive.

- **Channels** (*string*) –

The number of audio channels in the output file. The following values are valid:

`auto` , `0` , `1` , `2`

One channel carries the information played by a single speaker. For example, a stereo track with two channels sends one channel to the left speaker, and the other channel to the right speaker. The output channels are organized into tracks. If you want Elastic Transcoder to automatically detect the number of audio channels in the input file and use that value for the output file, select `auto` .

The output of a specific channel value and inputs are as follows:

- **auto channel specified, with any input:** Pass through up to eight input channels.
- **0 channels specified, with any input:** Audio omitted from the output.
- **1 channel specified, with at least one input channel:** Mono sound.
- **2 channels specified, with any input:** Two identical mono channels or stereo. For more information about tracks, see `Audio:AudioPackingMode` .

For more information about how Elastic Transcoder organizes channels and tracks, see `Audio:AudioPackingMode` .

- **AudioPackingMode** (*string*) –

The method of organizing audio channels and tracks. Use `Audio:Channels` to specify the number of channels in your output, and `Audio:AudioPackingMode` to specify the number of tracks and their relation to the channels. If you do not specify an `Audio:AudioPackingMode` , Elastic Transcoder uses `SingleTrack` .

The following values are valid:

`SingleTrack` , `OneChannelPerTrack` , and `OneChannelPerTrackWithMosTo8Tracks`

When you specify `SingleTrack` , Elastic Transcoder creates a single track for your output. The track can have up to eight channels. Use `SingleTrack` for all non-mxf containers.

The outputs of `SingleTrack` for a specific channel value and inputs are as follows:

- **0 channels with any input:** Audio omitted from the output
- **1, 2, or auto channels with no audio input:** Audio omitted from the output
- **1 channel with any input with audio:** One track with one channel, downmixed if necessary
- **2 channels with one track with one channel:** One track with two identical channels
- **2 or auto channels with two tracks with one channel each:** One track with two channels
- **2 or auto channels with one track with two channels:** One track with two channels
- **2 channels with one track with multiple channels:** One track with two channels
- **auto channels with one track with one channel:** One track with one channel
- **auto channels with one track with multiple channels:** One track with multiple channels

When you specify `OneChannelPerTrack` , Elastic Transcoder creates a new track for every channel in your output. Your output can have up to eight single-channel tracks.

The outputs of `OneChannelPerTrack` for a specific channel value and inputs are as follows:

- **0 channels with any input:** Audio omitted from the output
- **1, 2, or auto channels with no audio input:** Audio omitted from the output
- **1 channel with any input with audio:** One track with one channel, downmixed if necessary
- **2 channels with one track with one channel:** Two tracks with one identical channel each
- **2 or auto channels with two tracks with one channel each:** Two tracks with one channel each
- **2 or auto channels with one track with two channels:** Two tracks with one channel each
- **2 channels with one track with multiple channels:** Two tracks with one channel each
- **auto channels with one track with one channel:** One track with one channel
- **auto channels with one track with multiple channels:** Up to eight tracks with one channel each

When you specify `OneChannelPerTrackWithMosTo8Tracks` , Elastic Transcoder creates eight single-channel tracks for your output. All tracks that do not contain audio data from an input channel are MOS, or Mit Out Sound, tracks.

The outputs of `OneChannelPerTrackWithMosTo8Tracks` for a specific channel value and inputs are as follows:

- **0 channels with any input:** Audio omitted from the output
- **1, 2, or auto channels with no audio input:** Audio omitted from the output
- **1 channel with any input with audio:** One track with one channel, downmixed if necessary, plus six MOS tracks
- **2 channels with one track with one channel:** Two tracks with one identical channel each, plus six MOS tracks
- **2 or auto channels with two tracks with one channel each:** Two tracks with one channel each, plus six MOS tracks
- **2 or auto channels with one track with two channels:** Two tracks with one channel each, plus six MOS tracks
- **2 channels with one track with multiple channels:** Two tracks with one channel each, plus six MOS tracks
- **auto channels with one track with one channel:** One track with one channel, plus seven MOS tracks
- **auto channels with one track with multiple channels:** Up to eight tracks with one channel each, plus MOS tracks until there are eight tracks in all
- **CodecOptions (dict)** –

If you specified AAC for `Audio:Codec`, this is the AAC compression profile to use. Valid values include:

`auto`, `AAC-LC`, `HE-AAC`, `HE-AACv2`

If you specify `auto`, Elastic Transcoder chooses a profile based on the bit rate of the output file.

- **Profile (string)** –

You can only choose an audio profile when you specify AAC for the value of `Audio:Codec`.

Specify the AAC profile for the output file. Elastic Transcoder supports the following profiles:

- `auto`: If you specify `auto`, Elastic Transcoder will select the profile based on the bit rate selected for the output file.
- `AAC-LC`: The most common AAC profile. Use for bit rates larger than 64 kbps.
- `HE-AAC`: Not supported on some older players and devices. Use for bit rates between 40 and 80 kbps.
- `HE-AACv2`: Not supported on some players and devices. Use for bit rates less than 48 kbps.

All outputs in a `Smooth` playlist must have the same value for `Profile`.

Note: If you created any presets before AAC profiles were added, Elastic Transcoder automatically updated your presets to use AAC-LC. You can change the value as required.

- **BitDepth (string)** –

You can only choose an audio bit depth when you specify `flac` or `pcm` for the value of `Audio:Codec`.

The bit depth of a sample is how many bits of information are included in the audio samples. The higher the bit depth, the

better the audio, but the larger the file.

Valid values are 16 and 24 .

The most common bit depth is 24 .

- **BitOrder** (*string*) –

You can only choose an audio bit order when you specify `pcm` for the value of `Audio:Codec`.

The order the bits of a PCM sample are stored in.

The supported value is `LittleEndian` .

- **Signed** (*string*) –

You can only choose whether an audio sample is signed when you specify `pcm` for the value of `Audio:Codec`.

Whether audio samples are represented with negative and positive numbers (signed) or only positive numbers (unsigned).

The supported value is `Signed` .

- * **Video** (*dict*) –

A section of the response body that provides information about the video preset values.

- **Codec** (*string*) –

The video codec for the output file. Valid values include `gif` , `H.264` , `mpeg2` , and `vp8` . You can only specify `vp8` when the container type is `webm` , `gif` when the container type is `gif` , and `mpeg2` when the container type is `mpg` .

- **CodecOptions** (*dict*) –

- Profile (H.264/VP8 Only)**

The H.264 profile that you want to use for the output file. Elastic Transcoder supports the following profiles:

- `baseline` : The profile most commonly used for videoconferencing and for mobile applications.
- `main` : The profile used for standard-definition digital TV broadcasts.
- `high` : The profile used for high-definition digital TV broadcasts and for Blu-ray discs.

- Level (H.264 Only)**

The H.264 level that you want to use for the output file. Elastic Transcoder supports the following levels:

1 , 1b , 1.1 , 1.2 , 1.3 , 2 , 2.1 , 2.2 , 3 , 3.1 , 3.2 , 4 , 4.1

- MaxReferenceFrames (H.264 Only)**

Applicable only when the value of `Video:Codec` is `H.264`. The maximum number of previously decoded frames to use as a reference for decoding future frames. Valid values are integers 0 through 16, but we recommend that you not use a value greater than the following:

```
Min(Floor(Maximum decoded
picture buffer in macroblocks *

```

```
256 / (Width in pixels * Height  
in pixels)), 16)
```

where *Width in pixels* and *Height in pixels* represent either `MaxWidth` and `MaxHeight`, or `Resolution`. *Maximum decoded picture buffer in macroblocks* depends on the value of the `Level` object. See the list below. (A macroblock is a block of pixels measuring 16x16.)

- 1 - 396
- 1b - 396
- 1.1 - 900
- 1.2 - 2376
- 1.3 - 2376
- 2 - 2376
- 2.1 - 4752
- 2.2 - 8100
- 3 - 8100
- 3.1 - 18000
- 3.2 - 20480
- 4 - 32768
- 4.1 - 32768

MaxBitRate (Optional, H.264/MPEG2/VP8 only)

The maximum number of bits per second in a video buffer; the size of the buffer is specified by `BufferSize`. Specify a value between 16 and 62,500. You can reduce the bandwidth required to stream a video by reducing the maximum bit rate, but this also reduces the quality of the video.

BufferSize (Optional, H.264/MPEG2/VP8 only)

The maximum number of bits in any *x* seconds of the output video. This window is commonly 10 seconds, the standard segment duration when you're using FMP4 or MPEG-TS for the container type of the output video. Specify an integer greater than 0. If you specify `MaxBitRate` and omit `BufferSize`, Elastic Transcoder sets `BufferSize` to 10 times the value of `MaxBitRate`.

InterlacedMode (Optional, H.264/MPEG2 Only)

The interlace mode for the output video.

Interlaced video is used to double the perceived frame rate for a video by interlacing two fields (one field on every other line, the other field on the other lines) so that the human eye registers multiple pictures per frame. Interlacing reduces the bandwidth required for transmitting a video, but can result in blurred images and flickering.

Valid values include `Progressive` (no interlacing, top to bottom), `TopFirst` (top field first), `BottomFirst` (bottom field first), and `Auto`.

If `InterlaceMode` is not specified, Elastic Transcoder uses `Progressive` for the output. If `Auto` is specified, Elastic Transcoder interlaces the output.

ColorSpaceConversionMode (Optional, H.264/MPEG2 Only)

The color space conversion Elastic Transcoder applies to the output video. Color spaces are the algorithms used by the computer to store information about how to render color. Bt . 601 is the standard for standard definition video, while Bt . 709 is the standard for high definition video.

Valid values include `None` , `Bt709toBt601` , `Bt601toBt709` , and `Auto` .

If you chose `Auto` for `ColorSpaceConversionMode` and your output is interlaced, your frame rate is one of `23.97` , `24` , `25` , `29.97` , `50` , or `60` , your `SegmentDuration` is null, and you are using one of the resolution changes from the list below, Elastic Transcoder applies the following color space conversions:

- *Standard to HD, 720x480 to 1920x1080* - Elastic Transcoder applies `Bt601ToBt709`
- *Standard to HD, 720x576 to 1920x1080* - Elastic Transcoder applies `Bt601ToBt709`
- *HD to Standard, 1920x1080 to 720x480* - Elastic Transcoder applies `Bt709ToBt601`
- *HD to Standard, 1920x1080 to 720x576* - Elastic Transcoder applies `Bt709ToBt601`

Note: Elastic Transcoder may change the behavior of the `ColorspaceConversionMode` `Auto` mode in the future. All outputs in a playlist must use the same `ColorSpaceConversionMode` .

If you do not specify a `ColorSpaceConversionMode` , Elastic Transcoder does not change the color space of a file. If you are unsure what `ColorSpaceConversionMode` was applied to your output file, you can check the `AppliedColorSpaceConversion` parameter included in your job response. If your job does not have an `AppliedColorSpaceConversion` in its response, no `ColorSpaceConversionMode` was applied.

ChromaSubsampling

The sampling pattern for the chroma (color) channels of the output video. Valid values include `yuv420p` and `yuv422p` .

`yuv420p` samples the chroma information of every other horizontal and every other vertical line, `yuv422p` samples the color information of every horizontal line and every other vertical line.

LoopCount (Gif Only)

The number of times you want the output gif to loop. Valid values include `Infinite` and integers between 0 and 100 , inclusive.

- *(string)* –
- *(string)* –
- **KeyframesMaxDist** *(string)* –

Applicable only when the value of Video:Codec is one of H.264 , MPEG2 , or VP8 .

The maximum number of frames between key frames. Key frames are fully encoded frames; the frames between key frames are encoded based, in part, on the content of the key frames. The value is an integer formatted as a string; valid values are between 1 (every frame is a key frame) and 100000, inclusive. A higher value results in higher compression but may also discernibly decrease video quality.

For Smooth outputs, the FrameRate must have a constant ratio to the KeyframesMaxDist . This allows Smooth playlists to switch between different quality levels while the file is being played.

For example, an input file can have a FrameRate of 30 with a KeyframesMaxDist of 90. The output file then needs to have a ratio of 1:3. Valid outputs would have FrameRate of 30, 25, and 10, and KeyframesMaxDist of 90, 75, and 30, respectively.

Alternately, this can be achieved by setting FrameRate to auto and having the same values for MaxFrameRate and KeyframesMaxDist .

- **FixedGOP** (*string*) –

Applicable only when the value of Video:Codec is one of H.264 , MPEG2 , or VP8 .

Whether to use a fixed value for FixedGOP . Valid values are true and false :

- true : Elastic Transcoder uses the value of KeyframesMaxDist for the distance between key frames (the number of frames in a group of pictures, or GOP).
- false : The distance between key frames can vary.

Warning: FixedGOP must be set to true for fmp4 containers.

- **BitRate** (*string*) –

The bit rate of the video stream in the output file, in kilobits/second. Valid values depend on the values of Level and Profile . If you specify auto , Elastic Transcoder uses the detected bit rate of the input source. If you specify a value other than auto , we recommend that you specify a value less than or equal to the maximum H.264-compliant value listed for your level and profile:

*Level - Maximum video bit rate in kilobits/second
(baseline and main Profile) : maximum video bit
rate in kilobits/second (high Profile)*

- 1 - 64 : 80
- 1b - 128 : 160
- 1.1 - 192 : 240
- 1.2 - 384 : 480
- 1.3 - 768 : 960

- 2 - 2000 : 2500
- 3 - 10000 : 12500
- 3.1 - 14000 : 17500
- 3.2 - 20000 : 25000
- 4 - 20000 : 25000
- 4.1 - 50000 : 62500
- **FrameRate** (*string*) –

The frames per second for the video stream in the output file.
Valid values include:

`auto`, `10`, `15`, `23.97`, `24`, `25`, `29.97`, `30`, `60`

If you specify `auto`, Elastic Transcoder uses the detected frame rate of the input source. If you specify a frame rate, we recommend that you perform the following calculation:

$$\text{Frame rate} = \frac{\text{maximum recommended decoding speed in luma samples/second}}{(\text{width in pixels} * \text{height in pixels})}$$

where:

- *width in pixels* and *height in pixels* represent the Resolution of the output video.
- *maximum recommended decoding speed in Luma samples/second* is less than or equal to the maximum value listed in the following table, based on the value that you specified for Level.

The maximum recommended decoding speed in Luma samples/second for each level is described in the following list (*Level - Decoding speed*):

- 1 - 380160
- 1b - 380160
- 1.1 - 76800
- 1.2 - 1536000
- 1.3 - 3041280
- 2 - 3041280
- 2.1 - 5068800
- 2.2 - 5184000
- 3 - 10368000
- 3.1 - 27648000
- 3.2 - 55296000
- 4 - 62914560
- 4.1 - 62914560
- **MaxFrameRate** (*string*) –

If you specify `auto` for `FrameRate`, Elastic Transcoder uses the frame rate of the input video for the frame rate of the output video. Specify the maximum frame rate that you want Elastic Transcoder to use when the frame rate of the input video is greater than the desired maximum frame rate of the output video. Valid values include: `10`, `15`, `23.97`, `24`, `25`, `29.97`, `30`, `60`.

- **Resolution** (*string*) –

Warning: To better control resolution and aspect ratio of output videos, we recommend that you use the values `MaxWidth`, `MaxHeight`, `SizingPolicy`, `PaddingPolicy`, and `DisplayAspectRatio` instead of `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The width and height of the video in the output file, in pixels. Valid values are `auto` and *width x height* :

- `auto` : Elastic Transcoder attempts to preserve the width and height of the input file, subject to the following rules.
- **width* x *height** : The width and height of the output video in pixels.

Note the following about specifying the width and height:

- The width must be an even integer between 128 and 4096, inclusive.
- The height must be an even integer between 96 and 3072, inclusive.
- If you specify a resolution that is less than the resolution of the input file, Elastic Transcoder rescales the output file to the lower resolution.
- If you specify a resolution that is greater than the resolution of the input file, Elastic Transcoder rescales the output to the higher resolution.
- We recommend that you specify a resolution for which the product of width and height is less than or equal to the applicable value in the following list (*List - Max width x height value*) :
- 1 - 25344
- 1b - 25344
- 1.1 - 101376
- 1.2 - 101376
- 1.3 - 101376
- 2 - 101376
- 2.1 - 202752
- 2.2 - 404720
- 3 - 404720
- 3.1 - 921600
- 3.2 - 1310720
- 4 - 2097152
- 4.1 - 2097152
- **AspectRatio** (*string*) –

Warning: To better control resolution and aspect ratio of output videos, we recommend that you use the values `MaxWidth`, `MaxHeight`, `SizingPolicy`, `PaddingPolicy`, and `DisplayAspectRatio` instead of `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The display aspect ratio of the video in the output file. Valid

values include:

`auto`, `1:1`, `4:3`, `3:2`, `16:9`

If you specify `auto`, Elastic Transcoder tries to preserve the aspect ratio of the input file.

If you specify an aspect ratio for the output file that differs from aspect ratio of the input file, Elastic Transcoder adds pillarboxing (black bars on the sides) or letterboxing (black bars on the top and bottom) to maintain the aspect ratio of the active region of the video.

- **MaxWidth** (*string*) –

The maximum width of the output video in pixels. If you specify `auto`, Elastic Transcoder uses 1920 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 128 and 4096.

- **MaxHeight** (*string*) –

The maximum height of the output video in pixels. If you specify `auto`, Elastic Transcoder uses 1080 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 96 and 3072.

- **DisplayAspectRatio** (*string*) –

The value that Elastic Transcoder adds to the metadata in the output file.

- **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of the output video:

- **Fit** : Elastic Transcoder scales the output video so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Fill** : Elastic Transcoder scales the output video so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output video and then crops it in the dimension (if any) that exceeds the maximum value.
- **Stretch** : Elastic Transcoder stretches the output video to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input video and the output video are different, the output video will be distorted.
- **Keep** : Elastic Transcoder does not scale the output video. If either dimension of the input video exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output video.
- **ShrinkToFit** : Elastic Transcoder scales the output video down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the video up.
- **ShrinkToFill** : Elastic Transcoder scales the output video down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without

dropping below either value. If you specify this option, Elastic Transcoder does not scale the video up.

- **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder may add black bars to the top and bottom and/or left and right sides of the output video to make the total size of the output video match the values that you specified for `MaxWidth` and `MaxHeight`.

- **Watermarks** (*list*) –

Settings for the size, location, and opacity of graphics that you want Elastic Transcoder to overlay over videos that are transcoded using this preset. You can specify settings for up to four watermarks. Watermarks appear in the specified size and location, and with the specified opacity for the duration of the transcoded video.

Watermarks can be in `.png` or `.jpg` format. If you want to display a watermark that is not rectangular, use the `.png` format, which supports transparency.

When you create a job that uses this preset, you specify the `.png` or `.jpg` graphics that you want Elastic Transcoder to include in the transcoded videos. You can specify fewer graphics in the job than you specify watermark settings in the preset, which allows you to use the same preset for up to four watermarks that have different dimensions.

- (*dict*) –

Settings for the size, location, and opacity of graphics that you want Elastic Transcoder to overlay over videos that are transcoded using this preset. You can specify settings for up to four watermarks. Watermarks appear in the specified size and location, and with the specified opacity for the duration of the transcoded video.

Watermarks can be in `.png` or `.jpg` format. If you want to display a watermark that is not rectangular, use the `.png` format, which supports transparency.

When you create a job that uses this preset, you specify the `.png` or `.jpg` graphics that you want Elastic Transcoder to include in the transcoded videos. You can specify fewer graphics in the job than you specify watermark settings in the preset, which allows you to use the same preset for up to four watermarks that have different dimensions.

- **Id** (*string*) – A unique identifier for the settings for one watermark. The value of `Id` can be up to 40 characters long.

- **MaxWidth** (*string*) –

The maximum width of the watermark in one of the following formats:

- number of pixels (px): The minimum value is 16 pixels, and the maximum value is the value of `MaxWidth`.
- integer percentage (%): The range of valid values is 0 to 100. Use the value of `Target` to specify whether you want Elastic

Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the calculation.

If you specify the value in pixels, it must be less than or equal to the value of `MaxWidth`.

- **MaxHeight** (*string*) –

The maximum height of the watermark in one of the following formats:

- number of pixels (px): The minimum value is 16 pixels, and the maximum value is the value of `MaxHeight`.
- integer percentage (%): The range of valid values is 0 to 100. Use the value of `Target` to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the calculation.

If you specify the value in pixels, it must be less than or equal to the value of `MaxHeight`.

- **SizingPolicy** (*string*) –

A value that controls scaling of the watermark:

- **Fit** : Elastic Transcoder scales the watermark so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Stretch** : Elastic Transcoder stretches the watermark to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the watermark and the values of `MaxWidth` and `MaxHeight` are different, the watermark will be distorted.
- **ShrinkToFit** : Elastic Transcoder scales the watermark down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the watermark up.

- **HorizontalAlign** (*string*) –

The horizontal position of the watermark unless you specify a non-zero value for `HorizontalOffset`:

- **Left** : The left edge of the watermark is aligned with the left border of the video.
- **Right** : The right edge of the watermark is aligned with the right border of the video.
- **Center** : The watermark is centered between the left and right borders.

- **HorizontalOffset** (*string*) –

The amount by which you want the horizontal position of the watermark to be offset from the position specified by `HorizontalAlign`:

- number of pixels (px): The minimum value is 0 pixels, and the maximum value is the value of `MaxWidth`.
- integer percentage (%): The range of valid values is 0 to 100. For example, if you specify `Left` for `HorizontalAlign` and `5px` for `HorizontalOffset`, the left side of the watermark appears 5 pixels from the left border of the output video.

`HorizontalOffset` is only valid when the value of

`HorizontalAlign` is `Left` or `Right`. If you specify an offset that causes the watermark to extend beyond the left or right border and Elastic Transcoder has not added black bars, the watermark is cropped. If Elastic Transcoder has added black bars, the watermark extends into the black bars. If the watermark extends beyond the black bars, it is cropped.

Use the value of `Target` to specify whether you want to include the black bars that are added by Elastic Transcoder, if any, in the offset calculation.

- **VerticalAlign** (*string*) –

The vertical position of the watermark unless you specify a non-zero value for `VerticalOffset`:

- **Top**: The top edge of the watermark is aligned with the top border of the video.
- **Bottom**: The bottom edge of the watermark is aligned with the bottom border of the video.
- **Center**: The watermark is centered between the top and bottom borders.
- **VerticalOffset** (*string*) – `VerticalOffset`

The amount by which you want the vertical position of the watermark to be offset from the position specified by `VerticalAlign`:

- number of pixels (px): The minimum value is 0 pixels, and the maximum value is the value of `MaxHeight`.
- integer percentage (%): The range of valid values is 0 to 100. For example, if you specify `Top` for `VerticalAlign` and `5px` for `VerticalOffset`, the top of the watermark appears 5 pixels from the top border of the output video.

`VerticalOffset` is only valid when the value of `VerticalAlign` is `Top` or `Bottom`.

If you specify an offset that causes the watermark to extend beyond the top or bottom border and Elastic Transcoder has not added black bars, the watermark is cropped. If Elastic Transcoder has added black bars, the watermark extends into the black bars. If the watermark extends beyond the black bars, it is cropped.

Use the value of `Target` to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the offset calculation.

- **Opacity** (*string*) –

A percentage that indicates how much you want a watermark to obscure the video in the location where it appears. Valid values are 0 (the watermark is invisible) to 100 (the watermark completely obscures the video in the specified location). The datatype of `Opacity` is float.

Elastic Transcoder supports transparent .png graphics. If you use a transparent .png, the transparent portion of the video appears as if you had specified a value of 0 for `Opacity`. The .jpg file format doesn't support transparency.

- **Target** (*string*) –

A value that determines how Elastic Transcoder interprets values that you specified for `HorizontalOffset`, `VerticalOffset`, `MaxWidth`, and `MaxHeight`:

- **Content**: `HorizontalOffset` and `VerticalOffset` values are calculated based on the borders of the video excluding black bars added by Elastic Transcoder, if any. In addition, `MaxWidth` and `MaxHeight`, if specified as a percentage, are calculated based on the borders of the video excluding black bars added by Elastic Transcoder, if any.
- **Frame**: `HorizontalOffset` and `VerticalOffset` values are calculated based on the borders of the video including black bars added by Elastic Transcoder, if any. In addition, `MaxWidth` and `MaxHeight`, if specified as a percentage, are calculated based on the borders of the video including black bars added by Elastic Transcoder, if any.

* **Thumbnails** (*dict*) –

A section of the response body that provides information about the thumbnail preset values, if any.

- **Format** (*string*) –

The format of thumbnails, if any. Valid values are `jpg` and `png`.

You specify whether you want Elastic Transcoder to create thumbnails when you create a job.

- **Interval** (*string*) –

The approximate number of seconds between thumbnails. Specify an integer value.

- **Resolution** (*string*) –

Warning: To better control resolution and aspect ratio of thumbnails, we recommend that you use the values `MaxWidth`, `MaxHeight`, `SizingPolicy`, and `PaddingPolicy` instead of `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The width and height of thumbnail files in pixels. Specify a value in the format `*width* x *height*` where both values are even integers. The values cannot exceed the width and height that you specified in the `Video:Resolution` object.

- **AspectRatio** (*string*) –

Warning: To better control resolution and aspect ratio of thumbnails, we recommend that you use the values `MaxWidth`, `MaxHeight`, `SizingPolicy`, and `PaddingPolicy` instead of `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The aspect ratio of thumbnails. Valid values include:

`auto`, `1:1`, `4:3`, `3:2`, `16:9`

If you specify `auto`, Elastic Transcoder tries to preserve the aspect ratio of the video in the output file.

- **MaxWidth** (*string*) –

The maximum width of thumbnails in pixels. If you specify `auto`, Elastic Transcoder uses 1920 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 32 and 4096.

- **MaxHeight** (*string*) –

The maximum height of thumbnails in pixels. If you specify `auto`, Elastic Transcoder uses 1080 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 32 and 3072.

- **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of thumbnails:

- **Fit** : Elastic Transcoder scales thumbnails so they match the value that you specified in thumbnail `MaxWidth` or `MaxHeight` settings without exceeding the other value.
- **Fill** : Elastic Transcoder scales thumbnails so they match the value that you specified in thumbnail `MaxWidth` or `MaxHeight` settings and matches or exceeds the other value. Elastic Transcoder centers the image in thumbnails and then crops in the dimension (if any) that exceeds the maximum value.
- **Stretch** : Elastic Transcoder stretches thumbnails to match the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings. If the relative proportions of the input video and thumbnails are different, the thumbnails will be distorted.
- **Keep** : Elastic Transcoder does not scale thumbnails. If either dimension of the input video exceeds the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings, Elastic Transcoder crops the thumbnails.
- **ShrinkToFit** : Elastic Transcoder scales thumbnails down so that their dimensions match the values that you specified for at least one of thumbnail `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale thumbnails up.
- **ShrinkToFill** : Elastic Transcoder scales thumbnails down so that their dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale thumbnails up.
- **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder may add black bars to the top and bottom and/or left and right sides of thumbnails to make the total size of the thumbnails match the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings.

- * **Type** (*string*) –

Whether the preset is a default preset provided by Elastic Transcoder

(`System`) or a preset that you have defined (`Custom`).

test_role (**kwargs)

The `TestRole` operation tests the IAM role used to create the pipeline.

The `TestRole` action lets you determine whether the IAM role you are using has sufficient permissions to let Elastic Transcoder perform tasks associated with the transcoding process. The action attempts to assume the specified IAM role, checks read access to the input and output buckets, and tries to send a test notification to Amazon SNS topics that you specify.

Request Syntax

```
response = client.test_role(
    Role='string',
    InputBucket='string',
    OutputBucket='string',
    Topics=[
        'string',
    ]
)
```

Parameters

- **Role** (*string*) – [REQUIRED]

The IAM Amazon Resource Name (ARN) for the role that you want Elastic Transcoder to test.

- **InputBucket** (*string*) – [REQUIRED]

The Amazon S3 bucket that contains media files to be transcoded. The action attempts to read from this bucket.

- **OutputBucket** (*string*) – [REQUIRED]

The Amazon S3 bucket that Elastic Transcoder will write transcoded media files to. The action attempts to read from this bucket.

- **Topics** (*list*) – [REQUIRED]

The ARNs of one or more Amazon Simple Notification Service (Amazon SNS) topics that you want the action to send a test notification to.

– (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'Success': 'string',
    'Messages': [
        'string',
    ]
}
```

Response Structure

- (*dict*) –

The `TestRoleResponse` structure.

– **Success** (*string*) –

If the operation is successful, this value is `true`; otherwise, the value is `false`.

– **Messages** (*list*) –

If the `Success` element contains `false`, this value is an array of one or more error messages that were generated during the test process.

* (*string*) –

update_pipeline (***kwargs*)

Use the `UpdatePipeline` operation to update settings for a pipeline.

Warning: When you change pipeline settings, your changes take effect immediately. Jobs that you have already submitted and that Elastic Transcoder has not started to process are affected in addition to jobs that you submit after you change settings.

Request Syntax

```
response = client.update_pipeline(
    Id='string',
    Name='string',
    InputBucket='string',
    Role='string',
    AwsKmsKeyArn='string',
    Notifications={
        'Progressing': 'string',
        'Completed': 'string',
        'Warning': 'string',
        'Error': 'string'
    },
    ContentConfig={
        'Bucket': 'string',
        'StorageClass': 'string',
        'Permissions': [
            {
                'GranteeType': 'string',
                'Grantee': 'string',
                'Access': [
                    'string',
                ]
            }
        ],
    },
    ThumbnailConfig={
        'Bucket': 'string',
        'StorageClass': 'string',
        'Permissions': [
            {
                'GranteeType': 'string',
                'Grantee': 'string',
                'Access': [
                    'string',
                ]
            }
        ],
    },
)
```

Parameters

- **Id** (*string*) – [REQUIRED]

The ID of the pipeline that you want to update.

- **Name** (*string*) – The name of the pipeline. We recommend that the name be unique within the AWS account, but uniqueness is not enforced.

Constraints: Maximum 40 characters

- **InputBucket** (*string*) – The Amazon S3 bucket in which you saved the media files that you want to transcode and the graphics that you want to use as water-marks.
- **Role** (*string*) – The IAM Amazon Resource Name (ARN) for the role that you want Elastic Transcoder to use to transcode jobs for this pipeline.
- **AwsKmsKeyArn** (*string*) – The AWS Key Management Service (AWS KMS) key that you want to use with this pipeline.

If you use either `S3` or `S3-AWS-KMS` as your `Encryption:Mode`, you don't need to provide a key with your job because a default key, known as an AWS-KMS key, is created for you automatically. You need to provide an AWS-KMS key only if you want to use a non-default AWS-KMS key, or if you are using an `Encryption:Mode` of `AES-PKCS7`, `AES-CTR`, or `AES-GCM`.

- **Notifications** (*dict*) – The Amazon Simple Notification Service (Amazon SNS) topic or topics to notify in order to report job status.

Warning: To receive notifications, you must also subscribe to the new topic in the Amazon SNS console.

- **Progressing** (*string*) –

The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process the job.

- **Completed** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing the job.

- **Warning** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition.

- **Error** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition.

- **ContentConfig** (*dict*) – The optional `ContentConfig` object specifies information about the Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists: which bucket to use, which users you want to have access to the files, the type of access you want users to have, and the storage class that you want to assign to the files.

If you specify values for `ContentConfig`, you must also specify values for `ThumbnailConfig`.

If you specify values for `ContentConfig` and `ThumbnailConfig`, omit the `OutputBucket` object.

- **Bucket** : The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists.
- **Permissions** (Optional): The `Permissions` object specifies which users you want to have access to transcoded files and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.
- **Grantee Type** : Specify the type of value that appears in the `Grantee` object: * **Canonical** : The value in the `Grantee` object is either the

canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. For more information about canonical user IDs, see Access Control List (ACL) Overview in the Amazon Simple Storage Service Developer Guide. For more information about using CloudFront origin access identities to require that users use CloudFront URLs instead of Amazon S3 URLs, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content. .. warning::A canonical user ID is not the same as an AWS account number.

- **Email** : The value in the `Grantee` object is the registered email address of an AWS account.
- **Group** : The value in the `Grantee` object is one of the following predefined Amazon S3 groups: `AllUsers` , `AuthenticatedUsers` , or `LogDelivery` .
- **Grantee** : The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group
- **Access** : The permission that you want to give to the AWS user that you specified in `Grantee` . Permissions are granted on the files that Elastic Transcoder adds to the bucket, including playlists and video files. Valid values include: * **READ** : The grantee can read the objects and metadata for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **READ_ACP** : The grantee can read the object ACL for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **WRITE_ACP** : The grantee can write the ACL for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **FULL_CONTROL** : The grantee has **READ** , **READ_ACP** , and **WRITE_ACP** permissions for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **StorageClass** : The Amazon S3 storage class, `Standard` or `ReducedRedundancy` , that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.
- **Bucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. Specify this value when all of the following are true:

- * You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.
- * You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- * You do not want to specify the permissions that Elastic Transcoder grants to the files.
- * You want to associate the transcoded files and thumbnails with the Amazon S3 Standard storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit `OutputBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

- **StorageClass** (*string*) –

The Amazon S3 storage class, `Standard` or `ReducedRedundancy` , that you want Elastic Transcoder to assign to the video files and playlists

that it stores in your Amazon S3 bucket.

– **Permissions** (*list*) –

Optional. The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions`, Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role`. If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions`, Elastic Transcoder grants full control over the transcoded files and playlists to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

* (*dict*) –

The `Permission` structure.

· **GranteeType** (*string*) –

The type of value that appears in the Grantee object:

- `Canonical`: Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.
- `Email`: The registered email address of an AWS account.
- `Group`: One of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.
- **Grantee** (*string*) –

The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group.

· **Access** (*list*) –

The permission that you want to give to the AWS user that is listed in Grantee. Valid values include:

- `READ`: The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `READ_ACP`: The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `WRITE_ACP`: The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `FULL_CONTROL`: The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- (*string*) –

- **ThumbnailConfig** (*dict*) – The `ThumbnailConfig` object specifies several values, including the Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files, which users you want to have access to the files, the type of access you want users to have, and the storage class that you want to assign to

the files.

If you specify values for `ContentConfig`, you must also specify values for `ThumbnailConfig` even if you don't want to create thumbnails.

If you specify values for `ContentConfig` and `ThumbnailConfig`, omit the `OutputBucket` object.

- **Bucket** : The Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files.
- **Permissions** (Optional): The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to thumbnail files, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.
- **GranteeType** : Specify the type of value that appears in the `Grantee` object: * **Canonical** : The value in the `Grantee` object is either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.
- **Email** : The value in the `Grantee` object is the registered email address of an AWS account.
- **Group** : The value in the `Grantee` object is one of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.
- **Grantee** : The AWS user or group that you want to have access to thumbnail files. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group.
- **Access** : The permission that you want to give to the AWS user that you specified in `Grantee`. Permissions are granted on the thumbnail files that Elastic Transcoder adds to the bucket. Valid values include: * **READ** : The grantee can read the thumbnails and metadata for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **READ_ACP** : The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **WRITE_ACP** : The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **FULL_CONTROL** : The grantee has **READ**, **READ_ACP**, and **WRITE_ACP** permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **StorageClass** : The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the thumbnails that it stores in your Amazon S3 bucket.
- **Bucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. Specify this value when all of the following are true:

- * You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.
- * You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- * You do not want to specify the permissions that Elastic Transcoder grants to the files.
- * You want to associate the transcoded files and thumbnails with the

Amazon S3 Standard storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit `OutputBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

– **StorageClass** (*string*) –

The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

– **Permissions** (*list*) –

Optional. The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions`, Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role`. If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions`, Elastic Transcoder grants full control over the transcoded files and playlists to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

* (*dict*) –

The `Permission` structure.

· **GranteeType** (*string*) –

The type of value that appears in the Grantee object:

- `Canonical`: Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.
- `Email`: The registered email address of an AWS account.
- `Group`: One of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.
- **Grantee** (*string*) –

The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group.

· **Access** (*list*) –

The permission that you want to give to the AWS user that is listed in Grantee. Valid values include:

- `READ`: The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `READ_ACP`: The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.

- `WRITE_ACP` : The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `FULL_CONTROL` : The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- *(string)* –

Return type dict

Returns

Response Syntax

```
{
  'Pipeline': {
    'Id': 'string',
    'Arn': 'string',
    'Name': 'string',
    'Status': 'string',
    'InputBucket': 'string',
    'OutputBucket': 'string',
    'Role': 'string',
    'AwsKmsKeyArn': 'string',
    'Notifications': {
      'Progressing': 'string',
      'Completed': 'string',
      'Warning': 'string',
      'Error': 'string'
    },
    'ContentConfig': {
      'Bucket': 'string',
      'StorageClass': 'string',
      'Permissions': [
        {
          'GranteeType': 'string',
          'Grantee': 'string',
          'Access': [
            'string',
          ]
        },
      ]
    },
    'ThumbnailConfig': {
      'Bucket': 'string',
      'StorageClass': 'string',
      'Permissions': [
        {
          'GranteeType': 'string',
          'Grantee': 'string',
          'Access': [
            'string',
          ]
        },
      ]
    }
  },
  'Warnings': [
    {
      'Code': 'string',
      'Message': 'string'
    }
  ]
}
```

```
    },
  ]
}
```

Response Structure

- (*dict*) –

When you update a pipeline, Elastic Transcoder returns the values that you specified in the request.

- **Pipeline** (*dict*) –

The pipeline (queue) that is used to manage jobs.

- * **Id** (*string*) –

The identifier for the pipeline. You use this value to identify the pipeline in which you want to perform a variety of operations, such as creating a job or a preset.

- * **Arn** (*string*) –

The Amazon Resource Name (ARN) for the pipeline.

- * **Name** (*string*) –

The name of the pipeline. We recommend that the name be unique within the AWS account, but uniqueness is not enforced.

Constraints: Maximum 40 characters

- * **Status** (*string*) –

The current status of the pipeline:

- **Active**: The pipeline is processing jobs.
- **Paused**: The pipeline is not currently processing jobs.

- * **InputBucket** (*string*) –

The Amazon S3 bucket from which Elastic Transcoder gets media files for transcoding and the graphics files, if any, that you want to use for watermarks.

- * **OutputBucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files, thumbnails, and playlists. Either you specify this value, or you specify both `ContentConfig` and `ThumbnailConfig`.

- * **Role** (*string*) –

The IAM Amazon Resource Name (ARN) for the role that Elastic Transcoder uses to transcode jobs for this pipeline.

- * **AwsKmsKeyArn** (*string*) –

The AWS Key Management Service (AWS KMS) key that you want to use with this pipeline.

If you use either `S3` or `S3-AWS-KMS` as your `Encryption:Mode`, you don't need to provide a key with your job because a default key, known as an AWS-KMS key, is created for you automatically. You need to provide an AWS-KMS key only if you want to use a non-default AWS-KMS key, or if you are using an `Encryption:Mode` of `AES-PKCS7`, `AES-CTR`, or `AES-GCM`.

- * **Notifications** (*dict*) –

The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify to report job status.

Warning: To receive notifications, you must also subscribe to the new topic in the Amazon SNS console.

- **Progressing** (optional): The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process the job.
- **Completed** (optional): The Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing the job.
- **Warning** (optional): The Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition.
- **Error** (optional): The Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition.

- **Progressing** (*string*) –

The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process the job.

- **Completed** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing the job.

- **Warning** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition.

- **Error** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition.

- * **ContentConfig** (*dict*) –

Information about the Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists. Either you specify both `ContentConfig` and `ThumbnailConfig`, or you specify `OutputBucket`.

- **Bucket**: The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists.
- **Permissions**: A list of the users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access that you want them to have. * **GranteeType**: The type of value that appears in the `Grantee` object: * **Canonical**: Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution.
- **Email**: The registered email address of an AWS account.
- **Group**: One of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.
- **Grantee**: The AWS user or group that you want to have access to transcoded files and playlists.
- **Access**: The permission that you want to give to the AWS user that is listed in `Grantee`. Valid values include: * `READ`

- : The grantee can read the objects and metadata for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- `READ_ACP` : The grantee can read the object ACL for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- `WRITE_ACP` : The grantee can write the ACL for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
- `FULL_CONTROL` : The grantee has `READ` , `READ_ACP` , and `WRITE_ACP` permissions for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **StorageClass** : The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.
- **Bucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. Specify this value when all of the following are true:

- You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.
- You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- You do not want to specify the permissions that Elastic Transcoder grants to the files.
- You want to associate the transcoded files and thumbnails with the Amazon S3 `Standard` storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit `OutputBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

- **StorageClass** (*string*) –

The Amazon S3 storage class, `Standard` or `ReducedRedundancy` , that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

- **Permissions** (*list*) –

Optional. The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions` , Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role` . If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions` , Elastic Transcoder grants full control over the transcoded files and playlists to the owner of the role specified by `Role` , and grants no other permissions

to any other user or group.

- *(dict)* –

The `Permission` structure.

- **GranteeType** (*string*) –

The type of value that appears in the Grantee object:

- **Canonical** : Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.
- **Email** : The registered email address of an AWS account.
- **Group** : One of the following predefined Amazon S3 groups: `AllUsers` , `AuthenticatedUsers` , or `LogDelivery` .
- **Grantee** (*string*) –

The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group.

- **Access** (*list*) –

The permission that you want to give to the AWS user that is listed in Grantee. Valid values include:

- **READ** : The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **READ_ACP** : The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **WRITE_ACP** : The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **FULL_CONTROL** : The grantee has **READ**, **READ_ACP**, and **WRITE_ACP** permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- (*string*) –

- * **ThumbnailConfig** (*dict*) –

Information about the Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files. Either you specify both `ContentConfig` and `ThumbnailConfig` , or you specify `OutputBucket` .

- **Bucket** : The Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files.
- **Permissions** : A list of the users and/or predefined Amazon S3 groups you want to have access to thumbnail files, and the type of access that you want them to have. * **GranteeType**: The type of value that appears in the Grantee object: * **Canonical** : Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.
- **Email** : The registered email address of an AWS account.
- **Group** : One of the following predefined Amazon S3

`groups:` `AllUsers` , `AuthenticatedUsers` , or `LogDelivery` .

- `Grantee` : The AWS user or group that you want to have access to thumbnail files.
- `Access`: The permission that you want to give to the AWS user that is listed in `Grantee`. Valid values include: * `READ` : The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `READ_ACP` : The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `WRITE_ACP` : The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `FULL_CONTROL` : The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `StorageClass` : The Amazon S3 storage class, `Standard` or `ReducedRedundancy` , that you want Elastic Transcoder to assign to the thumbnails that it stores in your Amazon S3 bucket.

- **Bucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. Specify this value when all of the following are true:

- You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.
- You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- You do not want to specify the permissions that Elastic Transcoder grants to the files.
- You want to associate the transcoded files and thumbnails with the Amazon S3 Standard storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit `OutputBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

- **StorageClass** (*string*) –

The Amazon S3 storage class, `Standard` or `ReducedRedundancy` , that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

- **Permissions** (*list*) –

Optional. The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions` , Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role` . If

you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions`, Elastic Transcoder grants full control over the transcoded files and playlists to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

- *(dict)* –

The `Permission` structure.

- **GranteeType** (*string*) –

The type of value that appears in the Grantee object:

- `Canonical`: Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.
- `Email`: The registered email address of an AWS account.
- `Group`: One of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.
- **Grantee** (*string*) –

The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group.

- **Access** (*list*) –

The permission that you want to give to the AWS user that is listed in Grantee. Valid values include:

- `READ`: The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `READ_ACP`: The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `WRITE_ACP`: The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `FULL_CONTROL`: The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- (*string*) –

- **Warnings** (*list*) –

Elastic Transcoder returns a warning if the resources used by your pipeline are not in the same region as the pipeline.

Using resources in the same region, such as your Amazon S3 buckets, Amazon SNS notification topics, and AWS KMS key, reduces processing time and prevents cross-regional charges.

- (*dict*) –

Elastic Transcoder returns a warning if the resources used by your pipeline are not in the same region as the pipeline.

Using resources in the same region, such as your Amazon S3 buck-

ets, Amazon SNS notification topics, and AWS KMS key, reduces processing time and prevents cross-regional charges.

- **Code** (*string*) –

The code of the cross-regional warning.

- **Message** (*string*) –

The message explaining what resources are in a different region from the pipeline.

Note: AWS KMS keys must be in the same region as the pipeline.

update_pipeline_notifications (***kwargs*)

With the UpdatePipelineNotifications operation, you can update Amazon Simple Notification Service (Amazon SNS) notifications for a pipeline.

When you update notifications for a pipeline, Elastic Transcoder returns the values that you specified in the request.

Request Syntax

```
response = client.update_pipeline_notifications(
    Id='string',
    Notifications={
        'Progressing': 'string',
        'Completed': 'string',
        'Warning': 'string',
        'Error': 'string'
    }
)
```

Parameters

- **Id** (*string*) – **[REQUIRED]**

The identifier of the pipeline for which you want to change notification settings.

- **Notifications** (*dict*) – **[REQUIRED]**

The topic ARN for the Amazon Simple Notification Service (Amazon SNS) topic that you want to notify to report job status.

Warning: To receive notifications, you must also subscribe to the new topic in the Amazon SNS console.

- **Progressing** : The topic ARN for the Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process jobs that are added to this pipeline. This is the ARN that Amazon SNS returned when you created the topic.
- **Completed** : The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing a job. This is the ARN that Amazon SNS returned when you created the topic.
- **Warning** : The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition. This is the ARN that Amazon SNS returned when you created the topic.
- **Error** : The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition. This is the ARN that Amazon SNS returned when you created the topic.
- **Progressing** (*string*) –

The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process the job.

– **Completed** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing the job.

– **Warning** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition.

– **Error** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition.

Return type dict

Returns

Response Syntax

```
{
  'Pipeline': {
    'Id': 'string',
    'Arn': 'string',
    'Name': 'string',
    'Status': 'string',
    'InputBucket': 'string',
    'OutputBucket': 'string',
    'Role': 'string',
    'AwsKmsKeyArn': 'string',
    'Notifications': {
      'Progressing': 'string',
      'Completed': 'string',
      'Warning': 'string',
      'Error': 'string'
    },
  },
  'ContentConfig': {
    'Bucket': 'string',
    'StorageClass': 'string',
    'Permissions': [
      {
        'GranteeType': 'string',
        'Grantee': 'string',
        'Access': [
          'string',
        ]
      },
    ],
  },
  'ThumbnailConfig': {
    'Bucket': 'string',
    'StorageClass': 'string',
    'Permissions': [
      {
        'GranteeType': 'string',
        'Grantee': 'string',
        'Access': [
          'string',
        ]
      },
    ],
  },
}
```

```

    ]
  }
}

```

Response Structure

- (*dict*) –

The `UpdatePipelineNotificationsResponse` structure.

- **Pipeline** (*dict*) –

A section of the response body that provides information about the pipeline.

- * **Id** (*string*) –

The identifier for the pipeline. You use this value to identify the pipeline in which you want to perform a variety of operations, such as creating a job or a preset.

- * **Arn** (*string*) –

The Amazon Resource Name (ARN) for the pipeline.

- * **Name** (*string*) –

The name of the pipeline. We recommend that the name be unique within the AWS account, but uniqueness is not enforced.

Constraints: Maximum 40 characters

- * **Status** (*string*) –

The current status of the pipeline:

- **Active**: The pipeline is processing jobs.
- **Paused**: The pipeline is not currently processing jobs.

- * **InputBucket** (*string*) –

The Amazon S3 bucket from which Elastic Transcoder gets media files for transcoding and the graphics files, if any, that you want to use for watermarks.

- * **OutputBucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files, thumbnails, and playlists. Either you specify this value, or you specify both `ContentConfig` and `ThumbnailConfig`.

- * **Role** (*string*) –

The IAM Amazon Resource Name (ARN) for the role that Elastic Transcoder uses to transcode jobs for this pipeline.

- * **AwsKmsKeyArn** (*string*) –

The AWS Key Management Service (AWS KMS) key that you want to use with this pipeline.

If you use either `S3` or `S3-AWS-KMS` as your `Encryption:Mode`, you don't need to provide a key with your job because a default key, known as an AWS-KMS key, is created for you automatically. You need to provide an AWS-KMS key only if you want to use a non-default AWS-KMS key, or if you are using an `Encryption:Mode` of `AES-PKCS7`, `AES-CTR`, or `AES-GCM`.

* **Notifications** (*dict*) –

The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify to report job status.

Warning: To receive notifications, you must also subscribe to the new topic in the Amazon SNS console.

- **Progressing** (optional): The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process the job.
- **Completed** (optional): The Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing the job.
- **Warning** (optional): The Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition.
- **Error** (optional): The Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition.

- **Progressing** (*string*) –

The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process the job.

- **Completed** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing the job.

- **Warning** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition.

- **Error** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition.

* **ContentConfig** (*dict*) –

Information about the Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists. Either you specify both `ContentConfig` and `ThumbnailConfig`, or you specify `OutputBucket`.

- **Bucket** : The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists.
- **Permissions** : A list of the users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access that you want them to have. * **GranteeType**: The type of value that appears in the `Grantee` object: * **Canonical** : Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution.
- **Email** : The registered email address of an AWS account.
- **Group** : One of the following predefined Amazon S3 groups: `AllUsers` , `AuthenticatedUsers` , or `LogDelivery` .
- **Grantee** : The AWS user or group that you want to have access to transcoded files and playlists.

- **Access** : The permission that you want to give to the AWS user that is listed in `Grantee` . Valid values include:
 - * `READ` : The grantee can read the objects and metadata for objects that Elastic Transcoder adds to the Amazon S3 bucket.
 - * `READ_ACP` : The grantee can read the object ACL for objects that Elastic Transcoder adds to the Amazon S3 bucket.
 - * `WRITE_ACP` : The grantee can write the ACL for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
 - * `FULL_CONTROL` : The grantee has `READ` , `READ_ACP` , and `WRITE_ACP` permissions for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **StorageClass** : The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

- **Bucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. Specify this value when all of the following are true:

- You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.
- You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- You do not want to specify the permissions that Elastic Transcoder grants to the files.
- You want to associate the transcoded files and thumbnails with the Amazon S3 `Standard` storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit `OutputBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

- **StorageClass** (*string*) –

The Amazon S3 storage class, `Standard` or `ReducedRedundancy` , that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

- **Permissions** (*list*) –

Optional. The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions` , Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role` . If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions` , Elastic Transcoder grants full

control over the transcoded files and playlists to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

- *(dict)* –

The `Permission` structure.

- **GranteeType** (*string*) –

The type of value that appears in the Grantee object:

- `Canonical` : Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.
- `Email` : The registered email address of an AWS account.
- `Group` : One of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.
- **Grantee** (*string*) –

The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group.

- **Access** (*list*) –

The permission that you want to give to the AWS user that is listed in Grantee. Valid values include:

- `READ` : The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `READ_ACP` : The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `WRITE_ACP` : The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `FULL_CONTROL` : The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- (*string*) –

- * **ThumbnailConfig** (*dict*) –

Information about the Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files. Either you specify both `ContentConfig` and `ThumbnailConfig`, or you specify `OutputBucket`.

- `Bucket` : The Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files.
- `Permissions` : A list of the users and/or predefined Amazon S3 groups you want to have access to thumbnail files, and the type of access that you want them to have. * **GranteeType**: The type of value that appears in the Grantee object: * `Canonical` : Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.

- **Email** : The registered email address of an AWS account.
- **Group** : One of the following predefined Amazon S3 groups: `AllUsers` , `AuthenticatedUsers` , or `LogDelivery` .
- **Grantee** : The AWS user or group that you want to have access to thumbnail files.
- **Access**: The permission that you want to give to the AWS user that is listed in **Grantee**. Valid values include:
 - * `READ` : The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `READ_ACP` : The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `WRITE_ACP` : The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `FULL_CONTROL` : The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **StorageClass** : The Amazon S3 storage class, `Standard` or `ReducedRedundancy` , that you want Elastic Transcoder to assign to the thumbnails that it stores in your Amazon S3 bucket.

- **Bucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. Specify this value when all of the following are true:

- You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.
- You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- You do not want to specify the permissions that Elastic Transcoder grants to the files.
- You want to associate the transcoded files and thumbnails with the Amazon S3 Standard storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit `OutputBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

- **StorageClass** (*string*) –

The Amazon S3 storage class, `Standard` or `ReducedRedundancy` , that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

- **Permissions** (*list*) –

Optional. The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions` , Elastic Transcoder grants

only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role`. If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions`, Elastic Transcoder grants full control over the transcoded files and playlists to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

- *(dict)* –

The `Permission` structure.

- **GranteeType** (*string*) –

The type of value that appears in the Grantee object:

- `Canonical`: Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning:: A canonical user ID is not the same as an AWS account number.
- `Email`: The registered email address of an AWS account.
- `Group`: One of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.
- **Grantee** (*string*) –

The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group.

- **Access** (*list*) –

The permission that you want to give to the AWS user that is listed in Grantee. Valid values include:

- `READ`: The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `READ_ACP`: The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `WRITE_ACP`: The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `FULL_CONTROL`: The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- (*string*) –

update_pipeline_status (***kwargs*)

The `UpdatePipelineStatus` operation pauses or reactivates a pipeline, so that the pipeline stops or restarts the processing of jobs.

Changing the pipeline status is useful if you want to cancel one or more jobs. You can't cancel jobs after Elastic Transcoder has started processing them; if you pause the pipeline to which you submitted the jobs, you have more time to get the job IDs for the jobs that you want to cancel, and to send a `CancelJob` request.

Request Syntax

```
response = client.update_pipeline_status(
    Id='string',
    Status='string'
)
```

Parameters

- **Id** (*string*) – **[REQUIRED]**

The identifier of the pipeline to update.

- **Status** (*string*) – **[REQUIRED]**

The desired status of the pipeline:

- **Active** : The pipeline is processing jobs.
- **Paused** : The pipeline is not currently processing jobs.

Return type dict

Returns

Response Syntax

```
{
  'Pipeline': {
    'Id': 'string',
    'Arn': 'string',
    'Name': 'string',
    'Status': 'string',
    'InputBucket': 'string',
    'OutputBucket': 'string',
    'Role': 'string',
    'AwsKmsKeyArn': 'string',
    'Notifications': {
      'Progressing': 'string',
      'Completed': 'string',
      'Warning': 'string',
      'Error': 'string'
    },
    'ContentConfig': {
      'Bucket': 'string',
      'StorageClass': 'string',
      'Permissions': [
        {
          'GranteeType': 'string',
          'Grantee': 'string',
          'Access': [
            'string',
          ]
        }
      ],
    },
  },
  'ThumbnailConfig': {
    'Bucket': 'string',
    'StorageClass': 'string',
    'Permissions': [
      {
        'GranteeType': 'string',
        'Grantee': 'string',
        'Access': [
          'string',
        ]
      }
    ],
  },
}
```

```
    },  
  ],  
}
```

Response Structure

- (*dict*) – When you update status for a pipeline, Elastic Transcoder returns the values that you specified in the request.

- **Pipeline** (*dict*) –

A section of the response body that provides information about the pipeline.

- * **Id** (*string*) –

The identifier for the pipeline. You use this value to identify the pipeline in which you want to perform a variety of operations, such as creating a job or a preset.

- * **Arn** (*string*) –

The Amazon Resource Name (ARN) for the pipeline.

- * **Name** (*string*) –

The name of the pipeline. We recommend that the name be unique within the AWS account, but uniqueness is not enforced.

Constraints: Maximum 40 characters

- * **Status** (*string*) –

The current status of the pipeline:

- **Active** : The pipeline is processing jobs.
- **Paused** : The pipeline is not currently processing jobs.

- * **InputBucket** (*string*) –

The Amazon S3 bucket from which Elastic Transcoder gets media files for transcoding and the graphics files, if any, that you want to use for watermarks.

- * **OutputBucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files, thumbnails, and playlists. Either you specify this value, or you specify both `ContentConfig` and `ThumbnailConfig`.

- * **Role** (*string*) –

The IAM Amazon Resource Name (ARN) for the role that Elastic Transcoder uses to transcode jobs for this pipeline.

- * **AwsKmsKeyArn** (*string*) –

The AWS Key Management Service (AWS KMS) key that you want to use with this pipeline.

If you use either `S3` or `S3-AWS-KMS` as your `Encryption:Mode`, you don't need to provide a key with your job because a default key, known as an AWS-KMS key, is created for you automatically. You need to provide an AWS-KMS key only if you want to use a non-default AWS-KMS key, or if you are using an `Encryption:Mode` of `AES-PKCS7`, `AES-CTR`, or `AES-GCM`.

* **Notifications** (*dict*) –

The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify to report job status.

Warning: To receive notifications, you must also subscribe to the new topic in the Amazon SNS console.

- **Progressing** (optional): The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process the job.
- **Completed** (optional): The Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing the job.
- **Warning** (optional): The Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition.
- **Error** (optional): The Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition.

· **Progressing** (*string*) –

The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process the job.

· **Completed** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing the job.

· **Warning** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition.

· **Error** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition.

* **ContentConfig** (*dict*) –

Information about the Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists. Either you specify both `ContentConfig` and `ThumbnailConfig`, or you specify `OutputBucket`.

- **Bucket** : The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists.
- **Permissions** : A list of the users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access that you want them to have. * **GranteeType**: The type of value that appears in the `Grantee` object: * **Canonical** : Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution.
- **Email** : The registered email address of an AWS account.
- **Group** : One of the following predefined Amazon S3 groups: `AllUsers` , `AuthenticatedUsers` , or `LogDelivery` .
- **Grantee** : The AWS user or group that you want to have access to transcoded files and playlists.

- **Access** : The permission that you want to give to the AWS user that is listed in `Grantee` . Valid values include: * `READ` : The grantee can read the objects and metadata for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- `READ_ACP` : The grantee can read the object ACL for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- `WRITE_ACP` : The grantee can write the ACL for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
- `FULL_CONTROL` : The grantee has `READ` , `READ_ACP` , and `WRITE_ACP` permissions for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **StorageClass** : The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

- **Bucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. Specify this value when all of the following are true:

- You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.
- You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- You do not want to specify the permissions that Elastic Transcoder grants to the files.
- You want to associate the transcoded files and thumbnails with the Amazon S3 `Standard` storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit `OutputBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

- **StorageClass** (*string*) –

The Amazon S3 storage class, `Standard` or `ReducedRedundancy` , that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

- **Permissions** (*list*) –

Optional. The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions` , Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role` . If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions` , Elastic Transcoder grants full

control over the transcoded files and playlists to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

- *(dict)* –

The `Permission` structure.

- **GranteeType** (*string*) –

The type of value that appears in the `Grantee` object:

- `Canonical` : Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.
- `Email` : The registered email address of an AWS account.
- `Group` : One of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.
- **Grantee** (*string*) –

The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group.

- **Access** (*list*) –

The permission that you want to give to the AWS user that is listed in `Grantee`. Valid values include:

- `READ` : The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `READ_ACP` : The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `WRITE_ACP` : The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `FULL_CONTROL` : The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- (*string*) –

- * **ThumbnailConfig** (*dict*) –

Information about the Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files. Either you specify both `ContentConfig` and `ThumbnailConfig`, or you specify `OutputBucket`.

- `Bucket` : The Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files.
- `Permissions` : A list of the users and/or predefined Amazon S3 groups you want to have access to thumbnail files, and the type of access that you want them to have. * **GranteeType**: The type of value that appears in the `Grantee` object: * `Canonical` : Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.

- **Email** : The registered email address of an AWS account.
- **Group** : One of the following predefined Amazon S3 groups: `AllUsers` , `AuthenticatedUsers` , or `LogDelivery` .
- **Grantee** : The AWS user or group that you want to have access to thumbnail files.
- **Access**: The permission that you want to give to the AWS user that is listed in **Grantee**. Valid values include: * `READ` : The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `READ_ACP` : The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `WRITE_ACP` : The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `FULL_CONTROL` : The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **StorageClass** : The Amazon S3 storage class, `Standard` or `ReducedRedundancy` , that you want Elastic Transcoder to assign to the thumbnails that it stores in your Amazon S3 bucket.

- **Bucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. Specify this value when all of the following are true:

- You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.
- You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- You do not want to specify the permissions that Elastic Transcoder grants to the files.
- You want to associate the transcoded files and thumbnails with the Amazon S3 Standard storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit `OutputBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

- **StorageClass** (*string*) –

The Amazon S3 storage class, `Standard` or `ReducedRedundancy` , that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

- **Permissions** (*list*) –

Optional. The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions` , Elastic Transcoder grants

only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role`. If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions`, Elastic Transcoder grants full control over the transcoded files and playlists to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

- *(dict)* –

The `Permission` structure.

- **GranteeType** (*string*) –

The type of value that appears in the Grantee object:

- `Canonical`: Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.
- `Email`: The registered email address of an AWS account.
- `Group`: One of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.
- **Grantee** (*string*) –

The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group.

- **Access** (*list*) –

The permission that you want to give to the AWS user that is listed in Grantee. Valid values include:

- `READ`: The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `READ_ACP`: The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `WRITE_ACP`: The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `FULL_CONTROL`: The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- (*string*) –

Paginators

The available paginators are:

- `ElasticTranscoder.Paginator.ListJobsByPipeline`
- `ElasticTranscoder.Paginator.ListJobsByStatus`
- `ElasticTranscoder.Paginator.ListPipelines`
- `ElasticTranscoder.Paginator.ListPresets`

`class ElasticTranscoder.Paginator.ListJobsByPipeline`

```
paginator = client.get_paginator('list_jobs_by_pipeline')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `ElasticTranscoder.Client.list_jobs_by_pipeline()`.

Request Syntax

```
response_iterator = paginator.paginate(
    PipelineId='string',
    Ascending='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **PipelineId** (*string*) – [REQUIRED]

The ID of the pipeline for which you want to get job information.

- **Ascending** (*string*) – To list jobs in chronological order by the date and time that they were submitted, enter `true`. To list jobs in reverse chronological order, enter `false`.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Jobs': [
    {
      'Id': 'string',
      'Arn': 'string',
      'PipelineId': 'string',
      'Input': {
        'Key': 'string',
        'FrameRate': 'string',
        'Resolution': 'string',
        'AspectRatio': 'string',
```

```

    'Interlaced': 'string',
    'Container': 'string',
    'Encryption': {
        'Mode': 'string',
        'Key': 'string',
        'KeyMd5': 'string',
        'InitializationVector': 'string'
    },
    'DetectedProperties': {
        'Width': 123,
        'Height': 123,
        'FrameRate': 'string',
        'FileSize': 123,
        'DurationMillis': 123
    }
},
'Output': {
    'Id': 'string',
    'Key': 'string',
    'ThumbnailPattern': 'string',
    'ThumbnailEncryption': {
        'Mode': 'string',
        'Key': 'string',
        'KeyMd5': 'string',
        'InitializationVector': 'string'
    },
    'Rotate': 'string',
    'PresetId': 'string',
    'SegmentDuration': 'string',
    'Status': 'string',
    'StatusDetail': 'string',
    'Duration': 123,
    'Width': 123,
    'Height': 123,
    'FrameRate': 'string',
    'FileSize': 123,
    'DurationMillis': 123,
    'Watermarks': [
        {
            'PresetWatermarkId': 'string',
            'InputKey': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        }
    ],
    'AlbumArt': {
        'MergePolicy': 'string',
        'Artwork': [
            {
                'InputKey': 'string',
                'MaxWidth': 'string',
                'MaxHeight': 'string',
                'SizingPolicy': 'string',
                'PaddingPolicy': 'string',

```

```
        'AlbumArtFormat': 'string',
        'Encryption': {
            'Mode': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'InitializationVector': 'string'
        }
    },
]
},
'Composition': [
    {
        'TimeSpan': {
            'StartTime': 'string',
            'Duration': 'string'
        }
    },
],
'Captions': {
    'MergePolicy': 'string',
    'CaptionSources': [
        {
            'Key': 'string',
            'Language': 'string',
            'TimeOffset': 'string',
            'Label': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        }
    ],
    'CaptionFormats': [
        {
            'Format': 'string',
            'Pattern': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        }
    ]
},
'Encryption': {
    'Mode': 'string',
    'Key': 'string',
    'KeyMd5': 'string',
    'InitializationVector': 'string'
},
'AppliedColorSpaceConversion': 'string'
},
'Outputs': [
    {
        'Id': 'string',
```

```

'Key': 'string',
'ThumbnailPattern': 'string',
'ThumbnailEncryption': {
    'Mode': 'string',
    'Key': 'string',
    'KeyMd5': 'string',
    'InitializationVector': 'string'
},
'Rotate': 'string',
'PresetId': 'string',
'SegmentDuration': 'string',
'Status': 'string',
'StatusDetail': 'string',
'Duration': 123,
'Width': 123,
'Height': 123,
'FrameRate': 'string',
'FileSize': 123,
'DurationMillis': 123,
'Watermarks': [
    {
        'PresetWatermarkId': 'string',
        'InputKey': 'string',
        'Encryption': {
            'Mode': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'InitializationVector': 'string'
        }
    },
],
'AlbumArt': {
    'MergePolicy': 'string',
    'Artwork': [
        {
            'InputKey': 'string',
            'MaxWidth': 'string',
            'MaxHeight': 'string',
            'SizingPolicy': 'string',
            'PaddingPolicy': 'string',
            'AlbumArtFormat': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        }
    ],
},
'Composition': [
    {
        'TimeSpan': {
            'StartTime': 'string',
            'Duration': 'string'
        }
    },
],

```

```
    'Captions': {
        'MergePolicy': 'string',
        'CaptionSources': [
            {
                'Key': 'string',
                'Language': 'string',
                'TimeOffset': 'string',
                'Label': 'string',
                'Encryption': {
                    'Mode': 'string',
                    'Key': 'string',
                    'KeyMd5': 'string',
                    'InitializationVector': 'string'
                }
            },
        ],
        'CaptionFormats': [
            {
                'Format': 'string',
                'Pattern': 'string',
                'Encryption': {
                    'Mode': 'string',
                    'Key': 'string',
                    'KeyMd5': 'string',
                    'InitializationVector': 'string'
                }
            },
        ],
    },
    'Encryption': {
        'Mode': 'string',
        'Key': 'string',
        'KeyMd5': 'string',
        'InitializationVector': 'string'
    },
    'AppliedColorSpaceConversion': 'string'
],
'OutputKeyPrefix': 'string',
'Playlists': [
    {
        'Name': 'string',
        'Format': 'string',
        'OutputKeys': [
            'string',
        ],
        'HlsContentProtection': {
            'Method': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'InitializationVector': 'string',
            'LicenseAcquisitionUrl': 'string',
            'KeyStoragePolicy': 'string'
        },
        'PlayReadyDrm': {
            'Format': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
        },
    },
]
```



```

        'KeyId': 'string',
        'InitializationVector': 'string',
        'LicenseAcquisitionUrl': 'string'
    },
    'Status': 'string',
    'StatusDetail': 'string'
},
],
'Status': 'string',
'UserMetadata': {
    'string': 'string'
},
'Timing': {
    'SubmitTimeMillis': 123,
    'StartTimeMillis': 123,
    'FinishTimeMillis': 123
}
},
],
'NextToken': 'string'
}

```

Response Structure

- *(dict)* –

The `ListJobsByPipelineResponse` structure.

- **Jobs** (*list*) –

An array of `Job` objects that are in the specified pipeline.

- * *(dict)* –

A section of the response body that provides information about the job that is created.

- **Id** (*string*) –

The identifier that Elastic Transcoder assigned to the job. You use this value to get settings for the job or to delete the job.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) for the job.

- **PipelineId** (*string*) –

The `Id` of the pipeline that you want Elastic Transcoder to use for transcoding. The pipeline determines several settings, including the Amazon S3 bucket from which Elastic Transcoder gets the files to transcode and the bucket into which Elastic Transcoder puts the transcoded files.

- **Input** (*dict*) –

A section of the request or response body that provides information about the file that is being transcoded.

- **Key** (*string*) –

The name of the file to transcode. Elsewhere in the body of the JSON block is the the ID of the pipeline to use for processing the job. The `InputBucket` object in that pipeline tells Elastic Transcoder which Amazon S3 bucket to get the file from.

If the file name includes a prefix, such as

`cooking/lasagna.mpg` , include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

- **FrameRate** (*string*) –

The frame rate of the input file. If you want Elastic Transcoder to automatically detect the frame rate of the input file, specify `auto` . If you want to specify the frame rate for the input file, enter one of the following values:

`10, 15, 23.97, 24, 25, 29.97, 30, 60`

If you specify a value other than `auto` , Elastic Transcoder disables automatic detection of the frame rate.

- **Resolution** (*string*) –

This value must be `auto` , which causes Elastic Transcoder to automatically detect the resolution of the input file.

- **AspectRatio** (*string*) –

The aspect ratio of the input file. If you want Elastic Transcoder to automatically detect the aspect ratio of the input file, specify `auto` . If you want to specify the aspect ratio for the output file, enter one of the following values:

`1:1, 4:3, 3:2, 16:9`

If you specify a value other than `auto` , Elastic Transcoder disables automatic detection of the aspect ratio.

- **Interlaced** (*string*) –

Whether the input file is interlaced. If you want Elastic Transcoder to automatically detect whether the input file is interlaced, specify `auto` . If you want to specify whether the input file is interlaced, enter one of the following values:

`true, false`

If you specify a value other than `auto` , Elastic Transcoder disables automatic detection of interlacing.

- **Container** (*string*) –

The container type for the input file. If you want Elastic Transcoder to automatically detect the container type of the input file, specify `auto` . If you want to specify the container type for the input file, enter one of the following values:

`3gp, aac, asf, avi, divx, flv, m4a, mkv, mov, mp3, mp4, mpeg, mpeg-ps, mpeg-ts, mxf, ogg, vob, wav, webm`

- **Encryption** (*dict*) –

The encryption settings, if any, that are used for decrypting your input files. If your input file is encrypted, you must specify the mode that Elastic Transcoder will use to decrypt your file.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **DetectedProperties** (*dict*) –

The detected properties of the input file.

- **Width** (*integer*) –

The detected width of the input file, in pixels.

- **Height** (*integer*) –

The detected height of the input file, in pixels.

- **FrameRate** (*string*) –

The detected frame rate of the input file, in frames per second.

- **FileSize** (*integer*) –

The detected file size of the input file, in bytes.

- **DurationMillis** (*integer*) –

The detected duration of the input file, in milliseconds.

- **Output** (*dict*) –

If you specified one output for a job, information about that output. If you specified multiple outputs for a job, the Output object lists information about the first output. This duplicates the information that is listed for the first output in the Outputs object.

Warning: Outputs recommended instead.

A section of the request or response body that provides information about the transcoded (target) file.

- **Id** (*string*) –

A sequential counter, starting with 1, that identifies an output among the outputs from the current job. In the Output syntax, this value is always 1.

- **Key** (*string*) –

The name to assign to the transcoded file. Elastic Transcoder saves the file in the Amazon S3 bucket specified by the OutputBucket object in the pipeline that is specified by the pipeline ID.

- **ThumbnailPattern** (*string*) –

Whether you want Elastic Transcoder to create thumbnails for your videos and, if so, how you want Elastic Transcoder to name the files.

If you don't want Elastic Transcoder to create thumbnails, specify "".

If you do want Elastic Transcoder to create thumbnails, specify the information that you want to include in the file name for each thumbnail. You can specify the following values in any sequence:

- **“{count}” (Required)** : If you want to create thumbnails, you must include {count} in the ThumbnailPattern object. Wherever you specify {count}, Elastic Transcoder adds a five-digit sequence number (beginning with **00001**) to thumbnail file names. The number indicates where a given thumbnail appears in the sequence of thumbnails for a transcoded file. .. warning::If you specify a literal value and/or {resolution} but you omit {count}, Elastic

Transcoder returns a validation error and does not create the job.

- **Literal values (Optional)** : You can specify literal values anywhere in the `ThumbnailPattern` object. For example, you can include them as a file name prefix or as a delimiter between `{resolution}` and `{count}`.
- **“{resolution}” (Optional)** : If you want Elastic Transcoder to include the resolution in the file name, include `{resolution}` in the `ThumbnailPattern` object. When creating thumbnails, Elastic Transcoder automatically saves the files in the format (.jpg or .png) that appears in the preset that you specified in the `PresetID` value of `CreateJobOutput`. Elastic Transcoder also appends the applicable file name extension.
- **ThumbnailEncryption (dict)** –

The encryption settings, if any, that you want Elastic Transcoder to apply to your thumbnail.

- **Mode (string)** –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don’t want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won’t be able to unencrypt your data.

- **Key (string)** –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-

encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Rotate** (*string*) –

The number of degrees clockwise by which you want Elastic Transcoder to rotate the output relative to the input. Enter one of the following values:

auto , 0 , 90 , 180 , 270

The value `auto` generally works only if the file that you're transcoding contains rotation metadata.

- **PresetId** (*string*) –

The value of the `Id` object for the preset that you want to use for this job. The preset determines the audio, video, and thumbnail settings that Elastic Transcoder uses for transcoding. To use a preset that you created, specify the preset ID that Elastic Transcoder returned in the response when you created the preset. You can also use the Elastic Transcoder system presets, which you can get with `ListPresets` .

- **SegmentDuration** (*string*) –

Warning: (Outputs in Fragmented MP4 or MPEG-TS format only.

If you specify a preset in `PresetId` for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `SegmentDuration` is the target maximum duration of each segment in seconds. For `HLSv3` format playlists, each media segment is stored in a separate `.ts` file. For `HLSv4` and `Smooth` playlists, all media segments for an output are stored in a single file. Each segment is approximately the length of the `SegmentDuration` , though individual segments might be shorter or longer.

The range of valid values is 1 to 60 seconds. If the duration of the video is not evenly divisible by `SegmentDuration`

, the duration of the last segment is the remainder of total length/SegmentDuration.

Elastic Transcoder creates an output-specific playlist for each output HLS output that you specify in `OutputKeys`. To add an output to the master playlist for this job, include it in the `OutputKeys` of the associated playlist.

- **Status** (*string*) –

The status of one output in a job. If you specified only one output for the job, `Outputs:Status` is always the same as `Job:Status`. If you specified more than one output:

- `Job:Status` and `Outputs:Status` for all of the outputs is `Submitted` until Elastic Transcoder starts to process the first output.
 - When Elastic Transcoder starts to process the first output, `Outputs:Status` for that output and `Job:Status` both change to `Progressing`. For each output, the value of `Outputs:Status` remains `Submitted` until Elastic Transcoder starts to process the output.
 - `Job:Status` remains `Progressing` until all of the outputs reach a terminal status, either `Complete` or `Error`.
 - When all of the outputs reach a terminal status, `Job:Status` changes to `Complete` only if `Outputs:Status` for all of the outputs is `Complete`. If `Outputs:Status` for one or more outputs is `Error`, the terminal status for `Job:Status` is also `Error`.
- The value of `Status` is one of the following: `Submitted`, `Progressing`, `Complete`, `Canceled`, or `Error`.

- **StatusDetail** (*string*) –

Information that further explains `Status`.

- **Duration** (*integer*) –

Duration of the output file, in seconds.

- **Width** (*integer*) –

Specifies the width of the output file in pixels.

- **Height** (*integer*) –

Height of the output file, in pixels.

- **FrameRate** (*string*) –

Frame rate of the output file, in frames per second.

- **FileSize** (*integer*) –

File size of the output file, in bytes.

- **DurationMillis** (*integer*) –

Duration of the output file, in milliseconds.

- **Watermarks** (*list*) –

Information about the watermarks that you want Elastic Transcoder to add to the video during transcoding. You can specify up to four watermarks for each output. Settings for each watermark must be defined in the preset that you specify in `Preset` for the current output.

Watermarks are added to the output video in the sequence in

which you list them in the job output the first watermark in the list is added to the output video first, the second watermark in the list is added next, and so on. As a result, if the settings in a preset cause Elastic Transcoder to place all watermarks in the same location, the second watermark that you add will cover the first one, the third one will cover the second, and the fourth one will cover the third.

- *(dict)* –

Watermarks can be in .png or .jpg format. If you want to display a watermark that is not rectangular, use the .png format, which supports transparency.

- **PresetWatermarkId** (*string*) –

The ID of the watermark settings that Elastic Transcoder uses to add watermarks to the video during transcoding. The settings are in the preset specified by `Preset` for the current output. In that preset, the value of `Watermarks Id` tells Elastic Transcoder which settings to use.

- **InputKey** (*string*) –

The name of the .png or .jpg file that you want to use for the watermark. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `Pipeline`; the `Input Bucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, **logos/128x64.png**, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your watermarks.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**

- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AlbumArt** (*dict*) –

The album art to be associated with the output file, if any.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder will handle the existence of multiple album artwork files.

- **Replace**: The specified album art will replace any existing album art.
- **Prepend**: The specified album art will be placed in front of any existing album art.
- **Append**: The specified album art will be placed after any existing album art.
- **Fallback**: If the original input file contains artwork, Elastic Transcoder will use that artwork for the output. If the original input does not contain artwork, Elastic Transcoder will use the specified album art file.
- **Artwork** (*list*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20. Valid formats are .jpg and .png

- *(dict)* –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20.

To remove artwork or leave the artwork empty, you can either set `Artwork` to `null`, or set the `Merge Policy` to “Replace” and use an empty `Artwork` array.

To pass through existing artwork unchanged, set the `Merge Policy` to “Prepend”, “Append”, or “Fallback”, and use an empty `Artwork` array.

- **InputKey** (*string*) –

The name of the file to be used as album art. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/pie.jpg`, include the prefix in the key. If the file isn’t in the specified bucket, Elastic Transcoder returns an error.

- **MaxWidth** (*string*) –

The maximum width of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

- **MaxHeight** (*string*) –

The maximum height of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.

- **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of the output album art:

- **Fit**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Fill**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output art and then crops it in the dimension (if any) that exceeds the maximum value.
- **Stretch**: Elastic Transcoder stretches the output art to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input art and the output art are different, the output art will be distorted.
- **Keep**: Elastic Transcoder does not scale the output art. If either dimension of the input art exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output art.
- **ShrinkToFit**: Elastic Transcoder scales the output art down so that its dimensions match the values that you spec-

ified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the art up.

- **ShrinkToFill** Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the art up.

- **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder may add white bars to the top and bottom and/or left and right sides of the output album art to make the total size of the output art match the values that you specified for `MaxWidth` and `MaxHeight`.

- **AlbumArtFormat** (*string*) –

The format of album art, if any. Valid formats are `.jpg` and `.png`.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your artwork.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Composition** (*list*) –

You can create an output file that contains an excerpt from the input file. This excerpt, called a clip, can come from the beginning, middle, or end of the file. The Composition object contains settings for the clips that make up an output file. For the current release, you can only specify settings for a single clip per output file. The Composition object cannot be null.

- (*dict*) –

Settings for one clip in a composition. All jobs in a playlist must have the same clip settings.

- **TimeSpan** (*dict*) –

Settings that determine when a clip begins and how long it lasts.

- **StartTime** (*string*) –

The place in the input file where you want a clip to start. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder starts at the beginning of the input file.

- **Duration** (*string*) –

The duration of the clip. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder creates an output file from StartTime to the end of the file.

If you specify a value longer than the duration of the input file, Elastic Transcoder transcodes the file and returns a warning

message.

- **Captions** (*dict*) –

You can configure Elastic Transcoder to transcode captions, or subtitles, from one format to another. All captions must be in UTF-8. Elastic Transcoder supports two types of captions:

- **Embedded:** Embedded captions are included in the same file as the audio and video. Elastic Transcoder supports only one embedded caption per language, to a maximum of 300 embedded captions per file. Valid input values include: `CEA-608` (`EIA-608`, first non-empty channel only), `CEA-708` (`EIA-708`, first non-empty channel only), and `mov-text`. Valid outputs include: `mov-text`. Elastic Transcoder supports a maximum of one embedded format per output.
- **Sidecar:** Sidecar captions are kept in a separate metadata file from the audio and video data. Sidecar captions require a player that is capable of understanding the relationship between the video file and the sidecar file. Elastic Transcoder supports only one sidecar caption per language, to a maximum of 20 sidecar captions per file. Valid input values include: `dfxp` (first div element only), `ebu-tt`, `scc`, `smpt`, `srt`, `ttml` (first div element only), and `webvtt`. Valid outputs include: `dfxp` (first div element only), `scc`, `srt`, and `webvtt`.

If you want `ttml` or `smpte-tt` compatible captions, specify `dfxp` as your output format.

Elastic Transcoder does not support OCR (Optical Character Recognition), does not accept pictures as a valid input for captions, and is not available for audio-only transcoding. Elastic Transcoder does not preserve text formatting (for example, italics) during the transcoding process.

To remove captions or leave the captions empty, set `Captions` to `null`. To pass through existing captions unchanged, set the `MergePolicy` to `MergeRetain`, and pass in a null `CaptionSources` array.

For more information on embedded files, see the Subtitles Wikipedia page.

For more information on sidecar files, see the Extensible Metadata Platform and Sidecar file Wikipedia pages.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder handles the existence of multiple captions.

- **MergeOverride:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the sidecar captions and ignores the embedded captions for that language.
- **MergeRetain:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the embedded captions and ignores

the sidecar captions for that language. If `CaptionSources` is empty, Elastic Transcoder omits all sidecar captions from the output files.

- **Override:** Elastic Transcoder transcodes only the sidecar captions that you specify in `CaptionSources`. `MergePolicy` cannot be null.

- **CaptionSources** (*list*) –

Source files for the input sidecar captions used during the transcoding process. To omit all sidecar captions, leave `CaptionSources` blank.

- (*dict*) –

A source file for the input sidecar captions used during the transcoding process.

- **Key** (*string*) –

The name of the sidecar caption file that you want Elastic Transcoder to include in the output file.

- **Language** (*string*) –

A string that specifies the language of the caption. Specify this as one of:

- 2-character ISO 639-1 code
- 3-character ISO 639-2 code

For more information on ISO language codes and language names, see the List of ISO 639-1 codes.

- **TimeOffset** (*string*) –

For clip generation or captions that do not start at the same time as the associated video file, the `TimeOffset` tells Elastic Transcoder how much of the video to encode before including captions.

Specify the `TimeOffset` in the form `[+/-]SS.sss` or `[+/-]HH:mm:SS.ss`.

- **Label** (*string*) –

The label of the caption shown in the player when choosing a language. We recommend that you put the caption language name here, in the language of the captions.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption sources.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.

- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **CaptionFormats** (*list*) –

The array of file formats for the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- (*dict*) –

The file format of the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- **Format** (*string*) –

The format you specify determines whether Elastic Transcoder generates an embedded or sidecar caption

for this output.

- **Valid Embedded Caption Formats:** * **for FLAC** : None
- **For MP3** : None
- **For MP4** : mov-text
- **For MPEG-TS** : None
- **For ogg** : None
- **For webm** : None
- **Valid Sidecar Caption Formats:** Elastic Transcoder supports dfxp (first div element only), scc, srt, and webvtt. If you want ttml or smpte-tt compatible captions, specify dfxp as your output format. * **For FMP4** : dfxp
- **Non-FMP4 outputs** : All sidecar types
fmp4 captions have an extension of .ismt
- **Pattern** (*string*) –

The prefix for caption filenames, in the form *description* - {language}, where:

- *description* is a description of the video.
- {language} is a literal value that Elastic Transcoder replaces with the two- or three-letter code for the language of the caption in the output file names.

If you don't include {language} in the file name pattern, Elastic Transcoder automatically appends "{language}" to the value that you specify for the description. In addition, Elastic Transcoder automatically appends the count to the end of the segment files.

For example, suppose you're transcoding into srt format. When you enter "Sydney-{language}-sunrise", and the language of the captions is English (en), the name of the first caption file will be Sydney-en-sunrise00000.srt.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption formats.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify S3-AWS-KMS and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following

settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your output files. If you choose to use encryption, you must specify a mode to use. If you choose not to use encryption, Elastic Transcoder will write an unencrypted file to your Amazon S3 bucket.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the

AWS-KMS key that you want to use to your pipeline.

- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AppliedColorSpaceConversion** (*string*) –

If Elastic Transcoder used a preset with a `ColorSpaceConversionMode` to transcode the output file, the `AppliedColorSpaceConversion` parameter shows the conversion used. If no `ColorSpaceConversionMode` was defined in the preset, this parameter will not be included in the job response.

- **Outputs** (*list*) –

Information about the output files. We recommend that you use the `Outputs` syntax for all jobs, even when you want Elastic Transcoder to transcode a file into only one format. Do not use both the `Outputs` and `Output` syntaxes in the same request. You can create a maximum of 30 outputs per job.

If you specify more than one output for a job, Elastic Transcoder creates the files for each output in the order in which you specify them in the job.

· *(dict)* –

Warning: Outputs recommended instead.

If you specified one output for a job, information about that output. If you specified multiple outputs for a job, the `Output` object lists information about the first output. This duplicates the information that is listed for the first output in the `Outputs` object.

· **Id** (*string*) –

A sequential counter, starting with 1, that identifies an output among the outputs from the current job. In the `Output` syntax, this value is always 1.

· **Key** (*string*) –

The name to assign to the transcoded file. Elastic Transcoder saves the file in the Amazon S3 bucket specified by the `OutputBucket` object in the pipeline that is specified by the pipeline ID.

· **ThumbnailPattern** (*string*) –

Whether you want Elastic Transcoder to create thumbnails for your videos and, if so, how you want Elastic Transcoder to name the files.

If you don't want Elastic Transcoder to create thumbnails, specify `""`.

If you do want Elastic Transcoder to create thumbnails, specify the information that you want to include in the file name for each thumbnail. You can specify the following values in any sequence:

- **"{count}" (Required)** : If you want to create thumbnails, you must include `{count}` in the `ThumbnailPattern` object. Wherever you specify `{count}`, Elastic Transcoder adds a five-digit sequence number (beginning with **00001**) to thumbnail file names. The number indicates where a given thumbnail appears in the sequence of thumbnails for a transcoded file. .. warning::If you specify a literal value and/or `{resolution}` but you omit `{count}`, Elastic Transcoder returns a validation error and does not create the job.
- **Literal values (Optional)** : You can specify literal values anywhere in the `ThumbnailPattern` object. For example, you can include them as a file name prefix or as a delimiter

between `{resolution}` and `{count}` .

- “**{resolution}**” (Optional) : If you want Elastic Transcoder to include the resolution in the file name, include `{resolution}` in the `ThumbnailPattern` object.

When creating thumbnails, Elastic Transcoder automatically saves the files in the format (.jpg or .png) that appears in the preset that you specified in the `PresetID` value of `CreateJobOutput` . Elastic Transcoder also appends the applicable file name extension.

- **ThumbnailEncryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your thumbnail.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3**: Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS**: Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don’t want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7**: A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR**: AES Counter Mode.
- **AES-GCM**: AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won’t be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Rotate** (*string*) –

The number of degrees clockwise by which you want Elastic Transcoder to rotate the output relative to the input. Enter one of the following values:

`auto, 0, 90, 180, 270`

The value `auto` generally works only if the file that you're transcoding contains rotation metadata.

- **PresetId** (*string*) –

The value of the `Id` object for the preset that you want to use for this job. The preset determines the audio, video, and thumbnail settings that Elastic Transcoder uses for transcoding. To use a preset that you created, specify the preset ID that Elastic Transcoder returned in the response when you created the preset. You can also use the Elastic Transcoder system presets, which you can get with `ListPresets`.

- **SegmentDuration** (*string*) –

Warning: (Outputs in Fragmented MP4 or MPEG-TS format only.

If you specify a preset in `PresetId` for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `SegmentDuration` is the target maximum duration of each segment in seconds. For `HLSv3` format playlists, each media segment is stored in a separate `.ts` file. For `HLSv4` and `Smooth` playlists, all media segments for an output are stored in a single file. Each segment is approximately the length of the `SegmentDuration`, though individual segments might be shorter or longer.

The range of valid values is 1 to 60 seconds. If the duration of the video is not evenly divisible by `SegmentDuration`, the duration of the last segment is the remainder of `total length / SegmentDuration`.

Elastic Transcoder creates an output-specific playlist for each output `HLS` output that you specify in `OutputKeys`. To add

an output to the master playlist for this job, include it in the `OutputKeys` of the associated playlist.

- **Status** (*string*) –

The status of one output in a job. If you specified only one output for the job, `Outputs:Status` is always the same as `Job:Status`. If you specified more than one output:

- `Job:Status` and `Outputs:Status` for all of the outputs is `Submitted` until Elastic Transcoder starts to process the first output.
- When Elastic Transcoder starts to process the first output, `Outputs:Status` for that output and `Job:Status` both change to `Progressing`. For each output, the value of `Outputs:Status` remains `Submitted` until Elastic Transcoder starts to process the output.
- `Job:Status` remains `Progressing` until all of the outputs reach a terminal status, either `Complete` or `Error`.
- When all of the outputs reach a terminal status, `Job:Status` changes to `Complete` only if `Outputs:Status` for all of the outputs is `Complete`. If `Outputs:Status` for one or more outputs is `Error`, the terminal status for `Job:Status` is also `Error`.

The value of `Status` is one of the following: `Submitted`, `Progressing`, `Complete`, `Canceled`, or `Error`.

- **StatusDetail** (*string*) –

Information that further explains `Status`.

- **Duration** (*integer*) –

Duration of the output file, in seconds.

- **Width** (*integer*) –

Specifies the width of the output file in pixels.

- **Height** (*integer*) –

Height of the output file, in pixels.

- **FrameRate** (*string*) –

Frame rate of the output file, in frames per second.

- **FileSize** (*integer*) –

File size of the output file, in bytes.

- **DurationMillis** (*integer*) –

Duration of the output file, in milliseconds.

- **Watermarks** (*list*) –

Information about the watermarks that you want Elastic Transcoder to add to the video during transcoding. You can specify up to four watermarks for each output. Settings for each watermark must be defined in the preset that you specify in `Preset` for the current output.

Watermarks are added to the output video in the sequence in which you list them in the job output. The first watermark in the list is added to the output video first, the second watermark in the list is added next, and so on. As a result, if the settings in a preset cause Elastic Transcoder to place all watermarks in the same location, the second watermark that you add will

cover the first one, the third one will cover the second, and the fourth one will cover the third.

- **(dict)** –

Watermarks can be in .png or .jpg format. If you want to display a watermark that is not rectangular, use the .png format, which supports transparency.

- **PresetWatermarkId** (*string*) –

The ID of the watermark settings that Elastic Transcoder uses to add watermarks to the video during transcoding. The settings are in the preset specified by Preset for the current output. In that preset, the value of Watermarks Id tells Elastic Transcoder which settings to use.

- **InputKey** (*string*) –

The name of the .png or .jpg file that you want to use for the watermark. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by Pipeline; the Input Bucket object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, **logos/128x64.png**, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your watermarks.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify S3-AWS-KMS and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AlbumArt** (*dict*) –

The album art to be associated with the output file, if any.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder will handle the existence of multiple album artwork files.

- **Replace**: The specified album art will replace any existing album art.
- **Prepend**: The specified album art will be placed in front of any existing album art.
- **Append**: The specified album art will be placed after any existing album art.
- **Fallback**: If the original input file contains artwork, Elastic Transcoder will use that artwork for the output. If the original input does not contain artwork, Elastic Transcoder will use the specified album art file.

- **Artwork** (*list*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20. Valid formats are .jpg and .png

- (*dict*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20.

To remove artwork or leave the artwork empty, you can either set `Artwork` to null, or set the `Merge Policy` to “Replace” and use an empty `Artwork` array.

To pass through existing artwork unchanged, set the `Merge Policy` to “Prepend”, “Append”, or “Fallback”, and use an empty `Artwork` array.

- **InputKey** (*string*) –

The name of the file to be used as album art. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/pie.jpg`, include the prefix in the key. If the file isn’t in the specified bucket, Elastic Transcoder returns an error.

- **MaxWidth** (*string*) –

The maximum width of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

- **MaxHeight** (*string*) –

The maximum height of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.

- **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of the output album art:

- **Fit**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Fill**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output art and then crops it in the dimension (if any) that exceeds the maximum value.
- **Stretch**: Elastic Transcoder stretches the output art to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input art and the output art are different, the output art will be distorted.
- **Keep**: Elastic Transcoder does not scale the output art. If either dimension of the input art exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output art.
- **ShrinkToFit**: Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` with-

out exceeding either value. If you specify this option, Elastic Transcoder does not scale the art up.

- **ShrinkToFill** Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the art up.
- **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder may add white bars to the top and bottom and/or left and right sides of the output album art to make the total size of the output art match the values that you specified for `MaxWidth` and `MaxHeight`.

- **AlbumArtFormat** (*string*) –

The format of album art, if any. Valid formats are `.jpg` and `.png`.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your artwork.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder

to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Composition** (*list*) –

You can create an output file that contains an excerpt from the input file. This excerpt, called a clip, can come from the beginning, middle, or end of the file. The Composition object contains settings for the clips that make up an output file. For the current release, you can only specify settings for a single clip per output file. The Composition object cannot be null.

- (*dict*) –

Settings for one clip in a composition. All jobs in a playlist must have the same clip settings.

- **TimeSpan** (*dict*) –

Settings that determine when a clip begins and how long it lasts.

- **StartTime** (*string*) –

The place in the input file where you want a clip to start. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder starts at the beginning of the input file.

- **Duration** (*string*) –

The duration of the clip. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder creates an output file from StartTime to the end of the file.

If you specify a value longer than the duration of the input file, Elastic Transcoder transcodes the file and returns a warning message.

- **Captions** (*dict*) –

You can configure Elastic Transcoder to transcode captions, or subtitles, from one format to another. All captions must be in UTF-8. Elastic Transcoder supports two types of captions:

- **Embedded:** Embedded captions are included in the same file as the audio and video. Elastic Transcoder supports only one embedded caption per language, to a maximum of 300 embedded captions per file. Valid input values include: `CEA-608` (`EIA-608`, first non-empty channel only), `CEA-708` (`EIA-708`, first non-empty channel only), and `mov-text`. Valid outputs include: `mov-text`. Elastic Transcoder supports a maximum of one embedded format per output.
- **Sidecar:** Sidecar captions are kept in a separate metadata file from the audio and video data. Sidecar captions require a player that is capable of understanding the relationship between the video file and the sidecar file. Elastic Transcoder supports only one sidecar caption per language, to a maximum of 20 sidecar captions per file. Valid input values include: `dfxp` (first div element only), `ebu-tt`, `scc`, `smpt`, `srt`, `ttml` (first div element only), and `webvtt`. Valid outputs include: `dfxp` (first div element only), `scc`, `srt`, and `webvtt`.

If you want `ttml` or `smpte-tt` compatible captions, specify `dfxp` as your output format.

Elastic Transcoder does not support OCR (Optical Character Recognition), does not accept pictures as a valid input for captions, and is not available for audio-only transcoding. Elastic Transcoder does not preserve text formatting (for example, italics) during the transcoding process.

To remove captions or leave the captions empty, set `Captions` to `null`. To pass through existing captions unchanged, set the `MergePolicy` to `MergeRetain`, and pass in a null `CaptionSources` array.

For more information on embedded files, see the Subtitles Wikipedia page.

For more information on sidecar files, see the Extensible Metadata Platform and Sidecar file Wikipedia pages.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder handles the existence of multiple captions.

- **MergeOverride:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the sidecar captions and ignores the embedded captions for that language.
- **MergeRetain:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the embedded captions and ignores the sidecar captions for that language. If `CaptionSources`

is empty, Elastic Transcoder omits all sidecar captions from the output files.

- **Override:** Elastic Transcoder transcodes only the sidecar captions that you specify in `CaptionSources`. `MergePolicy` cannot be null.

- **CaptionSources** (*list*) –

Source files for the input sidecar captions used during the transcoding process. To omit all sidecar captions, leave `CaptionSources` blank.

- (*dict*) –

A source file for the input sidecar captions used during the transcoding process.

- **Key** (*string*) –

The name of the sidecar caption file that you want Elastic Transcoder to include in the output file.

- **Language** (*string*) –

A string that specifies the language of the caption. Specify this as one of:

- 2-character ISO 639-1 code
- 3-character ISO 639-2 code

For more information on ISO language codes and language names, see the List of ISO 639-1 codes.

- **TimeOffset** (*string*) –

For clip generation or captions that do not start at the same time as the associated video file, the `TimeOffset` tells Elastic Transcoder how much of the video to encode before including captions.

Specify the `TimeOffset` in the form `[+-]SS.sss` or `[+-]HH:mm:SS.ss`.

- **Label** (*string*) –

The label of the caption shown in the player when choosing a language. We recommend that you put the caption language name here, in the language of the captions.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption sources.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.

- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **CaptionFormats** (*list*) –

The array of file formats for the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- (*dict*) –

The file format of the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- **Format** (*string*) –

The format you specify determines whether Elastic Transcoder generates an embedded or sidecar caption

for this output.

- **Valid Embedded Caption Formats:** * for **FLAC** : None
- **For MP3** : None
- **For MP4** : mov-text
- **For MPEG-TS** : None
- **For ogg** : None
- **For webm** : None
- **Valid Sidecar Caption Formats:** Elastic Transcoder supports dfxp (first div element only), scc, srt, and webvtt. If you want ttml or smpte-tt compatible captions, specify dfxp as your output format. * **For FMP4** : dfxp
- **Non-FMP4 outputs** : All sidecar types
fmp4 captions have an extension of .ismt
- **Pattern** (*string*) –

The prefix for caption filenames, in the form *description* - {language} , where:

- *description* is a description of the video.
- {language} is a literal value that Elastic Transcoder replaces with the two- or three-letter code for the language of the caption in the output file names.

If you don't include {language} in the file name pattern, Elastic Transcoder automatically appends "{language}" to the value that you specify for the description. In addition, Elastic Transcoder automatically appends the count to the end of the segment files.

For example, suppose you're transcoding into srt format. When you enter "Sydney-{language}-sunrise", and the language of the captions is English (en), the name of the first caption file will be Sydney-en-sunrise00000.srt.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption formats.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify S3-AWS-KMS and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following

settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your output files. If you choose to use encryption, you must specify a mode to use. If you choose not to use encryption, Elastic Transcoder will write an unencrypted file to your Amazon S3 bucket.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the

AWS-KMS key that you want to use to your pipeline.

- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AppliedColorSpaceConversion** (*string*) –

If Elastic Transcoder used a preset with a `ColorSpaceConversionMode` to transcode the output file, the `AppliedColorSpaceConversion` parameter shows the conversion used. If no `ColorSpaceConversionMode` was defined in the preset, this parameter will not be included in the job response.

- **OutputKeyPrefix** (*string*) –

The value, if any, that you want Elastic Transcoder to prepend to the names of all files that this job creates, including output files, thumbnails, and playlists. We recommend that you add a / or some other delimiter to the end of the `OutputKeyPrefix`.

- **Playlists** (*list*) –

Warning: Outputs in Fragmented MP4 or MPEG-TS format only.

If you specify a preset in `PresetId` for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `Playlists` contains information about the master playlists that you want Elastic Transcoder to create.

The maximum number of master playlists in a job is 30.

- (*dict*) –

Use Only for Fragmented MP4 or MPEG-TS Outputs. If you specify a preset for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `Playlists` contains information about the master playlists that you want Elastic Transcoder to create. We recommend that you create only one master playlist per output format. The maximum number of master playlists in a job is 30.

- **Name** (*string*) –

The name that you want Elastic Transcoder to assign to the master playlist, for example, `nyc-vacation.m3u8`. If the name includes a / character, the section of the name before the last / must be identical for all `Name` objects. If you create more than one master playlist, the values of all `Name` objects must be unique.

Note : Elastic Transcoder automatically appends the relevant file extension to the file name (`.m3u8` for `HLSv3` and `HLSv4` playlists, and `.ism` and `.ismc` for `Smooth` playlists). If you include a file extension in `Name`, the file name will have two extensions.

- **Format** (*string*) –

The format of the output playlist. Valid formats include `HLSv3`, `HLSv4`, and `Smooth`.

- **OutputKeys** (*list*) –

For each output in this job that you want to include in a master playlist, the value of the `Outputs:Key` object.

- If your output is not `HLS` or does not have a segment duration set, the name of the output file is a concatenation of `OutputKeyPrefix` and `Outputs:Key:OutputKeyPrefix“Outputs:Key“`
- If your output is `HLSv3` and has a segment duration set, or is not included in a playlist, Elastic Transcoder creates an output playlist file with a file extension of `.m3u8`, and a series of `.ts` files that include a five-digit sequential counter beginning with 00000: `OutputKeyPrefix“Outputs:Key“ .m3u8`
`OutputKeyPrefix“Outputs:Key“ 00000.ts`

- If your output is `HLSv4`, has a segment duration set, and is included in an `HLSv4` playlist, Elastic Transcoder creates an output playlist file with a file extension of `_v4.m3u8`. If the output is video, Elastic Transcoder also creates an output file with an extension of `_iframe.m3u8`: `OutputKeyPrefix“Outputs:Key“_v4.m3u8` `OutputKeyPrefix“Outputs:Key“_iframe.m3u8` `OutputKeyPrefix“Outputs:Key“.ts`

Elastic Transcoder automatically appends the relevant file extension to the file name. If you include a file extension in Output Key, the file name will have two extensions.

If you include more than one output in a playlist, any segment duration settings, clip settings, or caption settings must be the same for all outputs in the playlist. For Smooth playlists, the `Audio:Profile`, `Video:Profile`, and `Video:FrameRate` to `Video:KeyframesMaxDist` ratio must be the same for all outputs.

- *(string)* –
- **HlsContentProtection** (*dict*) –

The HLS content protection settings, if any, that you want Elastic Transcoder to apply to the output files associated with this playlist.

- **Method** (*string*) –

The content protection method for your output. The only valid value is: `aes-128`.

This value will be written into the method attribute of the `EXT-X-KEY` metadata tag in the output playlist.

- **Key** (*string*) –

If you want Elastic Transcoder to generate a key for you, leave this field blank.

If you choose to supply your own key, you must encrypt the key by using AWS KMS. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

`128`, `192`, or `256`.

- **KeyMd5** (*string*) –

If Elastic Transcoder is generating your key for you, you must leave this field blank.

The MD5 digest of the key that you want Elastic Transcoder to use to encrypt your output file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

- **InitializationVector** (*string*) –

If Elastic Transcoder is generating your key for you, you must leave this field blank.

The series of random bits created by a random bit generator,

unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

- **LicenseAcquisitionUrl** (*string*) –

The location of the license key required to decrypt your HLS playlist. The URL must be an absolute path, and is referenced in the URI attribute of the EXT-X-KEY metadata tag in the playlist file.

- **KeyStoragePolicy** (*string*) –

Specify whether you want Elastic Transcoder to write your HLS license key to an Amazon S3 bucket. If you choose `WithVariantPlaylists`, `LicenseAcquisitionUrl` must be left blank and Elastic Transcoder writes your data key into the same bucket as the associated playlist.

- **PlayReadyDrm** (*dict*) –

The DRM settings, if any, that you want Elastic Transcoder to apply to the output files associated with this playlist.

- **Format** (*string*) –

The type of DRM, if any, that you want Elastic Transcoder to apply to the output files associated with this playlist.

- **Key** (*string*) –

The DRM key for your file, provided by your DRM license provider. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using AWS KMS.

- **KeyMd5** (*string*) –

The MD5 digest of the key used for DRM on your file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

- **KeyId** (*string*) –

The ID for your DRM key, so that your DRM license provider knows which key to provide.

The key ID must be provided in big endian, and Elastic Transcoder will convert it to little endian before inserting it into the PlayReady DRM headers. If you are unsure whether your license server provides your key ID in big or little endian, check with your DRM provider.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your files. The initialization vector must be base64-encoded, and it must be exactly 8 bytes

long before being base64-encoded. If no initialization vector is provided, Elastic Transcoder generates one for you.

- **LicenseAcquisitionUrl** (*string*) –

The location of the license key required to play DRM content. The URL must be an absolute path, and is referenced by the PlayReady header. The PlayReady header is referenced in the protection header of the client manifest for Smooth Streaming outputs, and in the EXT-X-DXDRM and EXT-XDXDRMINFO metadata tags for HLS playlist outputs. An example URL looks like this: <https://www.example.com/exampleKey/>

- **Status** (*string*) –

The status of the job with which the playlist is associated.

- **StatusDetail** (*string*) –

Information that further explains the status.

- **Status** (*string*) –

The status of the job: Submitted , Progressing , Complete , Canceled , or Error .

- **UserMetadata** (*dict*) –

User-defined metadata that you want to associate with an Elastic Transcoder job. You specify metadata in key/value pairs, and you can add up to 10 key/value pairs per job. Elastic Transcoder does not guarantee that key/value pairs will be returned in the same order in which you specify them.

Metadata keys and values must use characters from the following list:

- 0–9
- A–Z and a–z
- Space
- The following symbols: _ . : / = + - % @
- (*string*) –
- (*string*) –
- **Timing** (*dict*) –

Details about the timing of a job.

- **SubmitTimeMillis** (*integer*) –

The time the job was submitted to Elastic Transcoder, in epoch milliseconds.

- **StartTimeMillis** (*integer*) –

The time the job began transcoding, in epoch milliseconds.

- **FinishTimeMillis** (*integer*) –

The time the job finished transcoding, in epoch milliseconds.

- **NextToken** (*string*) –

A token to resume pagination.

class ElasticTranscoder.Paginator.**ListJobsByStatus**

```
paginator = client.get_paginator('list_jobs_by_status')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `ElasticTranscoder.Client.list_jobs_by_status()`.

Request Syntax

```
response_iterator = paginator.paginate(
    Status='string',
    Ascending='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **Status** (*string*) – **[REQUIRED]**

To get information about all of the jobs associated with the current AWS account that have a given status, specify the following status: `Submitted`, `Progressing`, `Complete`, `Canceled`, or `Error`.

- **Ascending** (*string*) – To list jobs in chronological order by the date and time that they were submitted, enter `true`. To list jobs in reverse chronological order, enter `false`.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns**Response Syntax**

```
{
  'Jobs': [
    {
      'Id': 'string',
      'Arn': 'string',
      'PipelineId': 'string',
      'Input': {
        'Key': 'string',
        'FrameRate': 'string',
        'Resolution': 'string',
        'AspectRatio': 'string',
        'Interlaced': 'string',
        'Container': 'string',
        'Encryption': {
          'Mode': 'string',
```

```

        'Key': 'string',
        'KeyMd5': 'string',
        'InitializationVector': 'string'
    },
    'DetectedProperties': {
        'Width': 123,
        'Height': 123,
        'FrameRate': 'string',
        'FileSize': 123,
        'DurationMillis': 123
    }
},
'Output': {
    'Id': 'string',
    'Key': 'string',
    'ThumbnailPattern': 'string',
    'ThumbnailEncryption': {
        'Mode': 'string',
        'Key': 'string',
        'KeyMd5': 'string',
        'InitializationVector': 'string'
    },
    'Rotate': 'string',
    'PresetId': 'string',
    'SegmentDuration': 'string',
    'Status': 'string',
    'StatusDetail': 'string',
    'Duration': 123,
    'Width': 123,
    'Height': 123,
    'FrameRate': 'string',
    'FileSize': 123,
    'DurationMillis': 123,
    'Watermarks': [
        {
            'PresetWatermarkId': 'string',
            'InputKey': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        },
    ],
    'AlbumArt': {
        'MergePolicy': 'string',
        'Artwork': [
            {
                'InputKey': 'string',
                'MaxWidth': 'string',
                'MaxHeight': 'string',
                'SizingPolicy': 'string',
                'PaddingPolicy': 'string',
                'AlbumArtFormat': 'string',
                'Encryption': {
                    'Mode': 'string',
                    'Key': 'string',

```

```
        'KeyMd5': 'string',
        'InitializationVector': 'string'
    }
},
]
},
'Composition': [
    {
        'TimeSpan': {
            'StartTime': 'string',
            'Duration': 'string'
        }
    },
],
'Captions': {
    'MergePolicy': 'string',
    'CaptionSources': [
        {
            'Key': 'string',
            'Language': 'string',
            'TimeOffset': 'string',
            'Label': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        }
    ],
    'CaptionFormats': [
        {
            'Format': 'string',
            'Pattern': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        }
    ],
},
'Encryption': {
    'Mode': 'string',
    'Key': 'string',
    'KeyMd5': 'string',
    'InitializationVector': 'string'
},
'AppliedColorSpaceConversion': 'string'
},
'Outputs': [
    {
        'Id': 'string',
        'Key': 'string',
        'ThumbnailPattern': 'string',
        'ThumbnailEncryption': {
            'Mode': 'string',
```



```

        'Key': 'string',
        'KeyMd5': 'string',
        'InitializationVector': 'string'
    },
    'Rotate': 'string',
    'PresetId': 'string',
    'SegmentDuration': 'string',
    'Status': 'string',
    'StatusDetail': 'string',
    'Duration': 123,
    'Width': 123,
    'Height': 123,
    'FrameRate': 'string',
    'FileSize': 123,
    'DurationMillis': 123,
    'Watermarks': [
        {
            'PresetWatermarkId': 'string',
            'InputKey': 'string',
            'Encryption': {
                'Mode': 'string',
                'Key': 'string',
                'KeyMd5': 'string',
                'InitializationVector': 'string'
            }
        },
    ],
    'AlbumArt': {
        'MergePolicy': 'string',
        'Artwork': [
            {
                'InputKey': 'string',
                'MaxWidth': 'string',
                'MaxHeight': 'string',
                'SizingPolicy': 'string',
                'PaddingPolicy': 'string',
                'AlbumArtFormat': 'string',
                'Encryption': {
                    'Mode': 'string',
                    'Key': 'string',
                    'KeyMd5': 'string',
                    'InitializationVector': 'string'
                }
            },
        ],
    },
    'Composition': [
        {
            'TimeSpan': {
                'StartTime': 'string',
                'Duration': 'string'
            }
        },
    ],
    'Captions': {
        'MergePolicy': 'string',
        'CaptionSources': [
            {

```

```
        'Key': 'string',
        'Language': 'string',
        'TimeOffset': 'string',
        'Label': 'string',
        'Encryption': {
            'Mode': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'InitializationVector': 'string'
        }
    },
],
'CaptionFormats': [
    {
        'Format': 'string',
        'Pattern': 'string',
        'Encryption': {
            'Mode': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'InitializationVector': 'string'
        }
    },
]
},
'Encryption': {
    'Mode': 'string',
    'Key': 'string',
    'KeyMd5': 'string',
    'InitializationVector': 'string'
},
'AppliedColorSpaceConversion': 'string'
},
],
'OutputKeyPrefix': 'string',
'Playlists': [
    {
        'Name': 'string',
        'Format': 'string',
        'OutputKeys': [
            'string',
        ],
        'HlsContentProtection': {
            'Method': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'InitializationVector': 'string',
            'LicenseAcquisitionUrl': 'string',
            'KeyStoragePolicy': 'string'
        },
        'PlayReadyDrm': {
            'Format': 'string',
            'Key': 'string',
            'KeyMd5': 'string',
            'KeyId': 'string',
            'InitializationVector': 'string',
            'LicenseAcquisitionUrl': 'string'
        }
    },
],
```

```

        'Status': 'string',
        'StatusDetail': 'string'
    },
    ],
    'Status': 'string',
    'UserMetadata': {
        'string': 'string'
    },
    'Timing': {
        'SubmitTimeMillis': 123,
        'StartTimeMillis': 123,
        'FinishTimeMillis': 123
    }
},
],
'NextToken': 'string'
}

```

Response Structure

- (*dict*) –

The `ListJobsByStatusResponse` structure.

- **Jobs** (*list*) –

An array of Job objects that have the specified status.

- * (*dict*) –

A section of the response body that provides information about the job that is created.

- **Id** (*string*) –

The identifier that Elastic Transcoder assigned to the job. You use this value to get settings for the job or to delete the job.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) for the job.

- **PipelineId** (*string*) –

The `Id` of the pipeline that you want Elastic Transcoder to use for transcoding. The pipeline determines several settings, including the Amazon S3 bucket from which Elastic Transcoder gets the files to transcode and the bucket into which Elastic Transcoder puts the transcoded files.

- **Input** (*dict*) –

A section of the request or response body that provides information about the file that is being transcoded.

- **Key** (*string*) –

The name of the file to transcode. Elsewhere in the body of the JSON block is the the ID of the pipeline to use for processing the job. The `InputBucket` object in that pipeline tells Elastic Transcoder which Amazon S3 bucket to get the file from.

If the file name includes a prefix, such as `cooking/lasagna.mpg`, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

- **FrameRate** (*string*) –

The frame rate of the input file. If you want Elastic Transcoder to automatically detect the frame rate of the input file, specify `auto`. If you want to specify the frame rate for the input file, enter one of the following values:

`10, 15, 23.97, 24, 25, 29.97, 30, 60`

If you specify a value other than `auto`, Elastic Transcoder disables automatic detection of the frame rate.

- **Resolution** (*string*) –

This value must be `auto`, which causes Elastic Transcoder to automatically detect the resolution of the input file.

- **AspectRatio** (*string*) –

The aspect ratio of the input file. If you want Elastic Transcoder to automatically detect the aspect ratio of the input file, specify `auto`. If you want to specify the aspect ratio for the output file, enter one of the following values:

`1:1, 4:3, 3:2, 16:9`

If you specify a value other than `auto`, Elastic Transcoder disables automatic detection of the aspect ratio.

- **Interlaced** (*string*) –

Whether the input file is interlaced. If you want Elastic Transcoder to automatically detect whether the input file is interlaced, specify `auto`. If you want to specify whether the input file is interlaced, enter one of the following values:

`true, false`

If you specify a value other than `auto`, Elastic Transcoder disables automatic detection of interlacing.

- **Container** (*string*) –

The container type for the input file. If you want Elastic Transcoder to automatically detect the container type of the input file, specify `auto`. If you want to specify the container type for the input file, enter one of the following values:

`3gp, aac, asf, avi, divx, flv, m4a, mkv, mov, mp3, mp4, mpeg, mpeg-ps, mpeg-ts, mxf, ogg, vob, wav, webm`

- **Encryption** (*dict*) –

The encryption settings, if any, that are used for decrypting your input files. If your input file is encrypted, you must specify the mode that Elastic Transcoder will use to decrypt your file.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are

used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.

- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **DetectedProperties** (*dict*) –

The detected properties of the input file.

- **Width** (*integer*) –

The detected width of the input file, in pixels.

- **Height** (*integer*) –

The detected height of the input file, in pixels.

- **FrameRate** (*string*) –

The detected frame rate of the input file, in frames per second.

- **FileSize** (*integer*) –

The detected file size of the input file, in bytes.

- **DurationMillis** (*integer*) –

The detected duration of the input file, in milliseconds.

- **Output** (*dict*) –

If you specified one output for a job, information about that output. If you specified multiple outputs for a job, the Output object lists information about the first output. This duplicates the information that is listed for the first output in the Outputs object.

Warning: Outputs recommended instead.

A section of the request or response body that provides information about the transcoded (target) file.

- **Id** (*string*) –

A sequential counter, starting with 1, that identifies an output among the outputs from the current job. In the Output syntax, this value is always 1.

- **Key** (*string*) –

The name to assign to the transcoded file. Elastic Transcoder saves the file in the Amazon S3 bucket specified by the `OutputBucket` object in the pipeline that is specified by the pipeline ID.

- **ThumbnailPattern** (*string*) –

Whether you want Elastic Transcoder to create thumbnails for your videos and, if so, how you want Elastic Transcoder to name the files.

If you don't want Elastic Transcoder to create thumbnails, specify "".

If you do want Elastic Transcoder to create thumbnails, specify the information that you want to include in the file name for each thumbnail. You can specify the following values in any sequence:

- **“{count}” (Required)** : If you want to create thumbnails, you must include {count} in the ThumbnailPattern object. Wherever you specify {count}, Elastic Transcoder adds a five-digit sequence number (beginning with **00001**) to thumbnail file names. The number indicates where a given thumbnail appears in the sequence of thumbnails for a transcoded file. .. warning::If you specify a literal value and/or {resolution} but you omit {count}, Elastic Transcoder returns a validation error and does not create the job.

- **Literal values (Optional)** : You can specify literal values anywhere in the ThumbnailPattern object. For example,

you can include them as a file name prefix or as a delimiter between `{resolution}` and `{count}`.

- “**{resolution}**” (Optional) : If you want Elastic Transcoder to include the resolution in the file name, include `{resolution}` in the `ThumbnailPattern` object.

When creating thumbnails, Elastic Transcoder automatically saves the files in the format (.jpg or .png) that appears in the preset that you specified in the `PresetID` value of `CreateJobOutput`. Elastic Transcoder also appends the applicable file name extension.

- **ThumbnailEncryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your thumbnail.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3**: Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS**: Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don’t want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7**: A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR**: AES Counter Mode.
- **AES-GCM**: AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won’t be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128, 192, or 256.

The key must also be encrypted by using the Amazon Key

Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Rotate** (*string*) –

The number of degrees clockwise by which you want Elastic Transcoder to rotate the output relative to the input. Enter one of the following values:

`auto, 0, 90, 180, 270`

The value `auto` generally works only if the file that you're transcoding contains rotation metadata.

- **PresetId** (*string*) –

The value of the `Id` object for the preset that you want to use for this job. The preset determines the audio, video, and thumbnail settings that Elastic Transcoder uses for transcoding. To use a preset that you created, specify the preset ID that Elastic Transcoder returned in the response when you created the preset. You can also use the Elastic Transcoder system presets, which you can get with `ListPresets`.

- **SegmentDuration** (*string*) –

Warning: (Outputs in Fragmented MP4 or MPEG-TS format only.

If you specify a preset in `PresetId` for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `SegmentDuration` is the target maximum duration of each segment in seconds. For `HLSv3` format playlists, each media segment is stored in a separate `.ts` file. For `HLSv4` and `Smooth` playlists, all media segments for an output are stored in a single file. Each segment is approximately the length of the `SegmentDuration`, though individual segments might be shorter or longer.

The range of valid values is 1 to 60 seconds. If the duration of the video is not evenly divisible by `SegmentDuration`, the duration of the last segment is the remainder of total length/`SegmentDuration`.

Elastic Transcoder creates an output-specific playlist for each output `HLS` output that you specify in `OutputKeys`. To add

an output to the master playlist for this job, include it in the `OutputKeys` of the associated playlist.

- **Status** (*string*) –

The status of one output in a job. If you specified only one output for the job, `Outputs:Status` is always the same as `Job:Status`. If you specified more than one output:

- `Job:Status` and `Outputs:Status` for all of the outputs is `Submitted` until Elastic Transcoder starts to process the first output.
- When Elastic Transcoder starts to process the first output, `Outputs:Status` for that output and `Job:Status` both change to `Progressing`. For each output, the value of `Outputs:Status` remains `Submitted` until Elastic Transcoder starts to process the output.
- `Job:Status` remains `Progressing` until all of the outputs reach a terminal status, either `Complete` or `Error`.
- When all of the outputs reach a terminal status, `Job:Status` changes to `Complete` only if `Outputs:Status` for all of the outputs is `Complete`. If `Outputs:Status` for one or more outputs is `Error`, the terminal status for `Job:Status` is also `Error`.

The value of `Status` is one of the following: `Submitted`, `Progressing`, `Complete`, `Canceled`, or `Error`.

- **StatusDetail** (*string*) –

Information that further explains `Status`.

- **Duration** (*integer*) –

Duration of the output file, in seconds.

- **Width** (*integer*) –

Specifies the width of the output file in pixels.

- **Height** (*integer*) –

Height of the output file, in pixels.

- **FrameRate** (*string*) –

Frame rate of the output file, in frames per second.

- **FileSize** (*integer*) –

File size of the output file, in bytes.

- **DurationMillis** (*integer*) –

Duration of the output file, in milliseconds.

- **Watermarks** (*list*) –

Information about the watermarks that you want Elastic Transcoder to add to the video during transcoding. You can specify up to four watermarks for each output. Settings for each watermark must be defined in the preset that you specify in `Preset` for the current output.

Watermarks are added to the output video in the sequence in which you list them in the job output. The first watermark in the list is added to the output video first, the second watermark in the list is added next, and so on. As a result, if the settings in a preset cause Elastic Transcoder to place all watermarks in the same location, the second watermark that you add will

cover the first one, the third one will cover the second, and the fourth one will cover the third.

- *(dict)* –

Watermarks can be in .png or .jpg format. If you want to display a watermark that is not rectangular, use the .png format, which supports transparency.

- **PresetWatermarkId** (*string*) –

The ID of the watermark settings that Elastic Transcoder uses to add watermarks to the video during transcoding. The settings are in the preset specified by `Preset` for the current output. In that preset, the value of `Watermarks Id` tells Elastic Transcoder which settings to use.

- **InputKey** (*string*) –

The name of the .png or .jpg file that you want to use for the watermark. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `Pipeline`; the `Input Bucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, **logos/128x64.png**, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your watermarks.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

· **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

· **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

· **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

· **AlbumArt** (*dict*) –

The album art to be associated with the output file, if any.

· **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder will handle the existence of multiple album artwork files.

- **Replace**: The specified album art will replace any existing album art.
- **Prepend**: The specified album art will be placed in front of any existing album art.
- **Append**: The specified album art will be placed after any existing album art.
- **Fallback**: If the original input file contains artwork, Elastic Transcoder will use that artwork for the output. If the original input does not contain artwork, Elastic Transcoder will use the specified album art file.

· **Artwork** (*list*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20. Valid formats are .jpg and .png

· (*dict*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20.

To remove artwork or leave the artwork empty, you can either set `Artwork` to null, or set the `Merge Policy` to “Replace” and use an empty `Artwork` array.

To pass through existing artwork unchanged, set the `Merge Policy` to “Prepend”, “Append”, or “Fallback”, and use an empty `Artwork` array.

- **InputKey** (*string*) –

The name of the file to be used as album art. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/pie.jpg`, include the prefix in the key. If the file isn’t in the specified bucket, Elastic Transcoder returns an error.

- **MaxWidth** (*string*) –

The maximum width of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

- **MaxHeight** (*string*) –

The maximum height of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.

- **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of the output album art:

- **Fit**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Fill**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output art and then crops it in the dimension (if any) that exceeds the maximum value.
- **Stretch**: Elastic Transcoder stretches the output art to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input art and the output art are different, the output art will be distorted.
- **Keep**: Elastic Transcoder does not scale the output art. If either dimension of the input art exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output art.
- **ShrinkToFit**: Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` with-

out exceeding either value. If you specify this option, Elastic Transcoder does not scale the art up.

- **ShrinkToFill** Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the art up.
- **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder may add white bars to the top and bottom and/or left and right sides of the output album art to make the total size of the output art match the values that you specified for `MaxWidth` and `MaxHeight`.

- **AlbumArtFormat** (*string*) –

The format of album art, if any. Valid formats are `.jpg` and `.png`.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your artwork.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder

to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Composition** (*list*) –

You can create an output file that contains an excerpt from the input file. This excerpt, called a clip, can come from the beginning, middle, or end of the file. The Composition object contains settings for the clips that make up an output file. For the current release, you can only specify settings for a single clip per output file. The Composition object cannot be null.

- (*dict*) –

Settings for one clip in a composition. All jobs in a playlist must have the same clip settings.

- **TimeSpan** (*dict*) –

Settings that determine when a clip begins and how long it lasts.

- **StartTime** (*string*) –

The place in the input file where you want a clip to start. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder starts at the beginning of the input file.

- **Duration** (*string*) –

The duration of the clip. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder creates an output file from StartTime to the end of the file.

If you specify a value longer than the duration of the input file, Elastic Transcoder transcodes the file and returns a warning message.

- **Captions** (*dict*) –

You can configure Elastic Transcoder to transcode captions, or subtitles, from one format to another. All captions must be in UTF-8. Elastic Transcoder supports two types of captions:

- **Embedded:** Embedded captions are included in the same file as the audio and video. Elastic Transcoder supports only one embedded caption per language, to a maximum of 300 embedded captions per file. Valid input values include: `CEA-608` (`EIA-608` , first non-empty channel only), `CEA-708` (`EIA-708` , first non-empty channel only), and `mov-text`. Valid outputs include: `mov-text`. Elastic Transcoder supports a maximum of one embedded format per output.
- **Sidecar:** Sidecar captions are kept in a separate metadata file from the audio and video data. Sidecar captions require a player that is capable of understanding the relationship between the video file and the sidecar file. Elastic Transcoder supports only one sidecar caption per language, to a maximum of 20 sidecar captions per file. Valid input values include: `dfxp` (first div element only), `ebu-tt` , `scc` , `smpt` , `srt` , `ttml` (first div element only), and `webvtt`. Valid outputs include: `dfxp` (first div element only), `scc` , `srt` , and `webvtt`.

If you want `ttml` or `smpte-tt` compatible captions, specify `dfxp` as your output format.

Elastic Transcoder does not support OCR (Optical Character Recognition), does not accept pictures as a valid input for captions, and is not available for audio-only transcoding. Elastic Transcoder does not preserve text formatting (for example, italics) during the transcoding process.

To remove captions or leave the captions empty, set `Captions` to `null`. To pass through existing captions unchanged, set the `MergePolicy` to `MergeRetain` , and pass in a null `CaptionSources` array.

For more information on embedded files, see the Subtitles Wikipedia page.

For more information on sidecar files, see the Extensible Metadata Platform and Sidecar file Wikipedia pages.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder handles the existence of multiple captions.

- **MergeOverride:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the sidecar captions and ignores the embedded captions for that language.
- **MergeRetain:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the embedded captions and ignores the sidecar captions for that language. If `CaptionSources`

is empty, Elastic Transcoder omits all sidecar captions from the output files.

- **Override:** Elastic Transcoder transcodes only the sidecar captions that you specify in `CaptionSources`. `MergePolicy` cannot be null.

- **CaptionSources** (*list*) –

Source files for the input sidecar captions used during the transcoding process. To omit all sidecar captions, leave `CaptionSources` blank.

- (*dict*) –

A source file for the input sidecar captions used during the transcoding process.

- **Key** (*string*) –

The name of the sidecar caption file that you want Elastic Transcoder to include in the output file.

- **Language** (*string*) –

A string that specifies the language of the caption. Specify this as one of:

- 2-character ISO 639-1 code
- 3-character ISO 639-2 code

For more information on ISO language codes and language names, see the List of ISO 639-1 codes.

- **TimeOffset** (*string*) –

For clip generation or captions that do not start at the same time as the associated video file, the `TimeOffset` tells Elastic Transcoder how much of the video to encode before including captions.

Specify the `TimeOffset` in the form `[+-]SS.sss` or `[+-]HH:mm:SS.ss`.

- **Label** (*string*) –

The label of the caption shown in the player when choosing a language. We recommend that you put the caption language name here, in the language of the captions.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption sources.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.

- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **CaptionFormats** (*list*) –

The array of file formats for the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- (*dict*) –

The file format of the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- **Format** (*string*) –

The format you specify determines whether Elastic Transcoder generates an embedded or sidecar caption

for this output.

- **Valid Embedded Caption Formats:** * **for FLAC** : None
- **For MP3** : None
- **For MP4** : mov-text
- **For MPEG-TS** : None
- **For ogg** : None
- **For webm** : None
- **Valid Sidecar Caption Formats:** Elastic Transcoder supports dfxp (first div element only), scc, srt, and webvtt. If you want ttml or smpte-tt compatible captions, specify dfxp as your output format. * **For FMP4** : dfxp
- **Non-FMP4 outputs** : All sidecar types
fmp4 captions have an extension of .ismt
- **Pattern** (*string*) –

The prefix for caption filenames, in the form *description* - {language}, where:

- *description* is a description of the video.
- {language} is a literal value that Elastic Transcoder replaces with the two- or three-letter code for the language of the caption in the output file names.

If you don't include {language} in the file name pattern, Elastic Transcoder automatically appends "{language}" to the value that you specify for the description. In addition, Elastic Transcoder automatically appends the count to the end of the segment files.

For example, suppose you're transcoding into srt format. When you enter "Sydney-{language}-sunrise", and the language of the captions is English (en), the name of the first caption file will be Sydney-en-sunrise00000.srt.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption formats.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify S3-AWS-KMS and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following

settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your output files. If you choose to use encryption, you must specify a mode to use. If you choose not to use encryption, Elastic Transcoder will write an unencrypted file to your Amazon S3 bucket.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the

AWS-KMS key that you want to use to your pipeline.

- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AppliedColorSpaceConversion** (*string*) –

If Elastic Transcoder used a preset with a `ColorSpaceConversionMode` to transcode the output file, the `AppliedColorSpaceConversion` parameter shows the conversion used. If no `ColorSpaceConversionMode` was defined in the preset, this parameter will not be included in the job response.

- **Outputs** (*list*) –

Information about the output files. We recommend that you use the `Outputs` syntax for all jobs, even when you want Elastic Transcoder to transcode a file into only one format. Do not use both the `Outputs` and `Output` syntaxes in the same request. You can create a maximum of 30 outputs per job.

If you specify more than one output for a job, Elastic Transcoder creates the files for each output in the order in which you specify them in the job.

· *(dict)* –

Warning: Outputs recommended instead.

If you specified one output for a job, information about that output. If you specified multiple outputs for a job, the `Output` object lists information about the first output. This duplicates the information that is listed for the first output in the `Outputs` object.

· **Id** (*string*) –

A sequential counter, starting with 1, that identifies an output among the outputs from the current job. In the `Output` syntax, this value is always 1.

· **Key** (*string*) –

The name to assign to the transcoded file. Elastic Transcoder saves the file in the Amazon S3 bucket specified by the `OutputBucket` object in the pipeline that is specified by the pipeline ID.

· **ThumbnailPattern** (*string*) –

Whether you want Elastic Transcoder to create thumbnails for your videos and, if so, how you want Elastic Transcoder to name the files.

If you don't want Elastic Transcoder to create thumbnails, specify `""`.

If you do want Elastic Transcoder to create thumbnails, specify the information that you want to include in the file name for each thumbnail. You can specify the following values in any sequence:

- **"{count}" (Required)** : If you want to create thumbnails, you must include `{count}` in the `ThumbnailPattern` object. Wherever you specify `{count}`, Elastic Transcoder adds a five-digit sequence number (beginning with **00001**) to thumbnail file names. The number indicates where a given thumbnail appears in the sequence of thumbnails for a transcoded file. .. warning::If you specify a literal value and/or `{resolution}` but you omit `{count}`, Elastic Transcoder returns a validation error and does not create the job.
- **Literal values (Optional)** : You can specify literal values anywhere in the `ThumbnailPattern` object. For example, you can include them as a file name prefix or as a delimiter

between `{resolution}` and `{count}` .

- “**{resolution}**” (Optional) : If you want Elastic Transcoder to include the resolution in the file name, include `{resolution}` in the `ThumbnailPattern` object.

When creating thumbnails, Elastic Transcoder automatically saves the files in the format (.jpg or .png) that appears in the preset that you specified in the `PresetID` value of `CreateJobOutput` . Elastic Transcoder also appends the applicable file name extension.

- **ThumbnailEncryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your thumbnail.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3**: Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS**: Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don’t want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7**: A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR**: AES Counter Mode.
- **AES-GCM**: AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won’t be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Rotate** (*string*) –

The number of degrees clockwise by which you want Elastic Transcoder to rotate the output relative to the input. Enter one of the following values:

```
auto, 0, 90, 180, 270
```

The value `auto` generally works only if the file that you're transcoding contains rotation metadata.

- **PresetId** (*string*) –

The value of the `Id` object for the preset that you want to use for this job. The preset determines the audio, video, and thumbnail settings that Elastic Transcoder uses for transcoding. To use a preset that you created, specify the preset ID that Elastic Transcoder returned in the response when you created the preset. You can also use the Elastic Transcoder system presets, which you can get with `ListPresets`.

- **SegmentDuration** (*string*) –

Warning: (Outputs in Fragmented MP4 or MPEG-TS format only.

If you specify a preset in `PresetId` for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `SegmentDuration` is the target maximum duration of each segment in seconds. For `HLSv3` format playlists, each media segment is stored in a separate `.ts` file. For `HLSv4` and `Smooth` playlists, all media segments for an output are stored in a single file. Each segment is approximately the length of the `SegmentDuration`, though individual segments might be shorter or longer.

The range of valid values is 1 to 60 seconds. If the duration of the video is not evenly divisible by `SegmentDuration`, the duration of the last segment is the remainder of total length/`SegmentDuration`.

Elastic Transcoder creates an output-specific playlist for each output `HLS` output that you specify in `OutputKeys`. To add

an output to the master playlist for this job, include it in the `OutputKeys` of the associated playlist.

- **Status** (*string*) –

The status of one output in a job. If you specified only one output for the job, `Outputs:Status` is always the same as `Job:Status`. If you specified more than one output:

- `Job:Status` and `Outputs:Status` for all of the outputs is `Submitted` until Elastic Transcoder starts to process the first output.
- When Elastic Transcoder starts to process the first output, `Outputs:Status` for that output and `Job:Status` both change to `Progressing`. For each output, the value of `Outputs:Status` remains `Submitted` until Elastic Transcoder starts to process the output.
- `Job:Status` remains `Progressing` until all of the outputs reach a terminal status, either `Complete` or `Error`.
- When all of the outputs reach a terminal status, `Job:Status` changes to `Complete` only if `Outputs:Status` for all of the outputs is `Complete`. If `Outputs:Status` for one or more outputs is `Error`, the terminal status for `Job:Status` is also `Error`.

The value of `Status` is one of the following: `Submitted`, `Progressing`, `Complete`, `Canceled`, or `Error`.

- **StatusDetail** (*string*) –

Information that further explains `Status`.

- **Duration** (*integer*) –

Duration of the output file, in seconds.

- **Width** (*integer*) –

Specifies the width of the output file in pixels.

- **Height** (*integer*) –

Height of the output file, in pixels.

- **FrameRate** (*string*) –

Frame rate of the output file, in frames per second.

- **FileSize** (*integer*) –

File size of the output file, in bytes.

- **DurationMillis** (*integer*) –

Duration of the output file, in milliseconds.

- **Watermarks** (*list*) –

Information about the watermarks that you want Elastic Transcoder to add to the video during transcoding. You can specify up to four watermarks for each output. Settings for each watermark must be defined in the preset that you specify in `Preset` for the current output.

Watermarks are added to the output video in the sequence in which you list them in the job output. The first watermark in the list is added to the output video first, the second watermark in the list is added next, and so on. As a result, if the settings in a preset cause Elastic Transcoder to place all watermarks in the same location, the second watermark that you add will

cover the first one, the third one will cover the second, and the fourth one will cover the third.

- **(dict)** –

Watermarks can be in .png or .jpg format. If you want to display a watermark that is not rectangular, use the .png format, which supports transparency.

- **PresetWatermarkId** (*string*) –

The ID of the watermark settings that Elastic Transcoder uses to add watermarks to the video during transcoding. The settings are in the preset specified by Preset for the current output. In that preset, the value of Watermarks Id tells Elastic Transcoder which settings to use.

- **InputKey** (*string*) –

The name of the .png or .jpg file that you want to use for the watermark. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by Pipeline; the Input Bucket object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, **logos/128x64.png**, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your watermarks.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify S3-AWS-KMS and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AlbumArt** (*dict*) –

The album art to be associated with the output file, if any.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder will handle the existence of multiple album artwork files.

- **Replace**: The specified album art will replace any existing album art.
- **Prepend**: The specified album art will be placed in front of any existing album art.
- **Append**: The specified album art will be placed after any existing album art.
- **Fallback**: If the original input file contains artwork, Elastic Transcoder will use that artwork for the output. If the original input does not contain artwork, Elastic Transcoder will use the specified album art file.

- **Artwork** (*list*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20. Valid formats are .jpg and .png

- (*dict*) –

The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20.

To remove artwork or leave the artwork empty, you can either set `Artwork` to null, or set the `Merge Policy` to “Replace” and use an empty `Artwork` array.

To pass through existing artwork unchanged, set the `Merge Policy` to “Prepend”, “Append”, or “Fallback”, and use an empty `Artwork` array.

- **InputKey** (*string*) –

The name of the file to be used as album art. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/pie.jpg`, include the prefix in the key. If the file isn’t in the specified bucket, Elastic Transcoder returns an error.

- **MaxWidth** (*string*) –

The maximum width of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

- **MaxHeight** (*string*) –

The maximum height of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.

- **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of the output album art:

- **Fit**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Fill**: Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output art and then crops it in the dimension (if any) that exceeds the maximum value.
- **Stretch**: Elastic Transcoder stretches the output art to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input art and the output art are different, the output art will be distorted.
- **Keep**: Elastic Transcoder does not scale the output art. If either dimension of the input art exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output art.
- **ShrinkToFit**: Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` with-

out exceeding either value. If you specify this option, Elastic Transcoder does not scale the art up.

- **ShrinkToFill** Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the art up.
- **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder may add white bars to the top and bottom and/or left and right sides of the output album art to make the total size of the output art match the values that you specified for `MaxWidth` and `MaxHeight`.

- **AlbumArtFormat** (*string*) –

The format of album art, if any. Valid formats are `.jpg` and `.png`.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your artwork.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder

to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Composition** (*list*) –

You can create an output file that contains an excerpt from the input file. This excerpt, called a clip, can come from the beginning, middle, or end of the file. The Composition object contains settings for the clips that make up an output file. For the current release, you can only specify settings for a single clip per output file. The Composition object cannot be null.

- (*dict*) –

Settings for one clip in a composition. All jobs in a playlist must have the same clip settings.

- **TimeSpan** (*dict*) –

Settings that determine when a clip begins and how long it lasts.

- **StartTime** (*string*) –

The place in the input file where you want a clip to start. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder starts at the beginning of the input file.

- **Duration** (*string*) –

The duration of the clip. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder creates an output file from StartTime to the end of the file.

If you specify a value longer than the duration of the input file, Elastic Transcoder transcodes the file and returns a warning message.

- **Captions** (*dict*) –

You can configure Elastic Transcoder to transcode captions, or subtitles, from one format to another. All captions must be in UTF-8. Elastic Transcoder supports two types of captions:

- **Embedded:** Embedded captions are included in the same file as the audio and video. Elastic Transcoder supports only one embedded caption per language, to a maximum of 300 embedded captions per file. Valid input values include: `CEA-608` (`EIA-608`, first non-empty channel only), `CEA-708` (`EIA-708`, first non-empty channel only), and `mov-text`. Valid outputs include: `mov-text`. Elastic Transcoder supports a maximum of one embedded format per output.
- **Sidecar:** Sidecar captions are kept in a separate metadata file from the audio and video data. Sidecar captions require a player that is capable of understanding the relationship between the video file and the sidecar file. Elastic Transcoder supports only one sidecar caption per language, to a maximum of 20 sidecar captions per file. Valid input values include: `dfxp` (first div element only), `ebu-tt`, `scc`, `smpt`, `srt`, `ttml` (first div element only), and `webvtt`. Valid outputs include: `dfxp` (first div element only), `scc`, `srt`, and `webvtt`.

If you want `ttml` or `smpte-tt` compatible captions, specify `dfxp` as your output format.

Elastic Transcoder does not support OCR (Optical Character Recognition), does not accept pictures as a valid input for captions, and is not available for audio-only transcoding. Elastic Transcoder does not preserve text formatting (for example, italics) during the transcoding process.

To remove captions or leave the captions empty, set `Captions` to `null`. To pass through existing captions unchanged, set the `MergePolicy` to `MergeRetain`, and pass in a null `CaptionSources` array.

For more information on embedded files, see the Subtitles Wikipedia page.

For more information on sidecar files, see the Extensible Metadata Platform and Sidecar file Wikipedia pages.

- **MergePolicy** (*string*) –

A policy that determines how Elastic Transcoder handles the existence of multiple captions.

- **MergeOverride:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the sidecar captions and ignores the embedded captions for that language.
- **MergeRetain:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the embedded captions and ignores the sidecar captions for that language. If `CaptionSources`

is empty, Elastic Transcoder omits all sidecar captions from the output files.

- **Override:** Elastic Transcoder transcodes only the sidecar captions that you specify in `CaptionSources`. `MergePolicy` cannot be null.

- **CaptionSources** (*list*) –

Source files for the input sidecar captions used during the transcoding process. To omit all sidecar captions, leave `CaptionSources` blank.

- (*dict*) –

A source file for the input sidecar captions used during the transcoding process.

- **Key** (*string*) –

The name of the sidecar caption file that you want Elastic Transcoder to include in the output file.

- **Language** (*string*) –

A string that specifies the language of the caption. Specify this as one of:

- 2-character ISO 639-1 code
- 3-character ISO 639-2 code

For more information on ISO language codes and language names, see the List of ISO 639-1 codes.

- **TimeOffset** (*string*) –

For clip generation or captions that do not start at the same time as the associated video file, the `TimeOffset` tells Elastic Transcoder how much of the video to encode before including captions.

Specify the `TimeOffset` in the form `[+-]SS.sss` or `[+-]HH:mm:SS.ss`.

- **Label** (*string*) –

The label of the caption shown in the player when choosing a language. We recommend that you put the caption language name here, in the language of the captions.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption sources.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.

- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **CaptionFormats** (*list*) –

The array of file formats for the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- (*dict*) –

The file format of the output captions. If you leave this value blank, Elastic Transcoder returns an error.

- **Format** (*string*) –

The format you specify determines whether Elastic Transcoder generates an embedded or sidecar caption

for this output.

- **Valid Embedded Caption Formats:** * for **FLAC** : None
- **For MP3** : None
- **For MP4** : mov-text
- **For MPEG-TS** : None
- **For ogg** : None
- **For webm** : None
- **Valid Sidecar Caption Formats:** Elastic Transcoder supports dfxp (first div element only), scc, srt, and webvtt. If you want ttml or smpte-tt compatible captions, specify dfxp as your output format. * **For FMP4** : dfxp
- **Non-FMP4 outputs** : All sidecar types
fmp4 captions have an extension of .ismt
- **Pattern** (*string*) –

The prefix for caption filenames, in the form *description* - {language} , where:

- *description* is a description of the video.
- {language} is a literal value that Elastic Transcoder replaces with the two- or three-letter code for the language of the caption in the output file names.

If you don't include {language} in the file name pattern, Elastic Transcoder automatically appends "{language}" to the value that you specify for the description. In addition, Elastic Transcoder automatically appends the count to the end of the segment files.

For example, suppose you're transcoding into srt format. When you enter "Sydney-{language}-sunrise", and the language of the captions is English (en), the name of the first caption file will be Sydney-en-sunrise00000.srt.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your caption formats.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify S3-AWS-KMS and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following

settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **Encryption** (*dict*) –

The encryption settings, if any, that you want Elastic Transcoder to apply to your output files. If you choose to use encryption, you must specify a mode to use. If you choose not to use encryption, Elastic Transcoder will write an unencrypted file to your Amazon S3 bucket.

- **Mode** (*string*) –

The specific server-side encryption mode that you want Elastic Transcoder to use when decrypting your input files or encrypting your output files. Elastic Transcoder supports the following options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.
- **S3-AWS-KMS:** Amazon S3 calls the Amazon Key Management Service, which creates and manages the keys that are used for encrypting your files. If you specify `S3-AWS-KMS` and you don't want to use the default key, you must add the

AWS-KMS key that you want to use to your pipeline.

- **AES-CBC-PKCS7:** A padded cipher-block mode of operation originally used for HLS files.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

For all three AES options, you must provide the following settings, which must be base64-encoded:

- **Key**
- **Key MD5**
- **Initialization Vector**

Warning: For the AES modes, your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to unencrypt your data.

- **Key** (*string*) –

The data encryption key that you want Elastic Transcoder to use to encrypt your output file, or that was used to encrypt your input file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using the Amazon Key Management Service.

- **KeyMd5** (*string*) –

The MD5 digest of the key that you used to encrypt your input file, or that you want Elastic Transcoder to use to encrypt your output file. Elastic Transcoder uses the key digest as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files or that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes long before being base64-encoded.

- **AppliedColorSpaceConversion** (*string*) –

If Elastic Transcoder used a preset with a `ColorSpaceConversionMode` to transcode the output file, the `AppliedColorSpaceConversion` parameter shows the conversion used. If no `ColorSpaceConversionMode` was defined in the preset, this parameter will not be included in the job response.

- **OutputKeyPrefix** (*string*) –

The value, if any, that you want Elastic Transcoder to prepend to the names of all files that this job creates, including output files, thumbnails, and playlists. We recommend that you add a / or some other delimiter to the end of the `OutputKeyPrefix`.

- **Playlists** (*list*) –

Warning: Outputs in Fragmented MP4 or MPEG-TS format only.

If you specify a preset in `PresetId` for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `Playlists` contains information about the master playlists that you want Elastic Transcoder to create.

The maximum number of master playlists in a job is 30.

- (*dict*) –

Use Only for Fragmented MP4 or MPEG-TS Outputs. If you specify a preset for which the value of `Container` is `mp4` (Fragmented MP4) or `ts` (MPEG-TS), `Playlists` contains information about the master playlists that you want Elastic Transcoder to create. We recommend that you create only one master playlist per output format. The maximum number of master playlists in a job is 30.

- **Name** (*string*) –

The name that you want Elastic Transcoder to assign to the master playlist, for example, `nyc-vacation.m3u8`. If the name includes a / character, the section of the name before the last / must be identical for all `Name` objects. If you create more than one master playlist, the values of all `Name` objects must be unique.

Note : Elastic Transcoder automatically appends the relevant file extension to the file name (`.m3u8` for `HLSv3` and `HLSv4` playlists, and `.ism` and `.ismc` for `Smooth` playlists). If you include a file extension in `Name`, the file name will have two extensions.

- **Format** (*string*) –

The format of the output playlist. Valid formats include `HLSv3`, `HLSv4`, and `Smooth`.

- **OutputKeys** (*list*) –

For each output in this job that you want to include in a master playlist, the value of the `Outputs:Key` object.

- If your output is not `HLS` or does not have a segment duration set, the name of the output file is a concatenation of `OutputKeyPrefix` and `Outputs:Key:OutputKeyPrefix“Outputs:Key“`
- If your output is `HLSv3` and has a segment duration set, or is not included in a playlist, Elastic Transcoder creates an output playlist file with a file extension of `.m3u8`, and a series of `.ts` files that include a five-digit sequential counter beginning with 00000: `OutputKeyPrefix“Outputs:Key“ .m3u8`
`OutputKeyPrefix“Outputs:Key“ 00000.ts`

- If your output is `HLSv4`, has a segment duration set, and is included in an `HLSv4` playlist, Elastic Transcoder creates an output playlist file with a file extension of `_v4.m3u8`. If the output is video, Elastic Transcoder also creates an output file with an extension of `_iframe.m3u8`: `OutputKeyPrefix“Outputs:Key“_v4.m3u8` `OutputKeyPrefix“Outputs:Key“_iframe.m3u8` `OutputKeyPrefix“Outputs:Key“.ts`

Elastic Transcoder automatically appends the relevant file extension to the file name. If you include a file extension in Output Key, the file name will have two extensions.

If you include more than one output in a playlist, any segment duration settings, clip settings, or caption settings must be the same for all outputs in the playlist. For Smooth playlists, the `Audio:Profile`, `Video:Profile`, and `Video:FrameRate` to `Video:KeyframesMaxDist` ratio must be the same for all outputs.

- *(string)* –
- **HlsContentProtection** (*dict*) –

The HLS content protection settings, if any, that you want Elastic Transcoder to apply to the output files associated with this playlist.

- **Method** (*string*) –

The content protection method for your output. The only valid value is: `aes-128`.

This value will be written into the method attribute of the `EXT-X-KEY` metadata tag in the output playlist.

- **Key** (*string*) –

If you want Elastic Transcoder to generate a key for you, leave this field blank.

If you choose to supply your own key, you must encrypt the key by using AWS KMS. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

`128`, `192`, or `256`.

- **KeyMd5** (*string*) –

If Elastic Transcoder is generating your key for you, you must leave this field blank.

The MD5 digest of the key that you want Elastic Transcoder to use to encrypt your output file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

- **InitializationVector** (*string*) –

If Elastic Transcoder is generating your key for you, you must leave this field blank.

The series of random bits created by a random bit generator,

unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

- **LicenseAcquisitionUrl** (*string*) –

The location of the license key required to decrypt your HLS playlist. The URL must be an absolute path, and is referenced in the URI attribute of the EXT-X-KEY metadata tag in the playlist file.

- **KeyStoragePolicy** (*string*) –

Specify whether you want Elastic Transcoder to write your HLS license key to an Amazon S3 bucket. If you choose `WithVariantPlaylists`, `LicenseAcquisitionUrl` must be left blank and Elastic Transcoder writes your data key into the same bucket as the associated playlist.

- **PlayReadyDrm** (*dict*) –

The DRM settings, if any, that you want Elastic Transcoder to apply to the output files associated with this playlist.

- **Format** (*string*) –

The type of DRM, if any, that you want Elastic Transcoder to apply to the output files associated with this playlist.

- **Key** (*string*) –

The DRM key for your file, provided by your DRM license provider. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

128 , 192 , or 256 .

The key must also be encrypted by using AWS KMS.

- **KeyMd5** (*string*) –

The MD5 digest of the key used for DRM on your file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

- **KeyId** (*string*) –

The ID for your DRM key, so that your DRM license provider knows which key to provide.

The key ID must be provided in big endian, and Elastic Transcoder will convert it to little endian before inserting it into the PlayReady DRM headers. If you are unsure whether your license server provides your key ID in big or little endian, check with your DRM provider.

- **InitializationVector** (*string*) –

The series of random bits created by a random bit generator, unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your files. The initialization vector must be base64-encoded, and it must be exactly 8 bytes

long before being base64-encoded. If no initialization vector is provided, Elastic Transcoder generates one for you.

- **LicenseAcquisitionUrl** (*string*) –

The location of the license key required to play DRM content. The URL must be an absolute path, and is referenced by the PlayReady header. The PlayReady header is referenced in the protection header of the client manifest for Smooth Streaming outputs, and in the EXT-X-DXDRM and EXT-XDXDRMINFO metadata tags for HLS playlist outputs. An example URL looks like this: <https://www.example.com/exampleKey/>

- **Status** (*string*) –

The status of the job with which the playlist is associated.

- **StatusDetail** (*string*) –

Information that further explains the status.

- **Status** (*string*) –

The status of the job: Submitted , Progressing , Complete , Canceled , or Error .

- **UserMetadata** (*dict*) –

User-defined metadata that you want to associate with an Elastic Transcoder job. You specify metadata in key/value pairs, and you can add up to 10 key/value pairs per job. Elastic Transcoder does not guarantee that key/value pairs will be returned in the same order in which you specify them.

Metadata keys and values must use characters from the following list:

- 0–9
- A–Z and a–z
- Space
- The following symbols: _ . : / = + - % @
- (*string*) –
- (*string*) –
- **Timing** (*dict*) –

Details about the timing of a job.

- **SubmitTimeMillis** (*integer*) –

The time the job was submitted to Elastic Transcoder, in epoch milliseconds.

- **StartTimeMillis** (*integer*) –

The time the job began transcoding, in epoch milliseconds.

- **FinishTimeMillis** (*integer*) –

The time the job finished transcoding, in epoch milliseconds.

- **NextToken** (*string*) –

A token to resume pagination.

class ElasticTranscoder.Paginator.**ListPipelines**

```
paginator = client.get_paginator('list_pipelines')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `ElasticTranscoder.Client.list_pipelines()`.

Request Syntax

```
response_iterator = paginator.paginate(
    Ascending='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **Ascending** (*string*) – To list pipelines in chronological order by the date and time that they were created, enter `true`. To list pipelines in reverse chronological order, enter `false`.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns**Response Syntax**

```
{
    'Pipelines': [
        {
            'Id': 'string',
            'Arn': 'string',
            'Name': 'string',
            'Status': 'string',
            'InputBucket': 'string',
            'OutputBucket': 'string',
            'Role': 'string',
            'AwsKmsKeyArn': 'string',
            'Notifications': {
                'Progressing': 'string',
                'Completed': 'string',
                'Warning': 'string',
                'Error': 'string'
            },
            'ContentConfig': {
                'Bucket': 'string',
                'StorageClass': 'string',
```



```

        'Permissions': [
            {
                'GranteeType': 'string',
                'Grantee': 'string',
                'Access': [
                    'string',
                ]
            },
        ]
    },
    'ThumbnailConfig': {
        'Bucket': 'string',
        'StorageClass': 'string',
        'Permissions': [
            {
                'GranteeType': 'string',
                'Grantee': 'string',
                'Access': [
                    'string',
                ]
            },
        ]
    }
},
'NextToken': 'string'
}

```

Response Structure

- *(dict)* –

A list of the pipelines associated with the current AWS account.

- **Pipelines** (*list*) –

An array of Pipeline objects.

- * *(dict)* –

The pipeline (queue) that is used to manage jobs.

- **Id** (*string*) –

The identifier for the pipeline. You use this value to identify the pipeline in which you want to perform a variety of operations, such as creating a job or a preset.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) for the pipeline.

- **Name** (*string*) –

The name of the pipeline. We recommend that the name be unique within the AWS account, but uniqueness is not enforced.

Constraints: Maximum 40 characters

- **Status** (*string*) –

The current status of the pipeline:

- **Active**: The pipeline is processing jobs.
- **Paused**: The pipeline is not currently processing jobs.
- **InputBucket** (*string*) –

The Amazon S3 bucket from which Elastic Transcoder gets media files for transcoding and the graphics files, if any, that you want to use for watermarks.

- **OutputBucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files, thumbnails, and playlists. Either you specify this value, or you specify both `ContentConfig` and `ThumbnailConfig`.

- **Role** (*string*) –

The IAM Amazon Resource Name (ARN) for the role that Elastic Transcoder uses to transcode jobs for this pipeline.

- **AwsKmsKeyArn** (*string*) –

The AWS Key Management Service (AWS KMS) key that you want to use with this pipeline.

If you use either `S3` or `S3-AWS-KMS` as your `Encryption:Mode`, you don't need to provide a key with your job because a default key, known as an AWS-KMS key, is created for you automatically. You need to provide an AWS-KMS key only if you want to use a non-default AWS-KMS key, or if you are using an `Encryption:Mode` of `AES-PKCS7`, `AES-CTR`, or `AES-GCM`.

- **Notifications** (*dict*) –

The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify to report job status.

Warning: To receive notifications, you must also subscribe to the new topic in the Amazon SNS console.

- **Progressing** (optional): The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process the job.
- **Completed** (optional): The Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing the job.
- **Warning** (optional): The Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition.
- **Error** (optional): The Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition.
- **Progressing** (*string*) –

The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process the job.

- **Completed** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing the job.

- **Warning** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition.

- **Error** (*string*) –

The Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition.

- **ContentConfig** (*dict*) –

Information about the Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists. Either you specify both `ContentConfig` and `ThumbnailConfig`, or you specify `OutputBucket`.

- **Bucket** : The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists.
- **Permissions** : A list of the users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access that you want them to have. * **GranteeType**: The type of value that appears in the `Grantee` object: * **Canonical** : Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution.
- **Email** : The registered email address of an AWS account.
- **Group** : One of the following predefined Amazon S3 groups: `AllUsers` , `AuthenticatedUsers` , or `LogDelivery`.
- **Grantee** : The AWS user or group that you want to have access to transcoded files and playlists.
- **Access** : The permission that you want to give to the AWS user that is listed in `Grantee`. Valid values include: * **READ** : The grantee can read the objects and metadata for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **READ_ACP** : The grantee can read the object ACL for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **WRITE_ACP** : The grantee can write the ACL for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **FULL_CONTROL** : The grantee has `READ` , `READ_ACP` , and `WRITE_ACP` permissions for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **StorageClass** : The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.
- **Bucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. Specify this value when all of the following are true:

- You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.
- You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- You do not want to specify the permissions that Elastic Transcoder grants to the files.
- You want to associate the transcoded files and thumbnails with the Amazon S3 `Standard` storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit Out-

`putBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

- **StorageClass** (*string*) –

The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

- **Permissions** (*list*) –

Optional. The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions`, Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role`. If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions`, Elastic Transcoder grants full control over the transcoded files and playlists to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

- (*dict*) –

The `Permission` structure.

- **GranteeType** (*string*) –

The type of value that appears in the `Grantee` object:

- `Canonical`: Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.
- `Email`: The registered email address of an AWS account.
- `Group`: One of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.
- **Grantee** (*string*) –

The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group.

- **Access** (*list*) –

The permission that you want to give to the AWS user that is listed in `Grantee`. Valid values include:

- `READ`: The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `READ_ACP`: The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.

- `WRITE_ACP` : The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `FULL_CONTROL` : The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- *(string)* –
- **ThumbnailConfig** (*dict*) –

Information about the Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files. Either you specify both `ContentConfig` and `ThumbnailConfig`, or you specify `OutputBucket`.

- `Bucket` : The Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files.
- `Permissions` : A list of the users and/or predefined Amazon S3 groups you want to have access to thumbnail files, and the type of access that you want them to have. * `GranteeType`: The type of value that appears in the `Grantee` object: * `Canonical` : Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. `warning::`A canonical user ID is not the same as an AWS account number.
- `Email` : The registered email address of an AWS account.
- `Group` : One of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.
- `Grantee` : The AWS user or group that you want to have access to thumbnail files.
- `Access`: The permission that you want to give to the AWS user that is listed in `Grantee`. Valid values include: * `READ` : The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `READ_ACP` : The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `WRITE_ACP` : The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `FULL_CONTROL` : The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- `StorageClass` : The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the thumbnails that it stores in your Amazon S3 bucket.
- **Bucket** (*string*) –

The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. Specify this value when all of the following are true:

- You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.
- You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- You do not want to specify the permissions that Elastic Transcoder grants to the files.
- You want to associate the transcoded files and thumbnails

with the Amazon S3 Standard storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit `OutputBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

- **StorageClass** (*string*) –

The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

- **Permissions** (*list*) –

Optional. The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions`, Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role`. If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions`, Elastic Transcoder grants full control over the transcoded files and playlists to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

- (*dict*) –

The `Permission` structure.

- **GranteeType** (*string*) –

The type of value that appears in the `Grantee` object:

- `Canonical`: Either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. .. warning::A canonical user ID is not the same as an AWS account number.
- `Email`: The registered email address of an AWS account.
- `Group`: One of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.

- **Grantee** (*string*) –

The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group.

- **Access** (*list*) –

The permission that you want to give to the AWS user that is listed in `Grantee`. Valid values include:

- **READ** : The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **READ_ACP** : The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **WRITE_ACP** : The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **FULL_CONTROL** : The grantee has **READ**, **READ_ACP**, and **WRITE_ACP** permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- *(string)* –
- **NextToken** *(string)* –

A token to resume pagination.

class ElasticTranscoder.Paginator.**ListPresets**

```
paginator = client.get_paginator('list_presets')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `ElasticTranscoder.Client.list_presets()`.

Request Syntax

```
response_iterator = paginator.paginate(
    Ascending='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **Ascending** *(string)* – To list presets in chronological order by the date and time that they were created, enter `true` . To list presets in reverse chronological order, enter `false` .
- **PaginationConfig** *(dict)* – A dictionary that provides parameters to control pagination.

- **MaxItems** *(integer)* –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** *(integer)* –

The size of each page.

- **StartingToken** *(string)* –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Presets': [
    {
      'Id': 'string',
      'Arn': 'string',
      'Name': 'string',
      'Description': 'string',
      'Container': 'string',
      'Audio': {
        'Codec': 'string',
        'SampleRate': 'string',
        'BitRate': 'string',
        'Channels': 'string',
        'AudioPackingMode': 'string',
        'CodecOptions': {
          'Profile': 'string',
          'BitDepth': 'string',
          'BitOrder': 'string',
          'Signed': 'string'
        }
      },
      'Video': {
        'Codec': 'string',
        'CodecOptions': {
          'string': 'string'
        },
        'KeyframesMaxDist': 'string',
        'FixedGOP': 'string',
        'BitRate': 'string',
        'FrameRate': 'string',
        'MaxFrameRate': 'string',
        'Resolution': 'string',
        'AspectRatio': 'string',
        'MaxWidth': 'string',
        'MaxHeight': 'string',
        'DisplayAspectRatio': 'string',
        'SizingPolicy': 'string',
        'PaddingPolicy': 'string',
        'Watermarks': [
          {
            'Id': 'string',
            'MaxWidth': 'string',
            'MaxHeight': 'string',
            'SizingPolicy': 'string',
            'HorizontalAlign': 'string',
            'HorizontalOffset': 'string',
            'VerticalAlign': 'string',
            'VerticalOffset': 'string',
            'Opacity': 'string',
            'Target': 'string'
          }
        ],
      },
    },
  ],
  'Thumbnails': {
    'Format': 'string',
    'Interval': 'string',
    'Resolution': 'string',
    'AspectRatio': 'string',
  },
}
```



```

        'MaxWidth': 'string',
        'MaxHeight': 'string',
        'SizingPolicy': 'string',
        'PaddingPolicy': 'string'
    },
    'Type': 'string'
},
],
'NextToken': 'string'
}

```

Response Structure

- (*dict*) –

The `ListPresetsResponse` structure.

- **Presets** (*list*) –

An array of `Preset` objects.

- * (*dict*) –

Presets are templates that contain most of the settings for transcoding media files from one format to another. Elastic Transcoder includes some default presets for common formats, for example, several iPod and iPhone versions. You can also create your own presets for formats that aren't included among the default presets. You specify which preset you want to use when you create a job.

- **Id** (*string*) –

Identifier for the new preset. You use this value to get settings for the preset or to delete it.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) for the preset.

- **Name** (*string*) –

The name of the preset.

- **Description** (*string*) –

A description of the preset.

- **Container** (*string*) –

The container type for the output file. Valid values include `flac`, `flv`, `fmp4`, `gif`, `mp3`, `mp4`, `mpg`, `mx4`, `oga`, `ogg`, `ts`, and `webm`.

- **Audio** (*dict*) –

A section of the response body that provides information about the audio preset values.

- **Codec** (*string*) –

The audio codec for the output file. Valid values include `aac`, `flac`, `mp2`, `mp3`, `pcm`, and `vorbis`.

- **SampleRate** (*string*) –

The sample rate of the audio stream in the output file, in Hertz. Valid values include:

`auto`, `22050`, `32000`, `44100`, `48000`, `96000`

If you specify `auto`, Elastic Transcoder automatically detects the sample rate.

- **BitRate** (*string*) –

The bit rate of the audio stream in the output file, in kilobits/second. Enter an integer between 64 and 320, inclusive.

- **Channels** (*string*) –

The number of audio channels in the output file. The following values are valid:

`auto`, `0`, `1`, `2`

One channel carries the information played by a single speaker. For example, a stereo track with two channels sends one channel to the left speaker, and the other channel to the right speaker. The output channels are organized into tracks. If you want Elastic Transcoder to automatically detect the number of audio channels in the input file and use that value for the output file, select `auto`.

The output of a specific channel value and inputs are as follows:

- `auto` **channel specified, with any input:** Pass through up to eight input channels.
- `0` **channels specified, with any input:** Audio omitted from the output.
- `1` **channel specified, with at least one input channel:** Mono sound.
- `2` **channels specified, with any input:** Two identical mono channels or stereo. For more information about tracks, see `Audio:AudioPackingMode`.

For more information about how Elastic Transcoder organizes channels and tracks, see `Audio:AudioPackingMode`.

- **AudioPackingMode** (*string*) –

The method of organizing audio channels and tracks. Use `Audio:Channels` to specify the number of channels in your output, and `Audio:AudioPackingMode` to specify the number of tracks and their relation to the channels. If you do not specify an `Audio:AudioPackingMode`, Elastic Transcoder uses `SingleTrack`.

The following values are valid:

`SingleTrack`, `OneChannelPerTrack`, and `OneChannelPerTrackWithMosTo8Tracks`

When you specify `SingleTrack`, Elastic Transcoder creates a single track for your output. The track can have up to eight channels. Use `SingleTrack` for all non-mxf containers.

The outputs of `SingleTrack` for a specific channel value and inputs are as follows:

- `0` **channels with any input:** Audio omitted from the output
- `1`, `2`, or `auto` **channels with no audio input:** Audio omitted from the output
- `1` **channel with any input with audio:** One track with one channel, downmixed if necessary

- **2 channels with one track with one channel:** One track with two identical channels
- **2 or auto channels with two tracks with one channel each:** One track with two channels
- **2 or auto channels with one track with two channels:** One track with two channels
- **2 channels with one track with multiple channels:** One track with two channels
- **auto channels with one track with one channel:** One track with one channel
- **auto channels with one track with multiple channels:** One track with multiple channels

When you specify `OneChannelPerTrack`, Elastic Transcoder creates a new track for every channel in your output. Your output can have up to eight single-channel tracks.

The outputs of `OneChannelPerTrack` for a specific channel value and inputs are as follows:

- **0 channels with any input:** Audio omitted from the output
- **1, 2, or auto channels with no audio input:** Audio omitted from the output
- **1 channel with any input with audio:** One track with one channel, downmixed if necessary
- **2 channels with one track with one channel:** Two tracks with one identical channel each
- **2 or auto channels with two tracks with one channel each:** Two tracks with one channel each
- **2 or auto channels with one track with two channels:** Two tracks with one channel each
- **2 channels with one track with multiple channels:** Two tracks with one channel each
- **auto channels with one track with one channel:** One track with one channel
- **auto channels with one track with multiple channels:** Up to eight tracks with one channel each

When you specify `OneChannelPerTrackWithMosTo8Tracks`, Elastic Transcoder creates eight single-channel tracks for your output. All tracks that do not contain audio data from an input channel are MOS, or Mit Out Sound, tracks.

The outputs of `OneChannelPerTrackWithMosTo8Tracks` for a specific channel value and inputs are as follows:

- **0 channels with any input:** Audio omitted from the output
- **1, 2, or auto channels with no audio input:** Audio omitted from the output
- **1 channel with any input with audio:** One track with one channel, downmixed if necessary, plus six MOS tracks
- **2 channels with one track with one channel:** Two tracks with one identical channel each, plus six MOS tracks
- **2 or auto channels with two tracks with one channel each:** Two tracks with one channel each, plus six MOS tracks
- **2 or auto channels with one track with two channels:** Two tracks with one channel each, plus six MOS tracks
- **2 channels with one track with multiple channels:** Two tracks with one channel each, plus six MOS tracks

- **auto channels with one track with one channel:** One track with one channel, plus seven MOS tracks
- **auto channels with one track with multiple channels:** Up to eight tracks with one channel each, plus MOS tracks until there are eight tracks in all
- **CodecOptions** (*dict*) –

If you specified AAC for `Audio:Codec`, this is the AAC compression profile to use. Valid values include:

`auto`, `AAC-LC`, `HE-AAC`, `HE-AACv2`

If you specify `auto`, Elastic Transcoder chooses a profile based on the bit rate of the output file.

- **Profile** (*string*) –

You can only choose an audio profile when you specify AAC for the value of `Audio:Codec`.

Specify the AAC profile for the output file. Elastic Transcoder supports the following profiles:

- `auto`: If you specify `auto`, Elastic Transcoder will select the profile based on the bit rate selected for the output file.
- `AAC-LC`: The most common AAC profile. Use for bit rates larger than 64 kbps.
- `HE-AAC`: Not supported on some older players and devices. Use for bit rates between 40 and 80 kbps.
- `HE-AACv2`: Not supported on some players and devices. Use for bit rates less than 48 kbps.

All outputs in a `Smooth` playlist must have the same value for `Profile`.

Note: If you created any presets before AAC profiles were added, Elastic Transcoder automatically updated your presets to use AAC-LC. You can change the value as required.

- **BitDepth** (*string*) –

You can only choose an audio bit depth when you specify `flac` or `pcm` for the value of `Audio:Codec`.

The bit depth of a sample is how many bits of information are included in the audio samples. The higher the bit depth, the better the audio, but the larger the file.

Valid values are `16` and `24`.

The most common bit depth is `24`.

- **BitOrder** (*string*) –

You can only choose an audio bit order when you specify `pcm` for the value of `Audio:Codec`.

The order the bits of a PCM sample are stored in.

The supported value is `LittleEndian`.

- **Signed** (*string*) –

You can only choose whether an audio sample is signed when you specify `pcm` for the value of `Audio:Codec`.

Whether audio samples are represented with negative and positive numbers (signed) or only positive numbers (unsigned).

The supported value is `Signed`.

- **Video** (*dict*) –

A section of the response body that provides information about the video preset values.

- **Codec** (*string*) –

The video codec for the output file. Valid values include `gif`, `H.264`, `mpeg2`, and `vp8`. You can only specify `vp8` when the container type is `webm`, `gif` when the container type is `gif`, and `mpeg2` when the container type is `mpg`.

- **CodecOptions** (*dict*) –

Profile (H.264/VP8 Only)

The H.264 profile that you want to use for the output file. Elastic Transcoder supports the following profiles:

- `baseline`: The profile most commonly used for videoconferencing and for mobile applications.
- `main`: The profile used for standard-definition digital TV broadcasts.
- `high`: The profile used for high-definition digital TV broadcasts and for Blu-ray discs.

Level (H.264 Only)

The H.264 level that you want to use for the output file. Elastic Transcoder supports the following levels:

`1`, `1b`, `1.1`, `1.2`, `1.3`, `2`, `2.1`, `2.2`, `3`, `3.1`, `3.2`, `4`, `4.1`

MaxReferenceFrames (H.264 Only)

Applicable only when the value of `Video:Codec` is `H.264`. The maximum number of previously decoded frames to use as a reference for decoding future frames. Valid values are integers 0 through 16, but we recommend that you not use a value greater than the following:

```
Min(Floor(Maximum decoded
picture buffer in macroblocks *
256 / (Width in pixels * Height
in pixels)), 16)
```

where *Width in pixels* and *Height in pixels* represent either `MaxWidth` and `MaxHeight`, or `Resolution`. *Maximum decoded picture buffer in macroblocks* depends on the value of the `Level` object. See the list below. (A macroblock is a block of pixels measuring 16x16.)

- `1` - 396
- `1b` - 396
- `1.1` - 900
- `1.2` - 2376
- `1.3` - 2376
- `2` - 2376
- `2.1` - 4752
- `2.2` - 8100
- `3` - 8100

- 3.1 - 18000
- 3.2 - 20480
- 4 - 32768
- 4.1 - 32768

MaxBitRate (Optional, H.264/MPEG2/VP8 only)

The maximum number of bits per second in a video buffer; the size of the buffer is specified by `BufferSize`. Specify a value between 16 and 62,500. You can reduce the bandwidth required to stream a video by reducing the maximum bit rate, but this also reduces the quality of the video.

BufferSize (Optional, H.264/MPEG2/VP8 only)

The maximum number of bits in any `x` seconds of the output video. This window is commonly 10 seconds, the standard segment duration when you're using FMP4 or MPEG-TS for the container type of the output video. Specify an integer greater than 0. If you specify `MaxBitRate` and omit `BufferSize`, Elastic Transcoder sets `BufferSize` to 10 times the value of `MaxBitRate`.

InterlacedMode (Optional, H.264/MPEG2 Only)

The interlace mode for the output video.

Interlaced video is used to double the perceived frame rate for a video by interlacing two fields (one field on every other line, the other field on the other lines) so that the human eye registers multiple pictures per frame. Interlacing reduces the bandwidth required for transmitting a video, but can result in blurred images and flickering.

Valid values include `Progressive` (no interlacing, top to bottom), `TopFirst` (top field first), `BottomFirst` (bottom field first), and `Auto`.

If `InterlaceMode` is not specified, Elastic Transcoder uses `Progressive` for the output. If `Auto` is specified, Elastic Transcoder interlaces the output.

ColorSpaceConversionMode (Optional, H.264/MPEG2 Only)

The color space conversion Elastic Transcoder applies to the output video. Color spaces are the algorithms used by the computer to store information about how to render color. `Bt.601` is the standard for standard definition video, while `Bt.709` is the standard for high definition video.

Valid values include `None`, `Bt709toBt601`, `Bt601toBt709`, and `Auto`.

If you chose `Auto` for `ColorSpaceConversionMode` and your output is interlaced, your frame rate is one of 23.97, 24, 25, 29.97, 50, or 60, your `SegmentDuration` is null, and you are using one of the resolution changes from the list below, Elastic Transcoder applies the following color space conversions:

- *Standard to HD, 720x480 to 1920x1080* - Elastic Transcoder applies Bt601ToBt709
- *Standard to HD, 720x576 to 1920x1080* - Elastic Transcoder applies Bt601ToBt709
- *HD to Standard, 1920x1080 to 720x480* - Elastic Transcoder applies Bt709ToBt601
- *HD to Standard, 1920x1080 to 720x576* - Elastic Transcoder applies Bt709ToBt601

Note: Elastic Transcoder may change the behavior of the `ColorSpaceConversionMode Auto` mode in the future. All outputs in a playlist must use the same `ColorSpaceConversionMode`.

If you do not specify a `ColorSpaceConversionMode`, Elastic Transcoder does not change the color space of a file. If you are unsure what `ColorSpaceConversionMode` was applied to your output file, you can check the `AppliedColorSpaceConversion` parameter included in your job response. If your job does not have an `AppliedColorSpaceConversion` in its response, no `ColorSpaceConversionMode` was applied.

ChromaSubsampling

The sampling pattern for the chroma (color) channels of the output video. Valid values include `yuv420p` and `yuv422p`.

`yuv420p` samples the chroma information of every other horizontal and every other vertical line, `yuv422p` samples the color information of every horizontal line and every other vertical line.

LoopCount (Gif Only)

The number of times you want the output gif to loop. Valid values include `Infinite` and integers between 0 and 100, inclusive.

- *(string)* –
- *(string)* –
- **KeyframesMaxDist** *(string)* –

Applicable only when the value of `Video:Codec` is one of `H.264`, `MPEG2`, or `VP8`.

The maximum number of frames between key frames. Key frames are fully encoded frames; the frames between key frames are encoded based, in part, on the content of the key frames. The value is an integer formatted as a string; valid values are between 1 (every frame is a key frame) and 100000, inclusive. A higher value results in higher compression but may also discernibly decrease video quality.

For Smooth outputs, the `FrameRate` must have a constant ratio to the `KeyframesMaxDist`. This allows Smooth playlists to switch between different quality levels while the file is being played.

For example, an input file can have a `FrameRate` of 30 with a `KeyframesMaxDist` of 90. The output file then needs to

have a ratio of 1:3. Valid outputs would have `FrameRate` of 30, 25, and 10, and `KeyframesMaxDist` of 90, 75, and 30, respectively.

Alternately, this can be achieved by setting `FrameRate` to `auto` and having the same values for `MaxFrameRate` and `KeyframesMaxDist`.

- **FixedGOP** (*string*) –

Applicable only when the value of `Video:Codec` is one of `H.264`, `MPEG2`, or `VP8`.

Whether to use a fixed value for `FixedGOP`. Valid values are `true` and `false`:

- `true`: Elastic Transcoder uses the value of `KeyframesMaxDist` for the distance between key frames (the number of frames in a group of pictures, or GOP).
- `false`: The distance between key frames can vary.

Warning: `FixedGOP` must be set to `true` for `h264` containers.

- **BitRate** (*string*) –

The bit rate of the video stream in the output file, in kilobits/second. Valid values depend on the values of `Level` and `Profile`. If you specify `auto`, Elastic Transcoder uses the detected bit rate of the input source. If you specify a value other than `auto`, we recommend that you specify a value less than or equal to the maximum H.264-compliant value listed for your level and profile:

*Level - Maximum video bit rate in kilobits/second
(baseline and main Profile) : maximum video bit
rate in kilobits/second (high Profile)*

- 1 - 64 : 80
- 1b - 128 : 160
- 1.1 - 192 : 240
- 1.2 - 384 : 480
- 1.3 - 768 : 960
- 2 - 2000 : 2500
- 3 - 10000 : 12500
- 3.1 - 14000 : 17500
- 3.2 - 20000 : 25000
- 4 - 20000 : 25000
- 4.1 - 50000 : 62500

- **FrameRate** (*string*) –

The frames per second for the video stream in the output file. Valid values include:

`auto`, `10`, `15`, `23.97`, `24`, `25`, `29.97`, `30`, `60`

If you specify `auto`, Elastic Transcoder uses the detected frame rate of the input source. If you specify a frame rate, we recommend that you perform the following calculation:

Frame rate = maximum recommended
decoding speed in luma


```
samples/second / (width in
pixels * height in pixels)
```

where:

- *width in pixels* and *height in pixels* represent the Resolution of the output video.
- *maximum recommended decoding speed in Luma samples/second* is less than or equal to the maximum value listed in the following table, based on the value that you specified for Level.

The maximum recommended decoding speed in Luma samples/second for each level is described in the following list (*Level - Decoding speed*):

- 1 - 380160
- 1b - 380160
- 1.1 - 76800
- 1.2 - 1536000
- 1.3 - 3041280
- 2 - 3041280
- 2.1 - 5068800
- 2.2 - 5184000
- 3 - 10368000
- 3.1 - 27648000
- 3.2 - 55296000
- 4 - 62914560
- 4.1 - 62914560
- **MaxFrameRate** (*string*) –

If you specify `auto` for `FrameRate`, Elastic Transcoder uses the frame rate of the input video for the frame rate of the output video. Specify the maximum frame rate that you want Elastic Transcoder to use when the frame rate of the input video is greater than the desired maximum frame rate of the output video. Valid values include: 10, 15, 23.97, 24, 25, 29.97, 30, 60.

- **Resolution** (*string*) –

Warning: To better control resolution and aspect ratio of output videos, we recommend that you use the values `MaxWidth`, `MaxHeight`, `SizingPolicy`, `PaddingPolicy`, and `DisplayAspectRatio` instead of `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The width and height of the video in the output file, in pixels. Valid values are `auto` and *width x height*:

- `auto`: Elastic Transcoder attempts to preserve the width and height of the input file, subject to the following rules.
- **width* x *height**: The width and height of the output video in pixels.

Note the following about specifying the width and height:

- The width must be an even integer between 128 and 4096, inclusive.
- The height must be an even integer between 96 and 3072,

inclusive.

- If you specify a resolution that is less than the resolution of the input file, Elastic Transcoder rescales the output file to the lower resolution.
- If you specify a resolution that is greater than the resolution of the input file, Elastic Transcoder rescales the output to the higher resolution.
- We recommend that you specify a resolution for which the product of width and height is less than or equal to the applicable value in the following list (*List - Max width x height value*):
- 1 - 25344
- 1b - 25344
- 1.1 - 101376
- 1.2 - 101376
- 1.3 - 101376
- 2 - 101376
- 2.1 - 202752
- 2.2 - 404720
- 3 - 404720
- 3.1 - 921600
- 3.2 - 1310720
- 4 - 2097152
- 4.1 - 2097152
- **AspectRatio** (*string*) –

Warning: To better control resolution and aspect ratio of output videos, we recommend that you use the values `MaxWidth`, `MaxHeight`, `SizingPolicy`, `PaddingPolicy`, and `DisplayAspectRatio` instead of `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The display aspect ratio of the video in the output file. Valid values include:

`auto`, `1:1`, `4:3`, `3:2`, `16:9`

If you specify `auto`, Elastic Transcoder tries to preserve the aspect ratio of the input file.

If you specify an aspect ratio for the output file that differs from aspect ratio of the input file, Elastic Transcoder adds pillarboxing (black bars on the sides) or letterboxing (black bars on the top and bottom) to maintain the aspect ratio of the active region of the video.

- **MaxWidth** (*string*) –

The maximum width of the output video in pixels. If you specify `auto`, Elastic Transcoder uses 1920 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 128 and 4096.

- **MaxHeight** (*string*) –

The maximum height of the output video in pixels. If you

specify `auto`, Elastic Transcoder uses 1080 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 96 and 3072.

- **DisplayAspectRatio** (*string*) –

The value that Elastic Transcoder adds to the metadata in the output file.

- **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of the output video:

- **Fit** : Elastic Transcoder scales the output video so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Fill** : Elastic Transcoder scales the output video so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output video and then crops it in the dimension (if any) that exceeds the maximum value.
- **Stretch** : Elastic Transcoder stretches the output video to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input video and the output video are different, the output video will be distorted.
- **Keep** : Elastic Transcoder does not scale the output video. If either dimension of the input video exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output video.
- **ShrinkToFit** : Elastic Transcoder scales the output video down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the video up.
- **ShrinkToFill** : Elastic Transcoder scales the output video down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the video up.

- **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder may add black bars to the top and bottom and/or left and right sides of the output video to make the total size of the output video match the values that you specified for `MaxWidth` and `MaxHeight`.

- **Watermarks** (*list*) –

Settings for the size, location, and opacity of graphics that you want Elastic Transcoder to overlay over videos that are transcoded using this preset. You can specify settings for up to four watermarks. Watermarks appear in the specified size and location, and with the specified opacity for the duration of the transcoded video.

Watermarks can be in `.png` or `.jpg` format. If you want to display a watermark that is not rectangular, use the `.png` format,

which supports transparency.

When you create a job that uses this preset, you specify the .png or .jpg graphics that you want Elastic Transcoder to include in the transcoded videos. You can specify fewer graphics in the job than you specify watermark settings in the preset, which allows you to use the same preset for up to four watermarks that have different dimensions.

- *(dict)* –

Settings for the size, location, and opacity of graphics that you want Elastic Transcoder to overlay over videos that are transcoded using this preset. You can specify settings for up to four watermarks. Watermarks appear in the specified size and location, and with the specified opacity for the duration of the transcoded video.

Watermarks can be in .png or .jpg format. If you want to display a watermark that is not rectangular, use the .png format, which supports transparency.

When you create a job that uses this preset, you specify the .png or .jpg graphics that you want Elastic Transcoder to include in the transcoded videos. You can specify fewer graphics in the job than you specify watermark settings in the preset, which allows you to use the same preset for up to four watermarks that have different dimensions.

- **Id** (*string*) – A unique identifier for the settings for one watermark. The value of `Id` can be up to 40 characters long.
- **MaxWidth** (*string*) –

The maximum width of the watermark in one of the following formats:

- number of pixels (px): The minimum value is 16 pixels, and the maximum value is the value of `MaxWidth`.
- integer percentage (%): The range of valid values is 0 to 100. Use the value of `Target` to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the calculation.

If you specify the value in pixels, it must be less than or equal to the value of `MaxWidth`.

- **MaxHeight** (*string*) –

The maximum height of the watermark in one of the following formats:

- number of pixels (px): The minimum value is 16 pixels, and the maximum value is the value of `MaxHeight`.
- integer percentage (%): The range of valid values is 0 to 100. Use the value of `Target` to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the calculation.

If you specify the value in pixels, it must be less than or equal to the value of `MaxHeight`.

- **SizingPolicy** (*string*) –

A value that controls scaling of the watermark:

- **Fit** : Elastic Transcoder scales the watermark so it matches

the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.

- **Stretch** : Elastic Transcoder stretches the watermark to match the values that you specified for `MaxWidth` and `MaxHeight` . If the relative proportions of the watermark and the values of `MaxWidth` and `MaxHeight` are different, the watermark will be distorted.
- **ShrinkToFit** : Elastic Transcoder scales the watermark down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the watermark up.
- **HorizontalAlign** (*string*) –

The horizontal position of the watermark unless you specify a non-zero value for `HorizontalOffset` :

- **Left** : The left edge of the watermark is aligned with the left border of the video.
- **Right** : The right edge of the watermark is aligned with the right border of the video.
- **Center** : The watermark is centered between the left and right borders.
- **HorizontalOffset** (*string*) –

The amount by which you want the horizontal position of the watermark to be offset from the position specified by `HorizontalAlign`:

- number of pixels (px): The minimum value is 0 pixels, and the maximum value is the value of `MaxWidth`.
- integer percentage (%): The range of valid values is 0 to 100. For example, if you specify `Left` for `HorizontalAlign` and `5px` for `HorizontalOffset` , the left side of the watermark appears 5 pixels from the left border of the output video.

`HorizontalOffset` is only valid when the value of `HorizontalAlign` is `Left` or `Right` . If you specify an offset that causes the watermark to extend beyond the left or right border and Elastic Transcoder has not added black bars, the watermark is cropped. If Elastic Transcoder has added black bars, the watermark extends into the black bars. If the watermark extends beyond the black bars, it is cropped.

Use the value of `Target` to specify whether you want to include the black bars that are added by Elastic Transcoder, if any, in the offset calculation.

- **VerticalAlign** (*string*) –

The vertical position of the watermark unless you specify a non-zero value for `VerticalOffset` :

- **Top** : The top edge of the watermark is aligned with the top border of the video.
- **Bottom** : The bottom edge of the watermark is aligned with the bottom border of the video.
- **Center** : The watermark is centered between the top and bottom borders.

- **VerticalOffset** (*string*) – VerticalOffset

The amount by which you want the vertical position of the watermark to be offset from the position specified by VerticalAlign:

- number of pixels (px): The minimum value is 0 pixels, and the maximum value is the value of MaxHeight .
- integer percentage (%): The range of valid values is 0 to 100. For example, if you specify Top for VerticalAlign and 5px for VerticalOffset , the top of the watermark appears 5 pixels from the top border of the output video.

VerticalOffset is only valid when the value of VerticalAlign is Top or Bottom.

If you specify an offset that causes the watermark to extend beyond the top or bottom border and Elastic Transcoder has not added black bars, the watermark is cropped. If Elastic Transcoder has added black bars, the watermark extends into the black bars. If the watermark extends beyond the black bars, it is cropped.

Use the value of Target to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the offset calculation.

- **Opacity** (*string*) –

A percentage that indicates how much you want a watermark to obscure the video in the location where it appears. Valid values are 0 (the watermark is invisible) to 100 (the watermark completely obscures the video in the specified location). The datatype of Opacity is float.

Elastic Transcoder supports transparent .png graphics. If you use a transparent .png, the transparent portion of the video appears as if you had specified a value of 0 for Opacity . The .jpg file format doesn't support transparency.

- **Target** (*string*) –

A value that determines how Elastic Transcoder interprets values that you specified for HorizontalOffset , VerticalOffset , MaxWidth , and MaxHeight :

- **Content** : HorizontalOffset and VerticalOffset values are calculated based on the borders of the video excluding black bars added by Elastic Transcoder, if any. In addition, MaxWidth and MaxHeight , if specified as a percentage, are calculated based on the borders of the video excluding black bars added by Elastic Transcoder, if any.
- **Frame** : HorizontalOffset and VerticalOffset values are calculated based on the borders of the video including black bars added by Elastic Transcoder, if any. In addition, MaxWidth and MaxHeight , if specified as a percentage, are calculated based on the borders of the video including black bars added by Elastic Transcoder, if any.
- **Thumbnails** (*dict*) –

A section of the response body that provides information about the thumbnail preset values, if any.

- **Format** (*string*) –

The format of thumbnails, if any. Valid values are `jpg` and `png`.

You specify whether you want Elastic Transcoder to create thumbnails when you create a job.

- **Interval** (*string*) –

The approximate number of seconds between thumbnails. Specify an integer value.

- **Resolution** (*string*) –

Warning: To better control resolution and aspect ratio of thumbnails, we recommend that you use the values `MaxWidth`, `MaxHeight`, `SizingPolicy`, and `PaddingPolicy` instead of `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The width and height of thumbnail files in pixels. Specify a value in the format `*width* x *height*` where both values are even integers. The values cannot exceed the width and height that you specified in the `Video:Resolution` object.

- **AspectRatio** (*string*) –

Warning: To better control resolution and aspect ratio of thumbnails, we recommend that you use the values `MaxWidth`, `MaxHeight`, `SizingPolicy`, and `PaddingPolicy` instead of `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The aspect ratio of thumbnails. Valid values include:

`auto`, `1:1`, `4:3`, `3:2`, `16:9`

If you specify `auto`, Elastic Transcoder tries to preserve the aspect ratio of the video in the output file.

- **MaxWidth** (*string*) –

The maximum width of thumbnails in pixels. If you specify `auto`, Elastic Transcoder uses 1920 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 32 and 4096.

- **MaxHeight** (*string*) –

The maximum height of thumbnails in pixels. If you specify `auto`, Elastic Transcoder uses 1080 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 32 and 3072.

- **SizingPolicy** (*string*) –

Specify one of the following values to control scaling of thumbnails:

- **Fit** : Elastic Transcoder scales thumbnails so they match the value that you specified in thumbnail `MaxWidth` or `MaxHeight` settings without exceeding the other value.
- **Fill** : Elastic Transcoder scales thumbnails so they match the value that you specified in thumbnail `MaxWidth` or `MaxHeight` settings and matches or exceeds the other value. Elastic Transcoder centers the image in thumbnails and then crops in the dimension (if any) that exceeds the maximum value.
- **Stretch** : Elastic Transcoder stretches thumbnails to match the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings. If the relative proportions of the input video and thumbnails are different, the thumbnails will be distorted.
- **Keep** : Elastic Transcoder does not scale thumbnails. If either dimension of the input video exceeds the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings, Elastic Transcoder crops the thumbnails.
- **ShrinkToFit** : Elastic Transcoder scales thumbnails down so that their dimensions match the values that you specified for at least one of thumbnail `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale thumbnails up.
- **ShrinkToFill** : Elastic Transcoder scales thumbnails down so that their dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale thumbnails up.
- **PaddingPolicy** (*string*) –

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder may add black bars to the top and bottom and/or left and right sides of thumbnails to make the total size of the thumbnails match the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings.

- **Type** (*string*) –

Whether the preset is a default preset provided by Elastic Transcoder (`System`) or a preset that you have defined (`Custom`).

- **NextToken** (*string*) –

A token to resume pagination.

Waiters

The available waiters are:

- `ElasticTranscoder.Waiter.JobComplete`

class `ElasticTranscoder.Waiter.JobComplete`

```
waiter = client.get_waiter('job_complete')
```


wait (**kwargs)

Polls `ElasticTranscoder.Client.read_job()` every 30 seconds until a successful state is reached. An error is returned after 120 failed checks.

Request Syntax

```
waiter.wait(
    Id='string'
)
```

Parameters `Id` (*string*) – [REQUIRED]

The identifier of the job for which you want to get detailed information.

Returns None

ElasticLoadBalancing

Table of Contents

- [ElasticLoadBalancing](#)
 - [Client](#)
 - [Paginators](#)

Client

class `ElasticLoadBalancing.Client`

A low-level client representing Elastic Load Balancing:

```
import boto3
```

```
client = boto3.client('elb')
```

These are the available methods:

- `add_tags()`
- `apply_security_groups_to_load_balancer()`
- `attach_load_balancer_to_subnets()`
- `can_paginate()`
- `configure_health_check()`
- `create_app_cookie_stickiness_policy()`
- `create_lb_cookie_stickiness_policy()`
- `create_load_balancer()`
- `create_load_balancer_listeners()`
- `create_load_balancer_policy()`
- `delete_load_balancer()`
- `delete_load_balancer_listeners()`
- `delete_load_balancer_policy()`
- `deregister_instances_from_load_balancer()`
- `describe_instance_health()`
- `describe_load_balancer_attributes()`
- `describe_load_balancer_policies()`
- `describe_load_balancer_policy_types()`
- `describe_load_balancers()`

- `describe_tags()`
- `detach_load_balancer_from_subnets()`
- `disable_availability_zones_for_load_balancer()`
- `enable_availability_zones_for_load_balancer()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_waiter()`
- `modify_load_balancer_attributes()`
- `register_instances_with_load_balancer()`
- `remove_tags()`
- `set_load_balancer_listener_ssl_certificate()`
- `set_load_balancer_policies_for_backend_server()`
- `set_load_balancer_policies_of_listener()`

add_tags (***kwargs*)

Adds the specified tags to the specified load balancer. Each load balancer can have a maximum of 10 tags.

Each tag consists of a key and an optional value. If a tag with the same key is already associated with the load balancer, AddTags updates its value.

For more information, see [Tag Your Load Balancer](#) in the *Elastic Load Balancing Developer Guide* .

Request Syntax

```
response = client.add_tags(
    LoadBalancerNames=[
        'string',
    ],
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **LoadBalancerNames** (*list*) – [REQUIRED]

The name of the load balancer. You can specify one load balancer only.

– (*string*) –

- **Tags** (*list*) – [REQUIRED]

The tags.

– (*dict*) –

Information about a tag.

- * **Key** (*string*) – [REQUIRED]

The key of the tag.

- * **Value** (*string*) –

The value of the tag.

Return type dict

Returns**Response Syntax**

```
{ }
```

Response Structure

- (*dict*) –

apply_security_groups_to_load_balancer (***kwargs*)

Associates one or more security groups with your load balancer in a virtual private cloud (VPC). The specified security groups override the previously associated security groups.

For more information, see [Security Groups for Load Balancers in a VPC](#) in the *Elastic Load Balancing Developer Guide*.

Request Syntax

```
response = client.apply_security_groups_to_load_balancer(
    LoadBalancerName='string',
    SecurityGroups=[
        'string',
    ]
)
```

Parameters

- **LoadBalancerName** (*string*) – [REQUIRED]

The name of the load balancer.

- **SecurityGroups** (*list*) – [REQUIRED]

The IDs of the security groups to associate with the load balancer. Note that you cannot specify the name of the security group.

– (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'SecurityGroups': [
        'string',
    ]
}
```

Response Structure

- (*dict*) –
 - **SecurityGroups** (*list*) –

The IDs of the security groups associated with the load balancer.

* (*string*) –

attach_load_balancer_to_subnets (***kwargs*)

Adds one or more subnets to the set of configured subnets for the specified load balancer.

The load balancer evenly distributes requests across all registered subnets. For more information, see [Add or Remove Subnets for Your Load Balancer in a VPC](#) in the *Elastic Load Balancing Developer Guide*.

Request Syntax

```
response = client.attach_load_balancer_to_subnets(
    LoadBalancerName='string',
    Subnets=[
        'string',
    ]
)
```

Parameters

- **LoadBalancerName** (*string*) – [REQUIRED]

The name of the load balancer.

- **Subnets** (*list*) – [REQUIRED]

The IDs of the subnets to add for the load balancer. You can add only one subnet per Availability Zone.

– (*string*) –

Return type dict

Returns**Response Syntax**

```
{
    'Subnets': [
        'string',
    ]
}
```

Response Structure

- (*dict*) –
 - **Subnets** (*list*) –

The IDs of the subnets attached to the load balancer.

* (*string*) –

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

configure_health_check (***kwargs*)

Specifies the health check settings to use when evaluating the health state of your back-end instances.

For more information, see [Configure Health Checks](#) in the *Elastic Load Balancing Developer Guide*.

Request Syntax

```
response = client.configure_health_check(
    LoadBalancerName='string',
    HealthCheck={
        'Target': 'string',
        'Interval': 123,
        'Timeout': 123,
        'UnhealthyThreshold': 123,
        'HealthyThreshold': 123
    }
)
```

Parameters

- **LoadBalancerName** (*string*) – [REQUIRED]

The name of the load balancer.

- **HealthCheck** (*dict*) – [REQUIRED]

The configuration information for the new health check.

- **Target** (*string*) – [REQUIRED]

The instance being checked. The protocol is either TCP, HTTP, HTTPS, or SSL. The range of valid ports is one (1) through 65535.

TCP is the default, specified as a TCP: port pair, for example “TCP:5000”. In this case, a health check simply attempts to open a TCP connection to the instance on the specified port. Failure to connect within the configured timeout is considered unhealthy.

SSL is also specified as SSL: port pair, for example, SSL:5000.

For HTTP/HTTPS, you must include a ping path in the string. HTTP is specified as a HTTP:port;/PathToPing; grouping, for example “HTTP:80/weather/us/wa/seattle”. In this case, a HTTP GET request is issued to the instance on the given port and path. Any answer other than “200 OK” within the timeout period is considered unhealthy.

The total length of the HTTP ping target must be 1024 16-bit Unicode characters or less.

- **Interval** (*integer*) – [REQUIRED]

The approximate interval, in seconds, between health checks of an individual instance.

- **Timeout** (*integer*) – [REQUIRED]

The amount of time, in seconds, during which no response means a failed health check.

This value must be less than the `Interval` value.

- **UnhealthyThreshold** (*integer*) – [REQUIRED]

The number of consecutive health check failures required before moving the instance to the `Unhealthy` state.

- **HealthyThreshold** (*integer*) – [REQUIRED]

The number of consecutive health checks successes required before moving the instance to the `Healthy` state.

Return type `dict`

Returns

Response Syntax

```
{
  'HealthCheck': {
    'Target': 'string',
    'Interval': 123,
    'Timeout': 123,
    'UnhealthyThreshold': 123,
    'HealthyThreshold': 123
  }
}
```

Response Structure

- (*dict*) –
 - **HealthCheck** (*dict*) –

The updated health check.

* **Target** (*string*) –

The instance being checked. The protocol is either TCP, HTTP, HTTPS, or SSL. The range of valid ports is one (1) through 65535.

TCP is the default, specified as a TCP: port pair, for example “TCP:5000”. In this case, a health check simply attempts to open a TCP connection to the instance on the specified port. Failure to connect within the configured timeout is considered unhealthy.

SSL is also specified as SSL: port pair, for example, SSL:5000.

For HTTP/HTTPS, you must include a ping path in the string. HTTP is specified as a HTTP:port;/PathToPing; grouping, for example “HTTP:80/weather/us/wa/seattle”. In this case, a HTTP GET request is issued to the instance on the given port and path. Any answer other than “200 OK” within the timeout period is considered unhealthy.

The total length of the HTTP ping target must be 1024 16-bit Unicode characters or less.

* **Interval** (*integer*) –

The approximate interval, in seconds, between health checks of an individual instance.

* **Timeout** (*integer*) –

The amount of time, in seconds, during which no response means a failed health check.

This value must be less than the `Interval` value.

* **UnhealthyThreshold** (*integer*) –

The number of consecutive health check failures required before moving the instance to the `Unhealthy` state.

* **HealthyThreshold** (*integer*) –

The number of consecutive health checks successes required before moving the instance to the `Healthy` state.

create_app_cookie_stickiness_policy (***kwargs*)

Generates a stickiness policy with sticky session lifetimes that follow that of an application-generated cookie. This policy can be associated only with HTTP/HTTPS listeners.

This policy is similar to the policy created by `CreateLBCookieStickinessPolicy`, except that the lifetime of the special Elastic Load Balancing cookie, `AWSELB`, follows the lifetime of the application-generated cookie specified in the policy configuration. The load balancer only inserts a new stickiness cookie when the application response includes a new application cookie.

If the application cookie is explicitly removed or expires, the session stops being sticky until a new application cookie is issued.

For more information, see [Application-Controlled Session Stickiness](#) in the *Elastic Load Balancing Developer Guide*.

Request Syntax

```
response = client.create_app_cookie_stickiness_policy(
    LoadBalancerName='string',
    PolicyName='string',
```

```
CookieName='string'
)
```

Parameters

- **LoadBalancerName** (*string*) – [REQUIRED]
The name of the load balancer.
- **PolicyName** (*string*) – [REQUIRED]
The name of the policy being created. This name must be unique within the set of policies for this load balancer.
- **CookieName** (*string*) – [REQUIRED]
The name of the application cookie used for stickiness.

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –

create_lb_cookie_stickiness_policy (***kwargs*)

Generates a stickiness policy with sticky session lifetimes controlled by the lifetime of the browser (user-agent) or a specified expiration period. This policy can be associated only with HTTP/HTTPS listeners.

When a load balancer implements this policy, the load balancer uses a special cookie to track the back-end server instance for each request. When the load balancer receives a request, it first checks to see if this cookie is present in the request. If so, the load balancer sends the request to the application server specified in the cookie. If not, the load balancer sends the request to a server that is chosen based on the existing load-balancing algorithm.

A cookie is inserted into the response for binding subsequent requests from the same user to that server. The validity of the cookie is based on the cookie expiration time, which is specified in the policy configuration.

For more information, see [Duration-Based Session Stickiness](#) in the *Elastic Load Balancing Developer Guide*.

Request Syntax

```
response = client.create_lb_cookie_stickiness_policy(
    LoadBalancerName='string',
    PolicyName='string',
    CookieExpirationPeriod=123
)
```

Parameters

- **LoadBalancerName** (*string*) – [REQUIRED]
The name of the load balancer.
- **PolicyName** (*string*) – [REQUIRED]
The name of the policy being created. This name must be unique within the set of policies for this load balancer.
- **CookieExpirationPeriod** (*integer*) – The time period, in seconds, after which the cookie should be considered stale. If you do not specify this parameter, the sticky session lasts for the duration of the browser session.

Return type dict

Returns

Response Syntax

```
{}
```

Response Structure

- (*dict*) –

create_load_balancer (***kwargs*)

Creates a load balancer.

If the call completes successfully, a new load balancer is created with a unique Domain Name Service (DNS) name. The load balancer receives incoming traffic and routes it to the registered instances. For more information, see [How Elastic Load Balancing Works](#) in the *Elastic Load Balancing Developer Guide* .

You can create up to 20 load balancers per region per account. You can request an increase for the number of load balancers for your account. For more information, see [Elastic Load Balancing Limits](#) in the *Elastic Load Balancing Developer Guide* .

Request Syntax

```
response = client.create_load_balancer(
    LoadBalancerName='string',
    Listeners=[
        {
            'Protocol': 'string',
            'LoadBalancerPort': 123,
            'InstanceProtocol': 'string',
            'InstancePort': 123,
            'SSLCertificateId': 'string'
        },
    ],
    AvailabilityZones=[
        'string',
    ],
    Subnets=[
        'string',
    ],
    SecurityGroups=[
        'string',
    ],
    Scheme='string',
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **LoadBalancerName** (*string*) – **[REQUIRED]**

The name of the load balancer.

This name must be unique within your AWS account, must have a maximum of 32 characters, must contain only alphanumeric characters or hyphens, and cannot begin or end with a hyphen.

- **Listeners** (*list*) – [REQUIRED]

The listeners.

For more information, see [Listeners for Your Load Balancer](#) in the *Elastic Load Balancing Developer Guide*.

- (*dict*) –

Information about a listener.

For information about the protocols and the ports supported by Elastic Load Balancing, see [Listener Configurations for Elastic Load Balancing](#) in the *Elastic Load Balancing Developer Guide*.

- * **Protocol** (*string*) – [REQUIRED]

The load balancer transport protocol to use for routing: HTTP, HTTPS, TCP, or SSL.

- * **LoadBalancerPort** (*integer*) – [REQUIRED]

The port on which the load balancer is listening. The supported ports are: 25, 80, 443, 465, 587, and 1024-65535.

- * **InstanceProtocol** (*string*) –

The protocol to use for routing traffic to back-end instances: HTTP, HTTPS, TCP, or SSL.

If the front-end protocol is HTTP, HTTPS, TCP, or SSL, `InstanceProtocol` must be at the same protocol.

If there is another listener with the same `InstancePort` whose `InstanceProtocol` is secure, (HTTPS or SSL), the listener's `InstanceProtocol` must also be secure.

If there is another listener with the same `InstancePort` whose `InstanceProtocol` is HTTP or TCP, the listener's `InstanceProtocol` must be HTTP or TCP.

- * **InstancePort** (*integer*) – [REQUIRED]

The port on which the instance is listening.

- * **SSLCertificateId** (*string*) –

The Amazon Resource Name (ARN) of the server certificate.

- **AvailabilityZones** (*list*) – One or more Availability Zones from the same region as the load balancer. Traffic is equally distributed across all specified Availability Zones.

You must specify at least one Availability Zone.

You can add more Availability Zones after you create the load balancer using `EnableAvailabilityZonesForLoadBalancer`.

- (*string*) –

- **Subnets** (*list*) – The IDs of the subnets in your VPC to attach to the load balancer. Specify one subnet per Availability Zone specified in `AvailabilityZones`.

- (*string*) –

- **SecurityGroups** (*list*) – The IDs of the security groups to assign to the load balancer.

- (*string*) –

- **Scheme** (*string*) – The type of a load balancer. Valid only for load balancers in a VPC.

By default, Elastic Load Balancing creates an Internet-facing load balancer with a publicly resolvable DNS name, which resolves to public IP addresses. For more information about Internet-facing and Internal load balancers, see [Internet-facing and Internal Load Balancers](#) in the *Elastic Load Balancing Developer Guide*.

Specify `internal` to create an internal load balancer with a DNS name that resolves to private IP addresses.

- **Tags** (*list*) – A list of tags to assign to the load balancer.

For more information about tagging your load balancer, see [Tagging](#) in the *Elastic Load Balancing Developer Guide*.

– (*dict*) –

Information about a tag.

* **Key** (*string*) – **[REQUIRED]**

The key of the tag.

* **Value** (*string*) –

The value of the tag.

Return type dict

Returns

Response Syntax

```
{
    'DNSName': 'string'
}
```

Response Structure

- (*dict*) –
 - **DNSName** (*string*) –

The DNS name of the load balancer.

create_load_balancer_listeners (***kwargs*)

Creates one or more listeners for the specified load balancer. If a listener with the specified port does not already exist, it is created; otherwise, the properties of the new listener must match the properties of the existing listener.

For more information, see [Add a Listener to Your Load Balancer](#) in the *Elastic Load Balancing Developer Guide*.

Request Syntax

```
response = client.create_load_balancer_listeners(
    LoadBalancerName='string',
    Listeners=[
        {
            'Protocol': 'string',
            'LoadBalancerPort': 123,
            'InstanceProtocol': 'string',
            'InstancePort': 123,
            'SSLCertificateId': 'string'
        },
    ],
```

```
]
)
```

Parameters

- **LoadBalancerName** (*string*) – [REQUIRED]

The name of the load balancer.

- **Listeners** (*list*) – [REQUIRED]

The listeners.

– (*dict*) –

Information about a listener.

For information about the protocols and the ports supported by Elastic Load Balancing, see [Listener Configurations for Elastic Load Balancing](#) in the *Elastic Load Balancing Developer Guide*.

- * **Protocol** (*string*) – [REQUIRED]

The load balancer transport protocol to use for routing: HTTP, HTTPS, TCP, or SSL.

- * **LoadBalancerPort** (*integer*) – [REQUIRED]

The port on which the load balancer is listening. The supported ports are: 25, 80, 443, 465, 587, and 1024-65535.

- * **InstanceProtocol** (*string*) –

The protocol to use for routing traffic to back-end instances: HTTP, HTTPS, TCP, or SSL.

If the front-end protocol is HTTP, HTTPS, TCP, or SSL, `InstanceProtocol` must be at the same protocol.

If there is another listener with the same `InstancePort` whose `InstanceProtocol` is secure, (HTTPS or SSL), the listener's `InstanceProtocol` must also be secure.

If there is another listener with the same `InstancePort` whose `InstanceProtocol` is HTTP or TCP, the listener's `InstanceProtocol` must be HTTP or TCP.

- * **InstancePort** (*integer*) – [REQUIRED]

The port on which the instance is listening.

- * **SSLCertificateId** (*string*) –

The Amazon Resource Name (ARN) of the server certificate.

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –

create_load_balancer_policy (**kwargs)

Creates a policy with the specified attributes for the specified load balancer.

Policies are settings that are saved for your load balancer and that can be applied to the front-end listener or the back-end application server, depending on the policy type.

Request Syntax

```
response = client.create_load_balancer_policy(
    LoadBalancerName='string',
    PolicyName='string',
    PolicyTypeName='string',
    PolicyAttributes=[
        {
            'AttributeName': 'string',
            'AttributeValue': 'string'
        },
    ]
)
```

Parameters

- **LoadBalancerName** (*string*) – **[REQUIRED]**
The name of the load balancer.
- **PolicyName** (*string*) – **[REQUIRED]**
The name of the load balancer policy to be created. This name must be unique within the set of policies for this load balancer.
- **PolicyTypeName** (*string*) – **[REQUIRED]**
The name of the base policy type. To get the list of policy types, use `DescribeLoadBalancerPolicyTypes`.
- **PolicyAttributes** (*list*) – The attributes for the policy.
 - (*dict*) –
Information about a policy attribute.
 - * **AttributeName** (*string*) –
The name of the attribute.
 - * **AttributeValue** (*string*) –
The value of the attribute.

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –

delete_load_balancer (***kwargs*)

Deletes the specified load balancer.

If you are attempting to recreate a load balancer, you must reconfigure all settings. The DNS name associated with a deleted load balancer are no longer usable. The name and associated DNS record of the deleted load balancer no longer exist and traffic sent to any of its IP addresses is no longer delivered to back-end instances.

If the load balancer does not exist or has already been deleted, the call to `DeleteLoadBalancer` still succeeds.

Request Syntax

```
response = client.delete_load_balancer(
    LoadBalancerName='string'
)
```

Parameters **LoadBalancerName** (*string*) – [REQUIRED]

The name of the load balancer.

Return type dict

Returns

Response Syntax

```
{}
```

Response Structure

- (*dict*) –

delete_load_balancer_listeners (***kwargs*)

Deletes the specified listeners from the specified load balancer.

Request Syntax

```
response = client.delete_load_balancer_listeners(
    LoadBalancerName='string',
    LoadBalancerPorts=[
        123,
    ]
)
```

Parameters

- **LoadBalancerName** (*string*) – [REQUIRED]

The name of the load balancer.

- **LoadBalancerPorts** (*list*) – [REQUIRED]

The client port numbers of the listeners.

– (*integer*) –

Return type dict

Returns

Response Syntax

```
{}
```

Response Structure

- (*dict*) –

delete_load_balancer_policy (***kwargs*)

Deletes the specified policy from the specified load balancer. This policy must not be enabled for any listeners.

Request Syntax

```
response = client.delete_load_balancer_policy(
    LoadBalancerName='string',
    PolicyName='string'
)
```

Parameters

- **LoadBalancerName** (*string*) – [REQUIRED]

The name of the load balancer.

- **PolicyName** (*string*) – [REQUIRED]

The name of the policy.

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –

deregister_instances_from_load_balancer (**kwargs)

Deregisters the specified instances from the specified load balancer. After the instance is deregistered, it no longer receives traffic from the load balancer.

You can use DescribeLoadBalancers to verify that the instance is deregistered from the load balancer.

For more information, see [Deregister and Register Amazon EC2 Instances](#) in the *Elastic Load Balancing Developer Guide* .

Request Syntax

```
response = client.deregister_instances_from_load_balancer(  
    LoadBalancerName='string',  
    Instances=[  
        {  
            'InstanceId': 'string'  
        },  
    ]  
)
```

Parameters

- **LoadBalancerName** (*string*) – [REQUIRED]

The name of the load balancer.

- **Instances** (*list*) – [REQUIRED]

The IDs of the instances.

– (*dict*) –

The ID of a back-end instance.

* **InstanceId** (*string*) –

The ID of the instance.

Return type dict

Returns

Response Syntax

```
{  
    'Instances': [  
        {  
            'InstanceId': 'string'  
        },  
    ]  
}
```

Response Structure

- *(dict)* –
 - **Instances** (*list*) –
 - The remaining instances registered with the load balancer.
 - * *(dict)* –
 - The ID of a back-end instance.
 - **InstanceId** (*string*) –
 - The ID of the instance.

describe_instance_health (***kwargs*)

Describes the state of the specified instances registered with the specified load balancer. If no instances are specified, the call describes the state of all instances registered with the load balancer, not including any terminated instances.

Request Syntax

```
response = client.describe_instance_health(
    LoadBalancerName='string',
    Instances=[
        {
            'InstanceId': 'string'
        },
    ]
)
```

Parameters

- **LoadBalancerName** (*string*) – [REQUIRED]
 - The name of the load balancer.
- **Instances** (*list*) – The IDs of the instances.
 - *(dict)* –
 - The ID of a back-end instance.
 - * **InstanceId** (*string*) –
 - The ID of the instance.

Return type dict

Returns**Response Syntax**

```
{
    'InstanceStates': [
        {
            'InstanceId': 'string',
            'State': 'string',
            'ReasonCode': 'string',
            'Description': 'string'
        },
    ]
}
```

Response Structure

- *(dict)* –
 - **InstanceStates** (*list*) –
 - Information about the health of the instances.

* (*dict*) –

Information about the state of a back-end instance.

- **InstanceId** (*string*) –

The ID of the instance.

- **State** (*string*) –

The current state of the instance.

Valid values: InService | OutOfService | Unknown

- **ReasonCode** (*string*) –

Information about the cause of OutOfService instances. Specifically, whether the cause is Elastic Load Balancing or the instance.

Valid values: ELB | Instance | N/A

- **Description** (*string*) –

A description of the instance state. This string can contain one or more of the following messages.

- N/A
- A transient error occurred. Please try again later.
- Instance has failed at least the UnhealthyThreshold number of health checks consecutively.
- Instance has not passed the configured HealthyThreshold number of health checks consecutively.
- Instance registration is still in progress.
- Instance is in the EC2 Availability Zone for which LoadBalancer is not configured to route traffic to.
- Instance is not currently registered with the LoadBalancer.
- Instance deregistration currently in progress.
- Disable Availability Zone is currently in progress.
- Instance is in pending state.
- Instance is in stopped state.
- Instance is in terminated state.

describe_load_balancer_attributes (***kwargs*)

Describes the attributes for the specified load balancer.

Request Syntax

```
response = client.describe_load_balancer_attributes(  
    LoadBalancerName='string'  
)
```

Parameters **LoadBalancerName** (*string*) – [REQUIRED]

The name of the load balancer.

Return type dict

Returns

Response Syntax

```
{
  'LoadBalancerAttributes': {
    'CrossZoneLoadBalancing': {
      'Enabled': True|False
    },
    'AccessLog': {
      'Enabled': True|False,
      'S3BucketName': 'string',
      'EmitInterval': 123,
      'S3BucketPrefix': 'string'
    },
    'ConnectionDraining': {
      'Enabled': True|False,
      'Timeout': 123
    },
    'ConnectionSettings': {
      'IdleTimeout': 123
    },
    'AdditionalAttributes': [
      {
        'Key': 'string',
        'Value': 'string'
      },
    ],
  ]
}
```

Response Structure

- (*dict*) –

- **LoadBalancerAttributes** (*dict*) –

Information about the load balancer attributes.

- **CrossZoneLoadBalancing** (*dict*) –

If enabled, the load balancer routes the request traffic evenly across all back-end instances regardless of the Availability Zones.

For more information, see [Enable Cross-Zone Load Balancing](#) in the *Elastic Load Balancing Developer Guide*.

- **Enabled** (*boolean*) –

Specifies whether cross-zone load balancing is enabled for the load balancer.

- **AccessLog** (*dict*) –

If enabled, the load balancer captures detailed information of all requests and delivers the information to the Amazon S3 bucket that you specify.

For more information, see [Enable Access Logs](#) in the *Elastic Load Balancing Developer Guide*.

- **Enabled** (*boolean*) –

Specifies whether access log is enabled for the load balancer.

- **S3BucketName** (*string*) –

The name of the Amazon S3 bucket where the access logs are stored.

- **EmitInterval** (*integer*) –

The interval for publishing the access logs. You can specify an interval of either 5 minutes or 60 minutes.

Default: 60 minutes

- **S3BucketPrefix** (*string*) –

The logical hierarchy you created for your Amazon S3 bucket, for example `my-bucket-prefix/prod` . If the prefix is not provided, the log is placed at the root level of the bucket.

- * **ConnectionDraining** (*dict*) –

If enabled, the load balancer allows existing requests to complete before the load balancer shifts traffic away from a deregistered or unhealthy back-end instance.

For more information, see [Enable Connection Draining](#) in the *Elastic Load Balancing Developer Guide* .

- **Enabled** (*boolean*) –

Specifies whether connection draining is enabled for the load balancer.

- **Timeout** (*integer*) –

The maximum time, in seconds, to keep the existing connections open before deregistering the instances.

- * **ConnectionSettings** (*dict*) –

If enabled, the load balancer allows the connections to remain idle (no data is sent over the connection) for the specified duration.

By default, Elastic Load Balancing maintains a 60-second idle connection timeout for both front-end and back-end connections of your load balancer. For more information, see [Configure Idle Connection Timeout](#) in the *Elastic Load Balancing Developer Guide* .

- **IdleTimeout** (*integer*) –

The time, in seconds, that the connection is allowed to be idle (no data has been sent over the connection) before it is closed by the load balancer.

- * **AdditionalAttributes** (*list*) –

This parameter is reserved.

- (*dict*) –

This data type is reserved.

- **Key** (*string*) –

This parameter is reserved.

- **Value** (*string*) –

This parameter is reserved.

describe_load_balancer_policies (***kwargs*)

Describes the specified policies.

If you specify a load balancer name, the action returns the descriptions of all policies created for the load balancer. If you specify a policy name associated with your load balancer, the action returns the description of that policy. If you don't specify a load balancer name, the action returns descriptions of the

specified sample policies, or descriptions of all sample policies. The names of the sample policies have the `ELBSample-` prefix.

Request Syntax

```
response = client.describe_load_balancer_policies(
    LoadBalancerName='string',
    PolicyNames=[
        'string',
    ]
)
```

Parameters

- **LoadBalancerName** (*string*) – The name of the load balancer.
- **PolicyNames** (*list*) – The names of the policies.
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'PolicyDescriptions': [
        {
            'PolicyName': 'string',
            'PolicyTypeName': 'string',
            'PolicyAttributeDescriptions': [
                {
                    'AttributeName': 'string',
                    'AttributeValue': 'string'
                },
            ],
        },
    ]
}
```

Response Structure

- (*dict*) –
 - **PolicyDescriptions** (*list*) –
 - Information about the policies.
 - * (*dict*) –
 - Information about a policy.
 - **PolicyName** (*string*) –
 - The name of the policy.
 - **PolicyTypeName** (*string*) –
 - The name of the policy type.
 - **PolicyAttributeDescriptions** (*list*) –
 - The policy attributes.
 - (*dict*) –
 - Information about a policy attribute.
 - **AttributeName** (*string*) –
 - The name of the attribute.

- **AttributeValue** (*string*) –

The value of the attribute.

describe_load_balancer_policy_types (***kwargs*)

Describes the specified load balancer policy types.

You can use these policy types with `CreateLoadBalancerPolicy` to create policy configurations for a load balancer.

Request Syntax

```
response = client.describe_load_balancer_policy_types(
    PolicyTypeNames=[
        'string',
    ]
)
```

Parameters **PolicyTypeNames** (*list*) – The names of the policy types. If no names are specified, describes all policy types defined by Elastic Load Balancing.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'PolicyTypeDescriptions': [
        {
            'PolicyTypeName': 'string',
            'Description': 'string',
            'PolicyAttributeTypeDescriptions': [
                {
                    'AttributeName': 'string',
                    'AttributeType': 'string',
                    'Description': 'string',
                    'DefaultValue': 'string',
                    'Cardinality': 'string'
                },
            ],
        },
    ],
}
```

Response Structure

- (*dict*) –

– **PolicyTypeDescriptions** (*list*) –

Information about the policy types.

- * (*dict*) –

Information about a policy type.

- **PolicyTypeName** (*string*) –

The name of the policy type.

- **Description** (*string*) –

A description of the policy type.

- **PolicyAttributeTypeDescriptions** (*list*) –

The description of the policy attributes associated with the policies defined by Elastic Load Balancing.

- *(dict)* –

Information about a policy attribute type.

- **AttributeName** (*string*) –

The name of the attribute.

- **AttributeType** (*string*) –

The type of the attribute. For example, `Boolean` or `Integer`.

- **Description** (*string*) –

A description of the attribute.

- **DefaultValue** (*string*) –

The default value of the attribute, if applicable.

- **Cardinality** (*string*) –

The cardinality of the attribute.

Valid values:

- `ONE(1)` : Single value required
- `ZERO_OR_ONE(0..1)` : Up to one value can be supplied
- `ZERO_OR_MORE(0..*)` : Optional. Multiple values are allowed
- `ONE_OR_MORE(1..*0)` : Required. Multiple values are allowed

describe_load_balancers (**kwargs)

Describes the specified the load balancers. If no load balancers are specified, the call describes all of your load balancers.

Request Syntax

```
response = client.describe_load_balancers(
    LoadBalancerNames=[
        'string',
    ],
    Marker='string',
    PageSize=123
)
```

Parameters

- **LoadBalancerNames** (*list*) – The names of the load balancers.
– (*string*) –
- **Marker** (*string*) – The marker for the next set of results. (You received this marker from a previous call.)
- **PageSize** (*integer*) – The maximum number of results to return with this call (a number from 1 to 400). The default is 400.

Return type dict

Returns

Response Syntax

```
{
    'LoadBalancerDescriptions': [
        {
```

```
'LoadBalancerName': 'string',
'DNSName': 'string',
'CanonicalHostedZoneName': 'string',
'CanonicalHostedZoneNameID': 'string',
'ListenerDescriptions': [
    {
        'Listener': {
            'Protocol': 'string',
            'LoadBalancerPort': 123,
            'InstanceProtocol': 'string',
            'InstancePort': 123,
            'SSLCertificateId': 'string'
        },
        'PolicyNames': [
            'string',
        ]
    },
],
'Policies': {
    'AppCookieStickinessPolicies': [
        {
            'PolicyName': 'string',
            'CookieName': 'string'
        },
    ],
    'LBCookieStickinessPolicies': [
        {
            'PolicyName': 'string',
            'CookieExpirationPeriod': 123
        },
    ],
    'OtherPolicies': [
        'string',
    ]
},
'BackendServerDescriptions': [
    {
        'InstancePort': 123,
        'PolicyNames': [
            'string',
        ]
    },
],
'AvailabilityZones': [
    'string',
],
'Subnets': [
    'string',
],
'VPCId': 'string',
'Instances': [
    {
        'InstanceId': 'string'
    },
],
'HealthCheck': {
    'Target': 'string',
    'Interval': 123,
```

```

        'Timeout': 123,
        'UnhealthyThreshold': 123,
        'HealthyThreshold': 123
    },
    'SourceSecurityGroup': {
        'OwnerAlias': 'string',
        'GroupName': 'string'
    },
    'SecurityGroups': [
        'string',
    ],
    'CreatedTime': datetime(2015, 1, 1),
    'Scheme': 'string'
},
],
'NextMarker': 'string'
}

```

Response Structure

- (*dict*) –

- **LoadBalancerDescriptions** (*list*) –

Information about the load balancers.

- * (*dict*) –

Information about a load balancer.

- **LoadBalancerName** (*string*) –

The name of the load balancer.

- **DNSName** (*string*) –

The external DNS name of the load balancer.

- **CanonicalHostedZoneName** (*string*) –

The Amazon Route 53 hosted zone associated with the load balancer.

For more information, see [Using Domain Names With Elastic Load Balancing](#) in the *Elastic Load Balancing Developer Guide*.

- **CanonicalHostedZoneNameID** (*string*) –

The ID of the Amazon Route 53 hosted zone name associated with the load balancer.

- **ListenerDescriptions** (*list*) –

The listeners for the load balancer.

- (*dict*) –

The policies enabled for a listener.

- **Listener** (*dict*) –

Information about a listener.

For information about the protocols and the ports supported by Elastic Load Balancing, see [Listener Configurations for Elastic Load Balancing](#) in the *Elastic Load Balancing Developer Guide*.

- **Protocol** (*string*) –

The load balancer transport protocol to use for routing: HTTP, HTTPS, TCP, or SSL.

- **LoadBalancerPort** (*integer*) –

The port on which the load balancer is listening. The supported ports are: 25, 80, 443, 465, 587, and 1024-65535.

- **InstanceProtocol** (*string*) –

The protocol to use for routing traffic to back-end instances: HTTP, HTTPS, TCP, or SSL.

If the front-end protocol is HTTP, HTTPS, TCP, or SSL, `InstanceProtocol` must be at the same protocol.

If there is another listener with the same `InstancePort` whose `InstanceProtocol` is secure, (HTTPS or SSL), the listener's `InstanceProtocol` must also be secure.

If there is another listener with the same `InstancePort` whose `InstanceProtocol` is HTTP or TCP, the listener's `InstanceProtocol` must be HTTP or TCP.

- **InstancePort** (*integer*) –

The port on which the instance is listening.

- **SSLCertificateId** (*string*) –

The Amazon Resource Name (ARN) of the server certificate.

- **PolicyNames** (*list*) –

The policies. If there are no policies enabled, the list is empty.

- (*string*) –

- **Policies** (*dict*) –

The policies defined for the load balancer.

- **AppCookieStickinessPolicies** (*list*) –

The stickiness policies created using `CreateAppCookieStickinessPolicy`.

- (*dict*) –

Information about a policy for application-controlled session stickiness.

- **PolicyName** (*string*) –

The mnemonic name for the policy being created. The name must be unique within a set of policies for this load balancer.

- **CookieName** (*string*) –

The name of the application cookie used for stickiness.

- **LBCookieStickinessPolicies** (*list*) –

The stickiness policies created using `CreateLBCookieStickinessPolicy`.

- (*dict*) –

Information about a policy for duration-based session stickiness.

- **PolicyName** (*string*) –

The name for the policy being created. The name must be unique within the set of policies for this load balancer.

- **CookieExpirationPeriod** (*integer*) –
The time period, in seconds, after which the cookie should be considered stale. If this parameter is not specified, the stickiness session lasts for the duration of the browser session.
- **OtherPolicies** (*list*) –
The policies other than the stickiness policies.
- (*string*) –
- **BackendServerDescriptions** (*list*) –
Information about the back-end servers.
- (*dict*) –
Information about the configuration of a back-end server.
- **InstancePort** (*integer*) –
The port on which the back-end server is listening.
- **PolicyNames** (*list*) –
The names of the policies enabled for the back-end server.
- (*string*) –
- **AvailabilityZones** (*list*) –
The Availability Zones for the load balancer.
- (*string*) –
- **Subnets** (*list*) –
The IDs of the subnets for the load balancer.
- (*string*) –
- **VPCId** (*string*) –
The ID of the VPC for the load balancer.
- **Instances** (*list*) –
The IDs of the instances for the load balancer.
- (*dict*) –
The ID of a back-end instance.
- **InstanceId** (*string*) –
The ID of the instance.
- **HealthCheck** (*dict*) –
Information about the health checks conducted on the load balancer.
- **Target** (*string*) –
The instance being checked. The protocol is either TCP, HTTP, HTTPS, or SSL. The range of valid ports is one (1) through 65535.

TCP is the default, specified as a TCP: port pair, for example “TCP:5000”. In this case, a health check simply attempts to open a TCP connection to the instance on the specified port. Failure to connect within the configured timeout is considered unhealthy.

SSL is also specified as SSL: port pair, for example, SSL:5000.

For HTTP/HTTPS, you must include a ping path in the string. HTTP is specified as a `HTTP:port;/PathToPing`; grouping, for example `"HTTP:80/weather/us/wa/seattle"`. In this case, a HTTP GET request is issued to the instance on the given port and path. Any answer other than `"200 OK"` within the timeout period is considered unhealthy.

The total length of the HTTP ping target must be 1024 16-bit Unicode characters or less.

- **Interval** (*integer*) –

The approximate interval, in seconds, between health checks of an individual instance.

- **Timeout** (*integer*) –

The amount of time, in seconds, during which no response means a failed health check.

This value must be less than the `Interval` value.

- **UnhealthyThreshold** (*integer*) –

The number of consecutive health check failures required before moving the instance to the `Unhealthy` state.

- **HealthyThreshold** (*integer*) –

The number of consecutive health checks successes required before moving the instance to the `Healthy` state.

- **SourceSecurityGroup** (*dict*) –

The security group that you can use as part of your inbound rules for your load balancer's back-end application instances. To only allow traffic from load balancers, add a security group rule to your back end instance that specifies this source security group as the inbound source.

- **OwnerAlias** (*string*) –

The owner of the security group.

- **GroupName** (*string*) –

The name of the security group.

- **SecurityGroups** (*list*) –

The security groups for the load balancer. Valid only for load balancers in a VPC.

- (*string*) –

- **CreatedTime** (*datetime*) –

The date and time the load balancer was created.

- **Scheme** (*string*) –

The type of load balancer. Valid only for load balancers in a VPC.

If `Scheme` is `internet-facing`, the load balancer has a public DNS name that resolves to a public IP address.

If `Scheme` is `internal`, the load balancer has a public DNS name that resolves to a private IP address.

- **NextMarker** (*string*) –

The marker to use when requesting the next set of results. If there are no additional results, the string is empty.

describe_tags (**kwargs)

Describes the tags associated with the specified load balancers.

Request Syntax

```
response = client.describe_tags(
    LoadBalancerNames=[
        'string',
    ]
)
```

Parameters **LoadBalancerNames** (*list*) – [REQUIRED]

The names of the load balancers.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'TagDescriptions': [
        {
            'LoadBalancerName': 'string',
            'Tags': [
                {
                    'Key': 'string',
                    'Value': 'string'
                },
            ],
        },
    ]
}
```

Response Structure

- (*dict*) –
 - **TagDescriptions** (*list*) –

Information about the tags.

- * (*dict*) –

The tags associated with a load balancer.

- **LoadBalancerName** (*string*) –

The name of the load balancer.

- **Tags** (*list*) –

The tags.

- (*dict*) –

Information about a tag.

- **Key** (*string*) –

The key of the tag.

- **Value** (*string*) –

The value of the tag.

detach_load_balancer_from_subnets (***kwargs*)

Removes the specified subnets from the set of configured subnets for the load balancer.

After a subnet is removed, all EC2 instances registered with the load balancer in the removed subnet go into the `OutOfService` state. Then, the load balancer balances the traffic among the remaining routable subnets.

Request Syntax

```
response = client.detach_load_balancer_from_subnets(
    LoadBalancerName='string',
    Subnets=[
        'string',
    ]
)
```

Parameters

- **LoadBalancerName** (*string*) – [REQUIRED]

The name of the load balancer.

- **Subnets** (*list*) – [REQUIRED]

The IDs of the subnets.

– (*string*) –

Return type dict

Returns**Response Syntax**

```
{
    'Subnets': [
        'string',
    ]
}
```

Response Structure

- (*dict*) –
 - **Subnets** (*list*) –

The IDs of the remaining subnets for the load balancer.

* (*string*) –

disable_availability_zones_for_load_balancer (***kwargs*)

Removes the specified Availability Zones from the set of Availability Zones for the specified load balancer.

There must be at least one Availability Zone registered with a load balancer at all times. After an Availability Zone is removed, all instances registered with the load balancer that are in the removed Availability Zone go into the `OutOfService` state. Then, the load balancer attempts to equally balance the traffic among its remaining Availability Zones.

For more information, see [Disable an Availability Zone from a Load-Balanced Application](#) in the *Elastic Load Balancing Developer Guide* .

Request Syntax

```
response = client.disable_availability_zones_for_load_balancer(
    LoadBalancerName='string',
    AvailabilityZones=[
        'string',
    ]
)
```

```
]
)
```

Parameters

- **LoadBalancerName** (*string*) – [REQUIRED]
The name of the load balancer.
- **AvailabilityZones** (*list*) – [REQUIRED]
The Availability Zones.
– (*string*) –

Return type dict**Returns****Response Syntax**

```
{
  'AvailabilityZones': [
    'string',
  ]
}
```

Response Structure

- (*dict*) –
 - **AvailabilityZones** (*list*) –
The remaining Availability Zones for the load balancer.
* (*string*) –

enable_availability_zones_for_load_balancer (***kwargs*)

Adds the specified Availability Zones to the set of Availability Zones for the specified load balancer.

The load balancer evenly distributes requests across all its registered Availability Zones that contain instances.

For more information, see [Add Availability Zone](#) in the *Elastic Load Balancing Developer Guide* .**Request Syntax**

```
response = client.enable_availability_zones_for_load_balancer(
    LoadBalancerName='string',
    AvailabilityZones=[
        'string',
    ]
)
```

Parameters

- **LoadBalancerName** (*string*) – [REQUIRED]
The name of the load balancer.
- **AvailabilityZones** (*list*) – [REQUIRED]
The Availability Zones. These must be in the same region as the load balancer.
– (*string*) –

Return type dict**Returns****Response Syntax**

```
{
    'AvailabilityZones': [
        'string',
    ]
}
```

Response Structure

- *(dict)* –
 - **AvailabilityZones** (*list*) –
The updated list of Availability Zones for the load balancer.
 - *(string)* –

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_waiter (*waiter_name*)

modify_load_balancer_attributes (***kwargs*)

Modifies the attributes of the specified load balancer.

You can modify the load balancer attributes, such as `AccessLogs`, `ConnectionDraining`, and `CrossZoneLoadBalancing` by either enabling or disabling them. Or, you can modify the load balancer attribute `ConnectionSettings` by specifying an idle connection timeout value for your load balancer.

For more information, see the following in the *Elastic Load Balancing Developer Guide* :

- [Cross-Zone Load Balancing](#)
- [Connection Draining](#)
- [Access Logs](#)
- [Idle Connection Timeout](#)

Request Syntax

```
response = client.modify_load_balancer_attributes(
    LoadBalancerName='string',
    LoadBalancerAttributes={
        'CrossZoneLoadBalancing': {
            'Enabled': True|False
```

```

    },
    'AccessLog': {
        'Enabled': True|False,
        'S3BucketName': 'string',
        'EmitInterval': 123,
        'S3BucketPrefix': 'string'
    },
    'ConnectionDraining': {
        'Enabled': True|False,
        'Timeout': 123
    },
    'ConnectionSettings': {
        'IdleTimeout': 123
    },
    'AdditionalAttributes': [
        {
            'Key': 'string',
            'Value': 'string'
        },
        ...
    ]
}
)

```

Parameters

- **LoadBalancerName** (*string*) – [REQUIRED]

The name of the load balancer.

- **LoadBalancerAttributes** (*dict*) – [REQUIRED]

The attributes of the load balancer.

- **CrossZoneLoadBalancing** (*dict*) –

If enabled, the load balancer routes the request traffic evenly across all back-end instances regardless of the Availability Zones.

For more information, see [Enable Cross-Zone Load Balancing](#) in the *Elastic Load Balancing Developer Guide*.

- * **Enabled** (*boolean*) – [REQUIRED]

Specifies whether cross-zone load balancing is enabled for the load balancer.

- **AccessLog** (*dict*) –

If enabled, the load balancer captures detailed information of all requests and delivers the information to the Amazon S3 bucket that you specify.

For more information, see [Enable Access Logs](#) in the *Elastic Load Balancing Developer Guide*.

- * **Enabled** (*boolean*) – [REQUIRED]

Specifies whether access log is enabled for the load balancer.

- * **S3BucketName** (*string*) –

The name of the Amazon S3 bucket where the access logs are stored.

- * **EmitInterval** (*integer*) –

The interval for publishing the access logs. You can specify an interval of either 5 minutes or 60 minutes.

Default: 60 minutes

- * **S3BucketPrefix** (*string*) –

The logical hierarchy you created for your Amazon S3 bucket, for example `my-bucket-prefix/prod`. If the prefix is not provided, the log is placed at the root level of the bucket.

- **ConnectionDraining** (*dict*) –

If enabled, the load balancer allows existing requests to complete before the load balancer shifts traffic away from a deregistered or unhealthy back-end instance.

For more information, see [Enable Connection Draining](#) in the *Elastic Load Balancing Developer Guide*.

- * **Enabled** (*boolean*) – **[REQUIRED]**

Specifies whether connection draining is enabled for the load balancer.

- * **Timeout** (*integer*) –

The maximum time, in seconds, to keep the existing connections open before deregistering the instances.

- **ConnectionSettings** (*dict*) –

If enabled, the load balancer allows the connections to remain idle (no data is sent over the connection) for the specified duration.

By default, Elastic Load Balancing maintains a 60-second idle connection timeout for both front-end and back-end connections of your load balancer. For more information, see [Configure Idle Connection Timeout](#) in the *Elastic Load Balancing Developer Guide*.

- * **IdleTimeout** (*integer*) – **[REQUIRED]**

The time, in seconds, that the connection is allowed to be idle (no data has been sent over the connection) before it is closed by the load balancer.

- **AdditionalAttributes** (*list*) –

This parameter is reserved.

- * (*dict*) –

This data type is reserved.

- **Key** (*string*) –

This parameter is reserved.

- **Value** (*string*) –

This parameter is reserved.

Return type dict

Returns

Response Syntax

```
{
  'LoadBalancerName': 'string',
  'LoadBalancerAttributes': {
    'CrossZoneLoadBalancing': {
      'Enabled': True|False
    },
    'AccessLog': {
      'Enabled': True|False,
      'S3BucketName': 'string',
```



```

        'EmitInterval': 123,
        'S3BucketPrefix': 'string'
    },
    'ConnectionDraining': {
        'Enabled': True|False,
        'Timeout': 123
    },
    'ConnectionSettings': {
        'IdleTimeout': 123
    },
    'AdditionalAttributes': [
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
]
}

```

Response Structure

- *(dict)* –
 - **LoadBalancerName** (*string*) –

The name of the load balancer.
 - **LoadBalancerAttributes** (*dict*) –

The attributes for a load balancer.

 - * **CrossZoneLoadBalancing** (*dict*) –

If enabled, the load balancer routes the request traffic evenly across all back-end instances regardless of the Availability Zones.

For more information, see [Enable Cross-Zone Load Balancing](#) in the *Elastic Load Balancing Developer Guide* .

 - **Enabled** (*boolean*) –

Specifies whether cross-zone load balancing is enabled for the load balancer.
 - * **AccessLog** (*dict*) –

If enabled, the load balancer captures detailed information of all requests and delivers the information to the Amazon S3 bucket that you specify.

For more information, see [Enable Access Logs](#) in the *Elastic Load Balancing Developer Guide* .

 - **Enabled** (*boolean*) –

Specifies whether access log is enabled for the load balancer.
 - **S3BucketName** (*string*) –

The name of the Amazon S3 bucket where the access logs are stored.
 - **EmitInterval** (*integer*) –

The interval for publishing the access logs. You can specify an interval of either 5 minutes or 60 minutes.

Default: 60 minutes
 - **S3BucketPrefix** (*string*) –

The logical hierarchy you created for your Amazon S3 bucket, for example `my-bucket-prefix/prod`. If the prefix is not provided, the log is placed at the root level of the bucket.

* **ConnectionDraining** (*dict*) –

If enabled, the load balancer allows existing requests to complete before the load balancer shifts traffic away from a deregistered or unhealthy back-end instance.

For more information, see [Enable Connection Draining](#) in the *Elastic Load Balancing Developer Guide*.

· **Enabled** (*boolean*) –

Specifies whether connection draining is enabled for the load balancer.

· **Timeout** (*integer*) –

The maximum time, in seconds, to keep the existing connections open before deregistering the instances.

* **ConnectionSettings** (*dict*) –

If enabled, the load balancer allows the connections to remain idle (no data is sent over the connection) for the specified duration.

By default, Elastic Load Balancing maintains a 60-second idle connection timeout for both front-end and back-end connections of your load balancer. For more information, see [Configure Idle Connection Timeout](#) in the *Elastic Load Balancing Developer Guide*.

· **IdleTimeout** (*integer*) –

The time, in seconds, that the connection is allowed to be idle (no data has been sent over the connection) before it is closed by the load balancer.

* **AdditionalAttributes** (*list*) –

This parameter is reserved.

· (*dict*) –

This data type is reserved.

· **Key** (*string*) –

This parameter is reserved.

· **Value** (*string*) –

This parameter is reserved.

register_instances_with_load_balancer (***kwargs*)

Adds the specified instances to the specified load balancer.

The instance must be a running instance in the same network as the load balancer (EC2-Classic or the same VPC). If you have EC2-Classic instances and a load balancer in a VPC with ClassicLink enabled, you can link the EC2-Classic instances to that VPC and then register the linked EC2-Classic instances with the load balancer in the VPC.

Note that `RegisterInstanceWithLoadBalancer` completes when the request has been registered. Instance registration happens shortly afterwards. To check the state of the registered instances, use `DescribeLoadBalancers` or `DescribeInstanceHealth`.

After the instance is registered, it starts receiving traffic and requests from the load balancer. Any instance that is not in one of the Availability Zones registered for the load balancer is moved to the

OutOfService state. If an Availability Zone is added to the load balancer later, any instances registered with the load balancer move to the InService state.

If you stop an instance registered with a load balancer and then start it, the IP addresses associated with the instance changes. Elastic Load Balancing cannot recognize the new IP address, which prevents it from routing traffic to the instances. We recommend that you use the following sequence: stop the instance, deregister the instance, start the instance, and then register the instance. To deregister instances from a load balancer, use `DeregisterInstancesFromLoadBalancer`.

For more information, see [Deregister and Register EC2 Instances](#) in the *Elastic Load Balancing Developer Guide*.

Request Syntax

```
response = client.register_instances_with_load_balancer(
    LoadBalancerName='string',
    Instances=[
        {
            'InstanceId': 'string'
        },
    ]
)
```

Parameters

- **LoadBalancerName** (*string*) – [REQUIRED]

The name of the load balancer.

- **Instances** (*list*) – [REQUIRED]

The IDs of the instances.

– (*dict*) –

The ID of a back-end instance.

* **InstanceId** (*string*) –

The ID of the instance.

Return type dict

Returns

Response Syntax

```
{
    'Instances': [
        {
            'InstanceId': 'string'
        },
    ]
}
```

Response Structure

- (*dict*) –

– **Instances** (*list*) –

The updated list of instances for the load balancer.

* (*dict*) –

The ID of a back-end instance.

· **InstanceId** (*string*) –

The ID of the instance.

remove_tags (**kwargs)

Removes one or more tags from the specified load balancer.

Request Syntax

```
response = client.remove_tags(
    LoadBalancerNames=[
        'string',
    ],
    Tags=[
        {
            'Key': 'string'
        },
    ]
)
```

Parameters

- **LoadBalancerNames** (*list*) – [REQUIRED]

The name of the load balancer. You can specify a maximum of one load balancer name.

– (*string*) –

- **Tags** (*list*) – [REQUIRED]

The list of tag keys to remove.

– (*dict*) –

The key of a tag.

* **Key** (*string*) –

The name of the key.

Return type dict

Returns**Response Syntax**

```
{}
```

Response Structure

- (*dict*) –

set_load_balancer_listener_ssl_certificate (**kwargs)

Sets the certificate that terminates the specified listener’s SSL connections. The specified certificate replaces any prior certificate that was used on the same load balancer and port.

For more information about updating your SSL certificate, see [Updating an SSL Certificate for a Load Balancer](#) in the *Elastic Load Balancing Developer Guide* .

Request Syntax

```
response = client.set_load_balancer_listener_ssl_certificate(
    LoadBalancerName='string',
    LoadBalancerPort=123,
    SSLCertificateId='string'
)
```

Parameters

- **LoadBalancerName** (*string*) – [REQUIRED]

The name of the load balancer.

- **LoadBalancerPort** (*integer*) – [REQUIRED]

The port that uses the specified SSL certificate.

- **SSLCertificateId** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the SSL certificate.

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –

set_load_balancer_policies_for_backend_server (***kwargs*)

Replaces the set of policies associated with the specified port on which the back-end server is listening with a new set of policies. At this time, only the back-end server authentication policy type can be applied to the back-end ports; this policy type is composed of multiple public key policies.

Each time you use `SetLoadBalancerPoliciesForBackendServer` to enable the policies, use the `PolicyNames` parameter to list the policies that you want to enable.

You can use `DescribeLoadBalancers` or `DescribeLoadBalancerPolicies` to verify that the policy is associated with the back-end server.

Request Syntax

```
response = client.set_load_balancer_policies_for_backend_server(
    LoadBalancerName='string',
    InstancePort=123,
    PolicyNames=[
        'string',
    ]
)
```

Parameters

- **LoadBalancerName** (*string*) – [REQUIRED]

The name of the load balancer.

- **InstancePort** (*integer*) – [REQUIRED]

The port number associated with the back-end server.

- **PolicyNames** (*list*) – [REQUIRED]

The names of the policies. If the list is empty, then all current policies are removed from the back-end server.

– (*string*) –

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –

set_load_balancer_policies_of_listener (***kwargs*)

Associates, updates, or disables a policy with a listener for the specified load balancer. You can associate multiple policies with a listener.

Request Syntax

```
response = client.set_load_balancer_policies_of_listener(
    LoadBalancerName='string',
    LoadBalancerPort=123,
    PolicyNames=[
        'string',
    ]
)
```

Parameters

- **LoadBalancerName** (*string*) – **[REQUIRED]**
The name of the load balancer.
- **LoadBalancerPort** (*integer*) – **[REQUIRED]**
The external port of the load balancer for the policy.
- **PolicyNames** (*list*) – **[REQUIRED]**
The names of the policies. If the list is empty, the current policy is removed from the listener.
– (*string*) –

Return type dict

Returns**Response Syntax**

```
{ }
```

Response Structure

- (*dict*) –

Paginators

The available paginators are:

- `ElasticLoadBalancing.Paginator.DescribeLoadBalancers`

class ElasticLoadBalancing.Paginator.**DescribeLoadBalancers**

```
paginator = client.get_paginator('describe_load_balancers')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `ElasticLoadBalancing.Client.describe_load_balancers()`.

Request Syntax

```
response_iterator = paginator.paginate(
    LoadBalancerNames=[
        'string',
    ],
    PaginationConfig={
```

```

        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)

```

Parameters

- **LoadBalancerNames** (*list*) – The names of the load balancers.
 - (*string*) –
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```

{
    'LoadBalancerDescriptions': [
        {
            'LoadBalancerName': 'string',
            'DNSName': 'string',
            'CanonicalHostedZoneName': 'string',
            'CanonicalHostedZoneNameID': 'string',
            'ListenerDescriptions': [
                {
                    'Listener': {
                        'Protocol': 'string',
                        'LoadBalancerPort': 123,
                        'InstanceProtocol': 'string',
                        'InstancePort': 123,
                        'SSLCertificateId': 'string'
                    },
                    'PolicyNames': [
                        'string',
                    ]
                },
            ],
            'Policies': {
                'AppCookieStickinessPolicies': [
                    {
                        'PolicyName': 'string',
                        'CookieName': 'string'
                    },
                ],
                'LBCookieStickinessPolicies': [
                    {

```

```
        'PolicyName': 'string',
        'CookieExpirationPeriod': 123
    },
    ],
    'OtherPolicies': [
        'string',
    ]
},
'BackendServerDescriptions': [
    {
        'InstancePort': 123,
        'PolicyNames': [
            'string',
        ]
    },
],
'AvailabilityZones': [
    'string',
],
'Subnets': [
    'string',
],
'VPCId': 'string',
'Instances': [
    {
        'InstanceId': 'string'
    },
],
'HealthCheck': {
    'Target': 'string',
    'Interval': 123,
    'Timeout': 123,
    'UnhealthyThreshold': 123,
    'HealthyThreshold': 123
},
'SourceSecurityGroup': {
    'OwnerAlias': 'string',
    'GroupName': 'string'
},
'SecurityGroups': [
    'string',
],
'CreatedTime': datetime(2015, 1, 1),
'Scheme': 'string'
},
],
'NextToken': 'string'
}
```

Response Structure

- (dict) –
 - **LoadBalancerDescriptions** (list) –
Information about the load balancers.
 - * (dict) –
Information about a load balancer.
 - **LoadBalancerName** (string) –

The name of the load balancer.

- **DNSName** (*string*) –

The external DNS name of the load balancer.

- **CanonicalHostedZoneName** (*string*) –

The Amazon Route 53 hosted zone associated with the load balancer.

For more information, see [Using Domain Names With Elastic Load Balancing](#) in the *Elastic Load Balancing Developer Guide*.

- **CanonicalHostedZoneNameID** (*string*) –

The ID of the Amazon Route 53 hosted zone name associated with the load balancer.

- **ListenerDescriptions** (*list*) –

The listeners for the load balancer.

- (*dict*) –

The policies enabled for a listener.

- **Listener** (*dict*) –

Information about a listener.

For information about the protocols and the ports supported by Elastic Load Balancing, see [Listener Configurations for Elastic Load Balancing](#) in the *Elastic Load Balancing Developer Guide*.

- **Protocol** (*string*) –

The load balancer transport protocol to use for routing: HTTP, HTTPS, TCP, or SSL.

- **LoadBalancerPort** (*integer*) –

The port on which the load balancer is listening. The supported ports are: 25, 80, 443, 465, 587, and 1024-65535.

- **InstanceProtocol** (*string*) –

The protocol to use for routing traffic to back-end instances: HTTP, HTTPS, TCP, or SSL.

If the front-end protocol is HTTP, HTTPS, TCP, or SSL, `InstanceProtocol` must be at the same protocol.

If there is another listener with the same `InstancePort` whose `InstanceProtocol` is secure, (HTTPS or SSL), the listener's `InstanceProtocol` must also be secure.

If there is another listener with the same `InstancePort` whose `InstanceProtocol` is HTTP or TCP, the listener's `InstanceProtocol` must be HTTP or TCP.

- **InstancePort** (*integer*) –

The port on which the instance is listening.

- **SSLCertificateId** (*string*) –

The Amazon Resource Name (ARN) of the server certificate.

- **PolicyNames** (*list*) –

The policies. If there are no policies enabled, the list is empty.

- *(string)* –
- **Policies** (*dict*) –
The policies defined for the load balancer.
- **AppCookieStickinessPolicies** (*list*) –
The stickiness policies created using `CreateAppCookieStickinessPolicy`.
- (*dict*) –
Information about a policy for application-controlled session stickiness.
- **PolicyName** (*string*) –
The mnemonic name for the policy being created. The name must be unique within a set of policies for this load balancer.
- **CookieName** (*string*) –
The name of the application cookie used for stickiness.
- **LBCookieStickinessPolicies** (*list*) –
The stickiness policies created using `CreateLBCookieStickinessPolicy`.
- (*dict*) –
Information about a policy for duration-based session stickiness.
- **PolicyName** (*string*) –
The name for the policy being created. The name must be unique within the set of policies for this load balancer.
- **CookieExpirationPeriod** (*integer*) –
The time period, in seconds, after which the cookie should be considered stale. If this parameter is not specified, the stickiness session lasts for the duration of the browser session.
- **OtherPolicies** (*list*) –
The policies other than the stickiness policies.
- (*string*) –
- **BackendServerDescriptions** (*list*) –
Information about the back-end servers.
- (*dict*) –
Information about the configuration of a back-end server.
- **InstancePort** (*integer*) –
The port on which the back-end server is listening.
- **PolicyNames** (*list*) –
The names of the policies enabled for the back-end server.
- (*string*) –
- **AvailabilityZones** (*list*) –
The Availability Zones for the load balancer.
- (*string*) –
- **Subnets** (*list*) –
The IDs of the subnets for the load balancer.
- (*string*) –

- **VPCId** (*string*) –
The ID of the VPC for the load balancer.
- **Instances** (*list*) –
The IDs of the instances for the load balancer.
- (*dict*) –
The ID of a back-end instance.
- **InstanceId** (*string*) –
The ID of the instance.
- **HealthCheck** (*dict*) –
Information about the health checks conducted on the load balancer.
- **Target** (*string*) –
The instance being checked. The protocol is either TCP, HTTP, HTTPS, or SSL. The range of valid ports is one (1) through 65535.

TCP is the default, specified as a TCP: port pair, for example “TCP:5000”. In this case, a health check simply attempts to open a TCP connection to the instance on the specified port. Failure to connect within the configured timeout is considered unhealthy.

SSL is also specified as SSL: port pair, for example, SSL:5000.

For HTTP/HTTPS, you must include a ping path in the string. HTTP is specified as a HTTP:port;/;PathToPing; grouping, for example “HTTP:80/weather/us/wa/seattle”. In this case, a HTTP GET request is issued to the instance on the given port and path. Any answer other than “200 OK” within the timeout period is considered unhealthy.

The total length of the HTTP ping target must be 1024 16-bit Unicode characters or less.
- **Interval** (*integer*) –
The approximate interval, in seconds, between health checks of an individual instance.
- **Timeout** (*integer*) –
The amount of time, in seconds, during which no response means a failed health check.

This value must be less than the `Interval` value.
- **UnhealthyThreshold** (*integer*) –
The number of consecutive health check failures required before moving the instance to the `Unhealthy` state.
- **HealthyThreshold** (*integer*) –
The number of consecutive health checks successes required before moving the instance to the `Healthy` state.
- **SourceSecurityGroup** (*dict*) –
The security group that you can use as part of your inbound rules for your load balancer’s back-end application instances.

To only allow traffic from load balancers, add a security group rule to your back end instance that specifies this source security group as the inbound source.

- **OwnerAlias** (*string*) –

The owner of the security group.

- **GroupName** (*string*) –

The name of the security group.

- **SecurityGroups** (*list*) –

The security groups for the load balancer. Valid only for load balancers in a VPC.

- (*string*) –

- **CreatedTime** (*datetime*) –

The date and time the load balancer was created.

- **Scheme** (*string*) –

The type of load balancer. Valid only for load balancers in a VPC.

If Scheme is `internet-facing`, the load balancer has a public DNS name that resolves to a public IP address.

If Scheme is `internal`, the load balancer has a public DNS name that resolves to a private IP address.

- **NextToken** (*string*) –

A token to resume pagination.

EMR

Table of Contents

- EMR
 - Client
 - Paginators
 - Waiters

Client

class `EMR.Client`

A low-level client representing Amazon Elastic MapReduce (EMR):

```
import boto3
```

```
client = boto3.client('emr')
```

These are the available methods:

- `add_instance_groups()`
- `add_job_flow_steps()`
- `add_tags()`
- `can_paginate()`
- `describe_cluster()`

- `describe_job_flows()`
- `describe_step()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_waiter()`
- `list_bootstrap_actions()`
- `list_clusters()`
- `list_instance_groups()`
- `list_instances()`
- `list_steps()`
- `modify_instance_groups()`
- `remove_tags()`
- `run_job_flow()`
- `set_termination_protection()`
- `set_visible_to_all_users()`
- `terminate_job_flows()`

`add_instance_groups(**kwargs)`

AddInstanceGroups adds an instance group to a running cluster.

Request Syntax

```
response = client.add_instance_groups(
    InstanceGroups=[
        {
            'Name': 'string',
            'Market': 'ON_DEMAND'|'SPOT',
            'InstanceRole': 'MASTER'|'CORE'|'TASK',
            'BidPrice': 'string',
            'InstanceType': 'string',
            'InstanceCount': 123,
            'Configurations': [
                {
                    'Classification': 'string',
                    'Configurations': {'... recursive ...'},
                    'Properties': {
                        'string': 'string'
                    }
                }
            ],
        },
    ],
    JobFlowId='string'
)
```

Parameters

- **InstanceGroups** (*list*) – **[REQUIRED]**

Instance Groups to add.

– (*dict*) –

Configuration defining a new instance group.

* **Name** (*string*) –

Friendly name given to the instance group.

* **Market** (*string*) –

Market type of the Amazon EC2 instances used to create a cluster node.

- * **InstanceRole** (*string*) – [REQUIRED]

The role of the instance group in the cluster.

- * **BidPrice** (*string*) –

Bid price for each Amazon EC2 instance in the instance group when launching nodes as Spot Instances, expressed in USD.

- * **InstanceType** (*string*) – [REQUIRED]

The Amazon EC2 instance type for all instances in the instance group.

- * **InstanceCount** (*integer*) – [REQUIRED]

Target number of instances for the instance group.

- * **Configurations** (*list*) –

Note: Amazon EMR releases 4.x or later.

The list of configurations supplied for an EMR cluster instance group. You can specify a separate configuration for each instance group (master, core, and task).

- (*dict*) –

Note: Amazon EMR releases 4.x or later.

Specifies a hardware and software configuration of the EMR cluster. This includes configurations for applications and software bundled with Amazon EMR. The Configuration object is a JSON object which is defined by a classification and a set of properties. Configurations can be nested, so a configuration may have its own Configuration objects listed.

- **Classification** (*string*) –

The classification of a configuration. For more information see, [Amazon EMR Configurations](#) .

- **Configurations** (*list*) –

A list of configurations you apply to this configuration object.

- **Properties** (*dict*) –

A set of properties supplied to the Configuration object.

- (*string*) –

- (*string*) –

- **JobFlowId** (*string*) – [REQUIRED]

Job flow in which to add the instance groups.

Return type dict

Returns

Response Syntax

```
{
    'JobFlowId': 'string',
    'InstanceGroupIds': [
        'string',
    ]
}
```

Response Structure

- (*dict*) –

Output from an AddInstanceGroups call.

- **JobFlowId** (*string*) –

The job flow ID in which the instance groups are added.

- **InstanceGroupIds** (*list*) –

Instance group IDs of the newly created instance groups.

* (*string*) –

add_job_flow_steps (***kwargs*)

AddJobFlowSteps adds new steps to a running job flow. A maximum of 256 steps are allowed in each job flow.

If your job flow is long-running (such as a Hive data warehouse) or complex, you may require more than 256 steps to process your data. You can bypass the 256-step limitation in various ways, including using the SSH shell to connect to the master node and submitting queries directly to the software running on the master node, such as Hive and Hadoop. For more information on how to do this, go to [Add More than 256 Steps to a Job Flow](#) in the *Amazon Elastic MapReduce Developer's Guide*.

A step specifies the location of a JAR file stored either on the master node of the job flow or in Amazon S3. Each step is performed by the main function of the main class of the JAR file. The main class can be specified either in the manifest of the JAR or by using the MainFunction parameter of the step.

Elastic MapReduce executes each step in the order listed. For a step to be considered complete, the main function must exit with a zero exit code and all Hadoop jobs started while the step was running must have completed and run successfully.

You can only add steps to a job flow that is in one of the following states: STARTING, BOOTSTRAPPING, RUNNING, or WAITING.

Request Syntax

```
response = client.add_job_flow_steps(
    JobFlowId='string',
    Steps=[
        {
            'Name': 'string',
            'ActionOnFailure': 'TERMINATE_JOB_FLOW' | 'TERMINATE_CLUSTER' | 'CANCEL_AND_WAIT' | '...',
            'HadoopJarStep': {
                'Properties': [
                    {
                        'Key': 'string',
                        'Value': 'string'
                    },
                ],
            },
            'Jar': 'string',
            'MainClass': 'string',
            'Args': [
                'string',
            ]
        },
    ],
)
```

Parameters

- **JobFlowId** (*string*) – [REQUIRED]

A string that uniquely identifies the job flow. This identifier is returned by RunJobFlow and can also be obtained from ListClusters .

- **Steps** (*list*) – **[REQUIRED]**

A list of StepConfig to be executed by the job flow.

- (*dict*) –

Specification of a job flow step.

- * **Name** (*string*) – **[REQUIRED]**

The name of the job flow step.

- * **ActionOnFailure** (*string*) –

The action to take if the job flow step fails.

- * **HadoopJarStep** (*dict*) – **[REQUIRED]**

The JAR file used for the job flow step.

- **Properties** (*list*) –

A list of Java properties that are set when the step runs. You can use these properties to pass key value pairs to your main function.

- (*dict*) –

A key value pair.

- **Key** (*string*) –

The unique identifier of a key value pair.

- **Value** (*string*) –

The value part of the identified key.

- **Jar** (*string*) – **[REQUIRED]**

A path to a JAR file run during the step.

- **MainClass** (*string*) –

The name of the main class in the specified Java file. If not specified, the JAR file should specify a Main-Class in its manifest file.

- **Args** (*list*) –

A list of command line arguments passed to the JAR file's main function when executed.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'StepIds': [
        'string',
    ]
}
```

Response Structure

- (*dict*) –

The output for the AddJobFlowSteps operation.

- **StepIds** (*list*) –

The identifiers of the list of steps added to the job flow.

* (*string*) –

add_tags (***kwargs*)

Adds tags to an Amazon EMR resource. Tags make it easier to associate clusters in various ways, such as grouping clusters to track your Amazon EMR resource allocation costs. For more information, see [Tagging Amazon EMR Resources](#).

Request Syntax

```
response = client.add_tags(
    ResourceId='string',
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **ResourceId** (*string*) – **[REQUIRED]**

The Amazon EMR resource identifier to which tags will be added. This value must be a cluster identifier.

- **Tags** (*list*) – **[REQUIRED]**

A list of tags to associate with a cluster and propagate to Amazon EC2 instances. Tags are user-defined key/value pairs that consist of a required key string with a maximum of 128 characters, and an optional value string with a maximum of 256 characters.

– (*dict*) –

A key/value pair containing user-defined metadata that you can associate with an Amazon EMR resource. Tags make it easier to associate clusters in various ways, such as grouping clusters to track your Amazon EMR resource allocation costs. For more information, see [Tagging Amazon EMR Resources](#).

* **Key** (*string*) –

A user-defined key, which is the minimum required information for a valid tag. For more information, see [Tagging Amazon EMR Resources](#).

* **Value** (*string*) –

A user-defined value, which is optional in a tag. For more information, see [Tagging Amazon EMR Resources](#).

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –

This output indicates the result of adding tags to a resource.

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters `operation_name` (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns `True` if the operation can be paginated, `False` otherwise.

describe_cluster (***kwargs*)

Provides cluster-level details including status, hardware and software configuration, VPC settings, and so on. For information about the cluster steps, see `ListSteps`.

Request Syntax

```
response = client.describe_cluster(
    ClusterId='string'
)
```

Parameters `ClusterId` (*string*) – [REQUIRED]

The identifier of the cluster to describe.

Return type dict

Returns

Response Syntax

```
{
  'Cluster': {
    'Id': 'string',
    'Name': 'string',
    'Status': {
      'State': 'STARTING'|'BOOTSTRAPPING'|'RUNNING'|'WAITING'|'TERMINATING'|'DELETING',
      'StateChangeReason': {
        'Code': 'INTERNAL_ERROR'|'VALIDATION_ERROR'|'INSTANCE_FAILURE'|'BOOTSTRAPPING_ERROR',
        'Message': 'string'
      },
      'Timeline': {
        'CreationDateTime': datetime(2015, 1, 1),
        'ReadyDateTime': datetime(2015, 1, 1),
        'EndDateTime': datetime(2015, 1, 1)
      }
    },
    'Ec2InstanceAttributes': {
      'Ec2KeyName': 'string',
      'Ec2SubnetId': 'string',
      'Ec2AvailabilityZone': 'string',
      'IamInstanceProfile': 'string',
      'EmrManagedMasterSecurityGroup': 'string',
      'EmrManagedSlaveSecurityGroup': 'string',
      'AdditionalMasterSecurityGroups': [
        'string',
      ],
      'AdditionalSlaveSecurityGroups': [
        'string',
      ]
    },
    'LogUri': 'string',
    'RequestedAmiVersion': 'string',
    'RunningAmiVersion': 'string',
    'ReleaseLabel': 'string',
```

```

'AutoTerminate': True|False,
'TerminationProtected': True|False,
'VisibleToAllUsers': True|False,
'Applications': [
    {
        'Name': 'string',
        'Version': 'string',
        'Args': [
            'string',
        ],
        'AdditionalInfo': {
            'string': 'string'
        }
    },
],
'Tags': [
    {
        'Key': 'string',
        'Value': 'string'
    },
],
'ServiceRole': 'string',
'NormalizedInstanceHours': 123,
'MasterPublicDnsName': 'string',
'Configurations': [
    {
        'Classification': 'string',
        'Configurations': {'... recursive ...'},
        'Properties': {
            'string': 'string'
        }
    },
],
]
}

```

Response Structure

- (*dict*) –

This output contains the description of the cluster.

- **Cluster** (*dict*) –

This output contains the details for the requested cluster.

- * **Id** (*string*) –

The unique identifier for the cluster.

- * **Name** (*string*) –

The name of the cluster.

- * **Status** (*dict*) –

The current status details about the cluster.

- **State** (*string*) –

The current state of the cluster.

- **StateChangeReason** (*dict*) –

The reason for the cluster status change.

- **Code** (*string*) –

The programmatic code for the state change reason.

- **Message** (*string*) –
The descriptive message for the state change reason.
- **Timeline** (*dict*) –
A timeline that represents the status of a cluster over the life-time of the cluster.
- **CreationDateTime** (*datetime*) –
The creation date and time of the cluster.
- **ReadyDateTime** (*datetime*) –
The date and time when the cluster was ready to execute steps.
- **EndDateTime** (*datetime*) –
The date and time when the cluster was terminated.

* **Ec2InstanceAttributes** (*dict*) –

Provides information about the EC2 instances in a cluster grouped by category. For example, key name, subnet ID, IAM instance profile, and so on.

- **Ec2KeyName** (*string*) –
The name of the Amazon EC2 key pair to use when connecting with SSH into the master node as a user named “hadoop”.
- **Ec2SubnetId** (*string*) –
To launch the job flow in Amazon VPC, set this parameter to the identifier of the Amazon VPC subnet where you want the job flow to launch. If you do not specify this value, the job flow is launched in the normal AWS cloud, outside of a VPC.

Amazon VPC currently does not support cluster compute quadruple extra large (cc1.4xlarge) instances. Thus, you cannot specify the cc1.4xlarge instance type for nodes of a job flow launched in a VPC.
- **Ec2AvailabilityZone** (*string*) –
The Availability Zone in which the cluster will run.
- **IamInstanceProfile** (*string*) –
The IAM role that was specified when the job flow was launched. The EC2 instances of the job flow assume this role.
- **EmrManagedMasterSecurityGroup** (*string*) –
The identifier of the Amazon EC2 security group (managed by Amazon Elastic MapReduce) for the master node.
- **EmrManagedSlaveSecurityGroup** (*string*) –
The identifier of the Amazon EC2 security group (managed by Amazon Elastic MapReduce) for the slave nodes.
- **AdditionalMasterSecurityGroups** (*list*) –
A list of additional Amazon EC2 security group IDs for the master node.
(*string*) –
- **AdditionalSlaveSecurityGroups** (*list*) –
A list of additional Amazon EC2 security group IDs for the slave nodes.
(*string*) –

* **LogUri** (*string*) –

The path to the Amazon S3 location where logs for this cluster are stored.

* **RequestedAmiVersion** (*string*) –

The AMI version requested for this cluster.

* **RunningAmiVersion** (*string*) –

The AMI version running on this cluster.

* **ReleaseLabel** (*string*) –

The release label for the Amazon EMR release. For Amazon EMR 3.x and 2.x AMIs, use `amiVersion` instead of `ReleaseLabel`.

* **AutoTerminate** (*boolean*) –

Specifies whether the cluster should terminate after completing all steps.

* **TerminationProtected** (*boolean*) –

Indicates whether Amazon EMR will lock the cluster to prevent the EC2 instances from being terminated by an API call or user intervention, or in the event of a cluster error.

* **VisibleToAllUsers** (*boolean*) –

Indicates whether the job flow is visible to all IAM users of the AWS account associated with the job flow. If this value is set to `true`, all IAM users of that AWS account can view and manage the job flow if they have the proper policy permissions set. If this value is `false`, only the IAM user that created the cluster can view and manage it. This value can be changed using the `SetVisibleToAllUsers` action.

* **Applications** (*list*) –

The applications installed on this cluster.

· (*dict*) –

An application is any Amazon or third-party software that you can add to the cluster. This structure contains a list of strings that indicates the software to use with the cluster and accepts a user argument list. Amazon EMR accepts and forwards the argument list to the corresponding installation script as bootstrap action argument. For more information, see [Launch a Job Flow on the MapR Distribution for Hadoop](#). Currently supported values are:

- “mapr-m3” - launch the job flow using MapR M3 Edition.
- “mapr-m5” - launch the job flow using MapR M5 Edition.
- “mapr” with the user arguments specifying “-edition,m3” or “-edition,m5” - launch the job flow using MapR M3 or M5 Edition, respectively.

Note: In Amazon EMR releases 4.0 and greater, the only accepted parameter is the application name. To pass arguments to applications, you supply a configuration for each application.

· **Name** (*string*) –

The name of the application.

· **Version** (*string*) –

The version of the application.

- **Args** (*list*) –

Arguments for Amazon EMR to pass to the application.

- (*string*) –
- **AdditionalInfo** (*dict*) –

This option is for advanced users only. This is meta information about third-party applications that third-party vendors use for testing purposes.

- (*string*) –
- (*string*) –

- * **Tags** (*list*) –

A list of tags associated with a cluster.

- (*dict*) –

A key/value pair containing user-defined metadata that you can associate with an Amazon EMR resource. Tags make it easier to associate clusters in various ways, such as grouping clusters to track your Amazon EMR resource allocation costs. For more information, see [Tagging Amazon EMR Resources](#).

- **Key** (*string*) –

A user-defined key, which is the minimum required information for a valid tag. For more information, see [Tagging Amazon EMR Resources](#).

- **Value** (*string*) –

A user-defined value, which is optional in a tag. For more information, see [Tagging Amazon EMR Resources](#).

- * **ServiceRole** (*string*) –

The IAM role that will be assumed by the Amazon EMR service to access AWS resources on your behalf.

- * **NormalizedInstanceHours** (*integer*) –

An approximation of the cost of the job flow, represented in m1.small/hours. This value is incremented one time for every hour an m1.small instance runs. Larger instances are weighted more, so an EC2 instance that is roughly four times more expensive would result in the normalized instance hours being incremented by four. This result is only an approximation and does not reflect the actual billing rate.

- * **MasterPublicDnsName** (*string*) –

The public DNS name of the master EC2 instance.

- * **Configurations** (*list*) –

Note: Amazon EMR releases 4.x or later.

The list of Configurations supplied to the EMR cluster.

- (*dict*) –

Note: Amazon EMR releases 4.x or later.

Specifies a hardware and software configuration of the EMR

cluster. This includes configurations for applications and software bundled with Amazon EMR. The Configuration object is a JSON object which is defined by a classification and a set of properties. Configurations can be nested, so a configuration may have its own Configuration objects listed.

- **Classification** (*string*) –

The classification of a configuration. For more information see, [Amazon EMR Configurations](#).

- **Configurations** (*list*) –

A list of configurations you apply to this configuration object.

- **Properties** (*dict*) –

A set of properties supplied to the Configuration object.

- (*string*) –

- (*string*) –

describe_job_flows (***kwargs*)

This API is deprecated and will eventually be removed. We recommend you use `ListClusters`, `DescribeCluster`, `ListSteps`, `ListInstanceGroups` and `ListBootstrapActions` instead.

`DescribeJobFlows` returns a list of job flows that match all of the supplied parameters. The parameters can include a list of job flow IDs, job flow states, and restrictions on job flow creation date and time.

Regardless of supplied parameters, only job flows created within the last two months are returned.

If no parameters are supplied, then job flows matching either of the following criteria are returned:

- Job flows created and completed in the last two weeks
- Job flows created within the last two months that are in one of the following states: `RUNNING`, `WAITING`, `SHUTTING_DOWN`, `STARTING`

Amazon Elastic MapReduce can return a maximum of 512 job flow descriptions.

Request Syntax

```
response = client.describe_job_flows(
    CreatedAfter=datetime(2015, 1, 1),
    CreatedBefore=datetime(2015, 1, 1),
    JobFlowIds=[
        'string',
    ],
    JobFlowStates=[
        'STARTING' | 'BOOTSTRAPPING' | 'RUNNING' | 'WAITING' | 'SHUTTING_DOWN' | 'TERMINATED' | 'COMPLE
    ]
)
```

Parameters

- **CreatedAfter** (*datetime*) – Return only job flows created after this date and time.
- **CreatedBefore** (*datetime*) – Return only job flows created before this date and time.
- **JobFlowIds** (*list*) – Return only job flows whose job flow ID is contained in this list.
 - (*string*) –
- **JobFlowStates** (*list*) – Return only job flows whose state is contained in this list.
 - (*string*) –

The type of instance.

A small instance

A large instance

Return type dict

Returns

Response Syntax

```
{
  'JobFlows': [
    {
      'JobFlowId': 'string',
      'Name': 'string',
      'LogUri': 'string',
      'AmiVersion': 'string',
      'ExecutionStatusDetail': {
        'State': 'STARTING'|'BOOTSTRAPPING'|'RUNNING'|'WAITING'|'SHUTTING_
        'CreationDateTime': datetime(2015, 1, 1),
        'StartDateTime': datetime(2015, 1, 1),
        'ReadyDateTime': datetime(2015, 1, 1),
        'EndDateTime': datetime(2015, 1, 1),
        'LastStateChangeReason': 'string'
      },
      'Instances': {
        'MasterInstanceType': 'string',
        'MasterPublicDnsName': 'string',
        'MasterInstanceId': 'string',
        'SlaveInstanceType': 'string',
        'InstanceCount': 123,
        'InstanceGroups': [
          {
            'InstanceGroupId': 'string',
            'Name': 'string',
            'Market': 'ON_DEMAND'|'SPOT',
            'InstanceRole': 'MASTER'|'CORE'|'TASK',
            'BidPrice': 'string',
            'InstanceType': 'string',
            'InstanceRequestCount': 123,
            'InstanceRunningCount': 123,
            'State': 'PROVISIONING'|'BOOTSTRAPPING'|'RUNNING'|'RESIZIN
            'LastStateChangeReason': 'string',
            'CreationDateTime': datetime(2015, 1, 1),
            'StartDateTime': datetime(2015, 1, 1),
            'ReadyDateTime': datetime(2015, 1, 1),
            'EndDateTime': datetime(2015, 1, 1)
          },
        ],
        'NormalizedInstanceHours': 123,
        'Ec2KeyName': 'string',
        'Ec2SubnetId': 'string',
        'Placement': {
          'AvailabilityZone': 'string'
        },
        'KeepJobFlowAliveWhenNoSteps': True|False,
        'TerminationProtected': True|False,
        'HadoopVersion': 'string'
      },
      'Steps': [
        {
          'StepConfig': {
```



```

        'Name': 'string',
        'ActionOnFailure': 'TERMINATE_JOB_FLOW'|'TERMINATE_CLUSTER',
        'HadoopJarStep': {
            'Properties': [
                {
                    'Key': 'string',
                    'Value': 'string'
                },
            ],
            'Jar': 'string',
            'MainClass': 'string',
            'Args': [
                'string',
            ]
        },
    },
    'ExecutionStatusDetail': {
        'State': 'PENDING'|'RUNNING'|'CONTINUE'|'COMPLETED'|'CANCELLED',
        'CreationDateTime': datetime(2015, 1, 1),
        'StartDateTime': datetime(2015, 1, 1),
        'EndDateTime': datetime(2015, 1, 1),
        'LastStateChangeReason': 'string'
    },
},
],
'BootstrapActions': [
    {
        'BootstrapActionConfig': {
            'Name': 'string',
            'ScriptBootstrapAction': {
                'Path': 'string',
                'Args': [
                    'string',
                ]
            }
        },
    },
],
'SupportedProducts': [
    'string',
],
'VisibleToAllUsers': True|False,
'JobFlowRole': 'string',
'ServiceRole': 'string'
},
]
}

```

Response Structure

- (*dict*) –

The output for the DescribeJobFlows operation.

- **JobFlows** (*list*) –

A list of job flows matching the parameters supplied.

- * (*dict*) –

A description of a job flow.

- **JobFlowId** (*string*) –

The job flow identifier.

- **Name** (*string*) –

The name of the job flow.

- **LogUri** (*string*) –

The location in Amazon S3 where log files for the job are stored.

- **AmiVersion** (*string*) –

The version of the AMI used to initialize Amazon EC2 instances in the job flow. For a list of AMI versions currently supported by Amazon ElasticMapReduce, go to [AMI Versions Supported in Elastic MapReduce](#) in the *Amazon Elastic MapReduce Developer Guide*.

- **ExecutionStatusDetail** (*dict*) –

Describes the execution status of the job flow.

- **State** (*string*) –

The state of the job flow.

- **CreationDateTime** (*datetime*) –

The creation date and time of the job flow.

- **StartDateTime** (*datetime*) –

The start date and time of the job flow.

- **ReadyDateTime** (*datetime*) –

The date and time when the job flow was ready to start running bootstrap actions.

- **EndDateTime** (*datetime*) –

The completion date and time of the job flow.

- **LastStateChangeReason** (*string*) –

Description of the job flow last changed state.

- **Instances** (*dict*) –

Describes the Amazon EC2 instances of the job flow.

- **MasterInstanceType** (*string*) –

The Amazon EC2 master node instance type.

- **MasterPublicDnsName** (*string*) –

The DNS name of the master node.

- **MasterInstanceId** (*string*) –

The Amazon EC2 instance identifier of the master node.

- **SlaveInstanceType** (*string*) –

The Amazon EC2 slave node instance type.

- **InstanceCount** (*integer*) –

The number of Amazon EC2 instances in the cluster. If the value is 1, the same instance serves as both the master and slave node. If the value is greater than 1, one instance is the master node and all others are slave nodes.

- **InstanceGroups** (*list*) –

Details about the job flow's instance groups.

- *(dict)* –
Detailed information about an instance group.
- **InstanceGroupId** (*string*) –
Unique identifier for the instance group.
- **Name** (*string*) –
Friendly name for the instance group.
- **Market** (*string*) –
Market type of the Amazon EC2 instances used to create a cluster node.
- **InstanceRole** (*string*) –
Instance group role in the cluster
- **BidPrice** (*string*) –
Bid price for EC2 Instances when launching nodes as Spot Instances, expressed in USD.
- **InstanceType** (*string*) –
Amazon EC2 Instance type.
- **InstanceRequestCount** (*integer*) –
Target number of instances to run in the instance group.
- **InstanceRunningCount** (*integer*) –
Actual count of running instances.
- **State** (*string*) –
State of instance group. The following values are deprecated: STARTING, TERMINATED, and FAILED.
- **LastStateChangeReason** (*string*) –
Details regarding the state of the instance group.
- **CreationDateTime** (*datetime*) –
The date/time the instance group was created.
- **StartDateTime** (*datetime*) –
The date/time the instance group was started.
- **ReadyDateTime** (*datetime*) –
The date/time the instance group was available to the cluster.
- **EndDateTime** (*datetime*) –
The date/time the instance group was terminated.
- **NormalizedInstanceHours** (*integer*) –
An approximation of the cost of the job flow, represented in m1.small/hours. This value is incremented once for every hour an m1.small runs. Larger instances are weighted more, so an Amazon EC2 instance that is roughly four times more expensive would result in the normalized instance hours being incremented by four. This result is only an approximation and does not reflect the actual billing rate.
- **Ec2KeyName** (*string*) –
The name of an Amazon EC2 key pair that can be used to ssh to the master node of job flow.

- **Ec2SubnetId** (*string*) –
For job flows launched within Amazon Virtual Private Cloud, this value specifies the identifier of the subnet where the job flow was launched.
- **Placement** (*dict*) –
The Amazon EC2 Availability Zone for the job flow.
- **AvailabilityZone** (*string*) –
The Amazon EC2 Availability Zone for the job flow.
- **KeepJobFlowAliveWhenNoSteps** (*boolean*) –
Specifies whether the job flow should terminate after completing all steps.
- **TerminationProtected** (*boolean*) –
Specifies whether the Amazon EC2 instances in the cluster are protected from termination by API calls, user intervention, or in the event of a job flow error.
- **HadoopVersion** (*string*) –
The Hadoop version for the job flow.
- **Steps** (*list*) –
A list of steps run by the job flow.
- (*dict*) –
Combines the execution state and configuration of a step.
- **StepConfig** (*dict*) –
The step configuration.
- **Name** (*string*) –
The name of the job flow step.
- **ActionOnFailure** (*string*) –
The action to take if the job flow step fails.
- **HadoopJarStep** (*dict*) –
The JAR file used for the job flow step.
- **Properties** (*list*) –
A list of Java properties that are set when the step runs. You can use these properties to pass key value pairs to your main function.
- (*dict*) –
A key value pair.
- **Key** (*string*) –
The unique identifier of a key value pair.
- **Value** (*string*) –
The value part of the identified key.
- **Jar** (*string*) –
A path to a JAR file run during the step.
- **MainClass** (*string*) –
The name of the main class in the specified Java file. If not specified, the JAR file should specify a Main-Class in its manifest file.

- **Args** (*list*) –
A list of command line arguments passed to the JAR file's main function when executed.
- (*string*) –
- **ExecutionStatusDetail** (*dict*) –
The description of the step status.
- **State** (*string*) –
The state of the job flow step.
- **CreationDateTime** (*datetime*) –
The creation date and time of the step.
- **StartDateTime** (*datetime*) –
The start date and time of the step.
- **EndDateTime** (*datetime*) –
The completion date and time of the step.
- **LastStateChangeReason** (*string*) –
A description of the step's current state.
- **BootstrapActions** (*list*) –
A list of the bootstrap actions run by the job flow.
- (*dict*) –
Reports the configuration of a bootstrap action in a job flow.
- **BootstrapActionConfig** (*dict*) –
A description of the bootstrap action.
- **Name** (*string*) –
The name of the bootstrap action.
- **ScriptBootstrapAction** (*dict*) –
The script run by the bootstrap action.
- **Path** (*string*) –
Location of the script to run during a bootstrap action. Can be either a location in Amazon S3 or on a local file system.
- **Args** (*list*) –
A list of command line arguments to pass to the bootstrap action script.
- (*string*) –
- **SupportedProducts** (*list*) –
A list of strings set by third party software when the job flow is launched. If you are not using third party software to manage the job flow this value is empty.
- (*string*) –
- **VisibleToAllUsers** (*boolean*) –
Specifies whether the job flow is visible to all IAM users of the AWS account associated with the job flow. If this value is set to `true`, all IAM users of that AWS account can view and (if they have the proper policy permissions set) manage the job flow. If it is set to `false`, only the IAM user that created the job flow can view and manage it. This value can be changed using the `SetVisibleToAllUsers` action.

- **JobFlowRole** (*string*) –
The IAM role that was specified when the job flow was launched. The EC2 instances of the job flow assume this role.
- **ServiceRole** (*string*) –
The IAM role that will be assumed by the Amazon EMR service to access AWS resources on your behalf.

describe_step (***kwargs*)

Provides more detail about the cluster step.

Request Syntax

```
response = client.describe_step(
    ClusterId='string',
    StepId='string'
)
```

Parameters

- **ClusterId** (*string*) – **[REQUIRED]**
The identifier of the cluster with steps to describe.
- **StepId** (*string*) – **[REQUIRED]**
The identifier of the step to describe.

Return type dict

Returns

Response Syntax

```
{
  'Step': {
    'Id': 'string',
    'Name': 'string',
    'Config': {
      'Jar': 'string',
      'Properties': {
        'string': 'string'
      },
      'MainClass': 'string',
      'Args': [
        'string',
      ]
    },
    'ActionOnFailure': 'TERMINATE_JOB_FLOW' | 'TERMINATE_CLUSTER' | 'CANCEL_AND_WAIT',
    'Status': {
      'State': 'PENDING' | 'RUNNING' | 'COMPLETED' | 'CANCELLED' | 'FAILED' | 'INTERRUPTED',
      'StateChangeReason': {
        'Code': 'NONE',
        'Message': 'string'
      },
      'Timeline': {
        'CreationDateTime': datetime(2015, 1, 1),
        'StartDateTime': datetime(2015, 1, 1),
        'EndDateTime': datetime(2015, 1, 1)
      }
    }
  }
}
```

Response Structure

- *(dict)* –

This output contains the description of the cluster step.

- **Step** *(dict)* –

The step details for the requested step identifier.

- * **Id** *(string)* –

The identifier of the cluster step.

- * **Name** *(string)* –

The name of the cluster step.

- * **Config** *(dict)* –

The Hadoop job configuration of the cluster step.

- **Jar** *(string)* –

The path to the JAR file that runs during the step.

- **Properties** *(dict)* –

The list of Java properties that are set when the step runs. You can use these properties to pass key value pairs to your main function.

- *(string)* –

- *(string)* –

- **MainClass** *(string)* –

The name of the main class in the specified Java file. If not specified, the JAR file should specify a main class in its manifest file.

- **Args** *(list)* –

The list of command line arguments to pass to the JAR file's main function for execution.

- *(string)* –

- * **ActionOnFailure** *(string)* –

This specifies what action to take when the cluster step fails. Possible values are TERMINATE_CLUSTER, CANCEL_AND_WAIT, and CONTINUE.

- * **Status** *(dict)* –

The current execution status details of the cluster step.

- **State** *(string)* –

The execution state of the cluster step.

- **StateChangeReason** *(dict)* –

The reason for the step execution status change.

- **Code** *(string)* –

The programmable code for the state change reason. Note: Currently, the service provides no code for the state change.

- **Message** *(string)* –

The descriptive message for the state change reason.

- **Timeline** *(dict)* –

The timeline of the cluster step status over time.

- **CreationDateTime** *(datetime)* –

The date and time when the cluster step was created.

- **StartDateTime** (*datetime*) –

The date and time when the cluster step execution started.

- **EndDateTime** (*datetime*) –

The date and time when the cluster step execution completed or failed.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to *ClientMethod*.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type L{botocore.paginate.Paginator}

Returns A paginator object.

get_waiter (*waiter_name*)

list_bootstrap_actions (***kwargs*)

Provides information about the bootstrap actions associated with a cluster.

Request Syntax

```
response = client.list_bootstrap_actions(  
    ClusterId='string',  
    Marker='string'  
)
```

Parameters

- **ClusterId** (*string*) – **[REQUIRED]**
The cluster identifier for the bootstrap actions to list .
- **Marker** (*string*) – The pagination token that indicates the next set of results to retrieve .

Return type dict

Returns

Response Syntax

```
{  
    'BootstrapActions': [  
        {  
            'Name': 'string',
```



```

        'ScriptPath': 'string',
        'Args': [
            'string',
        ],
    },
],
'Marker': 'string'
}

```

Response Structure

- (*dict*) –

This output contains the bootstrap actions detail .

- **BootstrapActions** (*list*) –

The bootstrap actions associated with the cluster .

- * (*dict*) –

An entity describing an executable that runs on a cluster.

- **Name** (*string*) –

The name of the command.

- **ScriptPath** (*string*) –

The Amazon S3 location of the command script.

- **Args** (*list*) –

Arguments for Amazon EMR to pass to the command for execution.

- (*string*) –

- **Marker** (*string*) –

The pagination token that indicates the next set of results to retrieve .

list_clusters (***kwargs*)

Provides the status of all clusters visible to this AWS account. Allows you to filter the list of clusters based on certain criteria; for example, filtering by cluster creation date and time or by status. This call returns a maximum of 50 clusters per call, but returns a marker to track the paging of the cluster list across multiple ListClusters calls.

Request Syntax

```

response = client.list_clusters(
    CreatedAfter=datetime(2015, 1, 1),
    CreatedBefore=datetime(2015, 1, 1),
    ClusterStates=[
        'STARTING' | 'BOOTSTRAPPING' | 'RUNNING' | 'WAITING' | 'TERMINATING' | 'TERMINATED' | 'TERMINAT
    ],
    Marker='string'
)

```

Parameters

- **CreatedAfter** (*datetime*) – The creation date and time beginning value filter for listing clusters .
- **CreatedBefore** (*datetime*) – The creation date and time end value filter for listing clusters .
- **ClusterStates** (*list*) – The cluster state filters to apply when listing clusters.
 - (*string*) –
- **Marker** (*string*) – The pagination token that indicates the next set of results to retrieve.

Return type dict

Returns

Response Syntax

```
{
  'Clusters': [
    {
      'Id': 'string',
      'Name': 'string',
      'Status': {
        'State': 'STARTING'|'BOOTSTRAPPING'|'RUNNING'|'WAITING'|'TERMINATING',
        'StateChangeReason': {
          'Code': 'INTERNAL_ERROR'|'VALIDATION_ERROR'|'INSTANCE_FAILURE',
          'Message': 'string'
        },
        'Timeline': {
          'CreationDateTime': datetime(2015, 1, 1),
          'ReadyDateTime': datetime(2015, 1, 1),
          'EndDateTime': datetime(2015, 1, 1)
        }
      },
      'NormalizedInstanceHours': 123
    },
    ...
  ],
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

This contains a ClusterSummaryList with the cluster details; for example, the cluster IDs, names, and status.

- **Clusters** (*list*) –

The list of clusters for the account based on the given filters.

- * (*dict*) –

The summary description of the cluster.

- **Id** (*string*) –

The unique identifier for the cluster.

- **Name** (*string*) –

The name of the cluster.

- **Status** (*dict*) –

The details about the current status of the cluster.

- **State** (*string*) –

The current state of the cluster.

- **StateChangeReason** (*dict*) –

The reason for the cluster status change.

- **Code** (*string*) –

The programmatic code for the state change reason.

- **Message** (*string*) –

The descriptive message for the state change reason.

- **Timeline** (*dict*) –
A timeline that represents the status of a cluster over the life-time of the cluster.
- **CreationDateTime** (*datetime*) –
The creation date and time of the cluster.
- **ReadyDateTime** (*datetime*) –
The date and time when the cluster was ready to execute steps.
- **EndDateTime** (*datetime*) –
The date and time when the cluster was terminated.
- **NormalizedInstanceHours** (*integer*) –
An approximation of the cost of the job flow, represented in m1.small/hours. This value is incremented one time for every hour an m1.small instance runs. Larger instances are weighted more, so an EC2 instance that is roughly four times more expensive would result in the normalized instance hours being incremented by four. This result is only an approximation and does not reflect the actual billing rate.

– **Marker** (*string*) –

The pagination token that indicates the next set of results to retrieve.

list_instance_groups (***kwargs*)

Provides all available details about the instance groups in a cluster.

Request Syntax

```
response = client.list_instance_groups(
    ClusterId='string',
    Marker='string'
)
```

Parameters

- **ClusterId** (*string*) – **[REQUIRED]**

The identifier of the cluster for which to list the instance groups.

- **Marker** (*string*) – The pagination token that indicates the next set of results to retrieve.

Return type dict

Returns

Response Syntax

```
{
    'InstanceGroups': [
        {
            'Id': 'string',
            'Name': 'string',
            'Market': 'ON_DEMAND' | 'SPOT',
            'InstanceGroupType': 'MASTER' | 'CORE' | 'TASK',
            'BidPrice': 'string',
            'InstanceType': 'string',
            'RequestedInstanceCount': 123,
            'RunningInstanceCount': 123,
            'Status': {
                'State': 'PROVISIONING' | 'BOOTSTRAPPING' | 'RUNNING' | 'RESIZING' | 'SUSPENDED'
```

```
        'StateChangeReason': {
            'Code': 'INTERNAL_ERROR'|'VALIDATION_ERROR'|'INSTANCE_FAILURE',
            'Message': 'string'
        },
        'Timeline': {
            'CreationDateTime': datetime(2015, 1, 1),
            'ReadyDateTime': datetime(2015, 1, 1),
            'EndDateTime': datetime(2015, 1, 1)
        }
    },
    'Configurations': [
        {
            'Classification': 'string',
            'Configurations': {'... recursive ...'},
            'Properties': {
                'string': 'string'
            }
        }
    ],
    'Marker': 'string'
}
```

Response Structure

- *(dict)* –

This input determines which instance groups to retrieve.

- **InstanceGroups** (*list*) –

The list of instance groups for the cluster and given filters.

- * *(dict)* –

This entity represents an instance group, which is a group of instances that have common purpose. For example, CORE instance group is used for HDFS.

- **Id** (*string*) –

The identifier of the instance group.

- **Name** (*string*) –

The name of the instance group.

- **Market** (*string*) –

The marketplace to provision instances for this group. Valid values are ON_DEMAND or SPOT.

- **InstanceGroupType** (*string*) –

The type of the instance group. Valid values are MASTER, CORE or TASK.

- **BidPrice** (*string*) –

The bid price for each EC2 instance in the instance group when launching nodes as Spot Instances, expressed in USD.

- **InstanceType** (*string*) –

The EC2 instance type for all instances in the instance group.

- **RequestedInstanceCount** (*integer*) –

The target number of instances for the instance group.

- **RunningInstanceCount** (*integer*) –
The number of instances currently running in this instance group.
- **Status** (*dict*) –
The current status of the instance group.
- **State** (*string*) –
The current state of the instance group.
- **StateChangeReason** (*dict*) –
The status change reason details for the instance group.
- **Code** (*string*) –
The programmable code for the state change reason.
- **Message** (*string*) –
The status change reason description.
- **Timeline** (*dict*) –
The timeline of the instance group status over time.
- **CreationDateTime** (*datetime*) –
The creation date and time of the instance group.
- **ReadyDateTime** (*datetime*) –
The date and time when the instance group became ready to perform tasks.
- **EndDateTime** (*datetime*) –
The date and time when the instance group terminated.
- **Configurations** (*list*) –

Note: Amazon EMR releases 4.x or later.

The list of configurations supplied for an EMR cluster instance group. You can specify a separate configuration for each instance group (master, core, and task).

- (*dict*) –

Note: Amazon EMR releases 4.x or later.

Specifies a hardware and software configuration of the EMR cluster. This includes configurations for applications and software bundled with Amazon EMR. The Configuration object is a JSON object which is defined by a classification and a set of properties. Configurations can be nested, so a configuration may have its own Configuration objects listed.

- **Classification** (*string*) –
The classification of a configuration. For more information see, [Amazon EMR Configurations](#).
- **Configurations** (*list*) –
A list of configurations you apply to this configuration object.
- **Properties** (*dict*) –
A set of properties supplied to the Configuration object.
- (*string*) –

- (*string*) –
- **Marker** (*string*) –

The pagination token that indicates the next set of results to retrieve.

list_instances (***kwargs*)

Provides information about the cluster instances that Amazon EMR provisions on behalf of a user when it creates the cluster. For example, this operation indicates when the EC2 instances reach the Ready state, when instances become available to Amazon EMR to use for jobs, and the IP addresses for cluster instances, etc.

Request Syntax

```
response = client.list_instances(
    ClusterId='string',
    InstanceGroupId='string',
    InstanceGroupTypes=[
        'MASTER'|'CORE'|'TASK',
    ],
    Marker='string'
)
```

Parameters

- **ClusterId** (*string*) – **[REQUIRED]**
The identifier of the cluster for which to list the instances.
- **InstanceGroupId** (*string*) – The identifier of the instance group for which to list the instances.
- **InstanceGroupTypes** (*list*) – The type of instance group for which to list the instances.
 - (*string*) –
- **Marker** (*string*) – The pagination token that indicates the next set of results to retrieve.

Return type dict

Returns

Response Syntax

```
{
    'Instances': [
        {
            'Id': 'string',
            'Ec2InstanceId': 'string',
            'PublicDnsName': 'string',
            'PublicIpAddress': 'string',
            'PrivateDnsName': 'string',
            'PrivateIpAddress': 'string',
            'Status': {
                'State': 'AWAITING_FULFILLMENT'|'PROVISIONING'|'BOOTSTRAPPING'|'RUNNING'|'TERMINATING'|'TERMINATED'|'REPLACING_INSTANCE_GROUP',
                'StateChangeReason': {
                    'Code': 'INTERNAL_ERROR'|'VALIDATION_ERROR'|'INSTANCE_FAILURE'|'REPLACING_INSTANCE_GROUP_FAILURE',
                    'Message': 'string'
                },
                'Timeline': {
                    'CreationDateTime': datetime(2015, 1, 1),
                    'ReadyDateTime': datetime(2015, 1, 1),
                    'EndDateTime': datetime(2015, 1, 1)
                }
            }
        }
    ]
}
```

```

        }
    },
],
'Marker': 'string'
}

```

Response Structure

- *(dict)* –

This output contains the list of instances.

- **Instances** *(list)* –

The list of instances for the cluster and given filters.

- * *(dict)* –

Represents an EC2 instance provisioned as part of cluster.

- **Id** *(string)* –

The unique identifier for the instance in Amazon EMR.

- **Ec2InstanceId** *(string)* –

The unique identifier of the instance in Amazon EC2.

- **PublicDnsName** *(string)* –

The public DNS name of the instance.

- **PublicIpAddress** *(string)* –

The public IP address of the instance.

- **PrivateDnsName** *(string)* –

The private DNS name of the instance.

- **PrivateIpAddress** *(string)* –

The private IP address of the instance.

- **Status** *(dict)* –

The current status of the instance.

- **State** *(string)* –

The current state of the instance.

- **StateChangeReason** *(dict)* –

The details of the status change reason for the instance.

- **Code** *(string)* –

The programmable code for the state change reason.

- **Message** *(string)* –

The status change reason description.

- **Timeline** *(dict)* –

The timeline of the instance status over time.

- **CreationDateTime** *(datetime)* –

The creation date and time of the instance.

- **ReadyDateTime** *(datetime)* –

The date and time when the instance was ready to perform tasks.

- **EndDateTime** *(datetime)* –

The date and time when the instance was terminated.

– **Marker** (*string*) –

The pagination token that indicates the next set of results to retrieve.

list_steps (***kwargs*)

Provides a list of steps for the cluster.

Request Syntax

```
response = client.list_steps(
    ClusterId='string',
    StepStates=[
        'PENDING' | 'RUNNING' | 'COMPLETED' | 'CANCELLED' | 'FAILED' | 'INTERRUPTED',
    ],
    StepIds=[
        'string',
    ],
    Marker='string'
)
```

Parameters

- **ClusterId** (*string*) – **[REQUIRED]**

The identifier of the cluster for which to list the steps.

- **StepStates** (*list*) – The filter to limit the step list based on certain states.

– (*string*) –

- **StepIds** (*list*) – The filter to limit the step list based on the identifier of the steps.

– (*string*) –

- **Marker** (*string*) – The pagination token that indicates the next set of results to retrieve.

Return type dict

Returns

Response Syntax

```
{
    'Steps': [
        {
            'Id': 'string',
            'Name': 'string',
            'Config': {
                'Jar': 'string',
                'Properties': {
                    'string': 'string'
                },
                'MainClass': 'string',
                'Args': [
                    'string',
                ]
            },
            'ActionOnFailure': 'TERMINATE_JOB_FLOW' | 'TERMINATE_CLUSTER' | 'CANCEL_AN...',
            'Status': {
                'State': 'PENDING' | 'RUNNING' | 'COMPLETED' | 'CANCELLED' | 'FAILED' | 'INT...',
                'StateChangeReason': {
                    'Code': 'NONE',
                    'Message': 'string'
                },
            },
            'Timeline': {
                'CreationDateTime': datetime(2015, 1, 1),
            },
        },
    ],
}
```



```

        'StartDateTime': datetime(2015, 1, 1),
        'EndDateTime': datetime(2015, 1, 1)
    }
    },
    ],
    'Marker': 'string'
}

```

Response Structure

- (*dict*) –

This output contains the list of steps.

- **Steps** (*list*) –

The filtered list of steps for the cluster.

- * (*dict*) –

The summary of the cluster step.

- **Id** (*string*) –

The identifier of the cluster step.

- **Name** (*string*) –

The name of the cluster step.

- **Config** (*dict*) –

The Hadoop job configuration of the cluster step.

- **Jar** (*string*) –

The path to the JAR file that runs during the step.

- **Properties** (*dict*) –

The list of Java properties that are set when the step runs. You can use these properties to pass key value pairs to your main function.

- (*string*) –

- (*string*) –

- **MainClass** (*string*) –

The name of the main class in the specified Java file. If not specified, the JAR file should specify a main class in its manifest file.

- **Args** (*list*) –

The list of command line arguments to pass to the JAR file's main function for execution.

- (*string*) –

- **ActionOnFailure** (*string*) –

This specifies what action to take when the cluster step fails. Possible values are TERMINATE_CLUSTER, CANCEL_AND_WAIT, and CONTINUE.

- **Status** (*dict*) –

The current execution status details of the cluster step.

- **State** (*string*) –

The execution state of the cluster step.

- **StateChangeReason** (*dict*) –

The reason for the step execution status change.

- **Code** (*string*) –

The programmable code for the state change reason. Note: Currently, the service provides no code for the state change.

- **Message** (*string*) –

The descriptive message for the state change reason.

- **Timeline** (*dict*) –

The timeline of the cluster step status over time.

- **CreationDateTime** (*datetime*) –

The date and time when the cluster step was created.

- **StartDateTime** (*datetime*) –

The date and time when the cluster step execution started.

- **EndDateTime** (*datetime*) –

The date and time when the cluster step execution completed or failed.

- **Marker** (*string*) –

The pagination token that indicates the next set of results to retrieve.

modify_instance_groups (***kwargs*)

ModifyInstanceGroups modifies the number of nodes and configuration settings of an instance group. The input parameters include the new target instance count for the group and the instance group ID. The call will either succeed or fail atomically.

Request Syntax

```
response = client.modify_instance_groups(
    InstanceGroups=[
        {
            'InstanceGroupId': 'string',
            'InstanceCount': 123,
            'EC2InstanceIdsToTerminate': [
                'string',
            ]
        },
    ]
)
```

Parameters **InstanceGroups** (*list*) – Instance groups to change.

- (*dict*) –

Modify an instance group size.

- **InstanceGroupId** (*string*) – **[REQUIRED]**

Unique ID of the instance group to expand or shrink.

- **InstanceCount** (*integer*) –

Target size for the instance group.

- **EC2InstanceIdsToTerminate** (*list*) –

The EC2 InstanceIds to terminate. For advanced users only. Once you terminate the instances, the instance group will not return to its original requested size.

- * (*string*) –

Returns None

remove_tags (**kwargs)

Removes tags from an Amazon EMR resource. Tags make it easier to associate clusters in various ways, such as grouping clusters to track your Amazon EMR resource allocation costs. For more information, see [Tagging Amazon EMR Resources](#).

The following example removes the stack tag with value Prod from a cluster:

Request Syntax

```
response = client.remove_tags(
    ResourceId='string',
    TagKeys=[
        'string',
    ]
)
```

Parameters

- **ResourceId** (*string*) – [REQUIRED]

The Amazon EMR resource identifier from which tags will be removed. This value must be a cluster identifier.

- **TagKeys** (*list*) – [REQUIRED]

A list of tag keys to remove from a resource.

– (*string*) –

Return type dict

Returns**Response Syntax**

```
{}
```

Response Structure

- (*dict*) –

This output indicates the result of removing tags from a resource.

run_job_flow (**kwargs)

RunJobFlow creates and starts running a new job flow. The job flow will run the steps specified. Once the job flow completes, the cluster is stopped and the HDFS partition is lost. To prevent loss of data, configure the last step of the job flow to store results in Amazon S3. If the JobFlowInstancesConfig KeepJobFlowAliveWhenNoSteps parameter is set to TRUE, the job flow will transition to the WAITING state rather than shutting down once the steps have completed.

For additional protection, you can set the JobFlowInstancesConfig TerminationProtected parameter to TRUE to lock the job flow and prevent it from being terminated by API call, user intervention, or in the event of a job flow error.

A maximum of 256 steps are allowed in each job flow.

If your job flow is long-running (such as a Hive data warehouse) or complex, you may require more than 256 steps to process your data. You can bypass the 256-step limitation in various ways, including using the SSH shell to connect to the master node and submitting queries directly to the software running on the master node, such as Hive and Hadoop. For more information on how to do this, go to [Add More than 256 Steps to a Job Flow](#) in the *Amazon Elastic MapReduce Developer's Guide*.

For long running job flows, we recommend that you periodically store your results.

Request Syntax

```
response = client.run_job_flow(
    Name='string',
    LogUri='string',
    AdditionalInfo='string',
    AmiVersion='string',
    ReleaseLabel='string',
    Instances={
        'MasterInstanceType': 'string',
        'SlaveInstanceType': 'string',
        'InstanceCount': 123,
        'InstanceGroups': [
            {
                'Name': 'string',
                'Market': 'ON_DEMAND'|'SPOT',
                'InstanceRole': 'MASTER'|'CORE'|'TASK',
                'BidPrice': 'string',
                'InstanceType': 'string',
                'InstanceCount': 123,
                'Configurations': [
                    {
                        'Classification': 'string',
                        'Configurations': {'... recursive ...'},
                        'Properties': {
                            'string': 'string'
                        }
                    }
                ],
            }
        ],
    },
    'Ec2KeyName': 'string',
    'Placement': {
        'AvailabilityZone': 'string'
    },
    'KeepJobFlowAliveWhenNoSteps': True|False,
    'TerminationProtected': True|False,
    'HadoopVersion': 'string',
    'Ec2SubnetId': 'string',
    'EmrManagedMasterSecurityGroup': 'string',
    'EmrManagedSlaveSecurityGroup': 'string',
    'AdditionalMasterSecurityGroups': [
        'string',
    ],
    'AdditionalSlaveSecurityGroups': [
        'string',
    ]
},
Steps=[
    {
        'Name': 'string',
        'ActionOnFailure': 'TERMINATE_JOB_FLOW'|'TERMINATE_CLUSTER'|'CANCEL_AND_WAIT'|'...',
        'HadoopJarStep': {
            'Properties': [
                {
                    'Key': 'string',
                    'Value': 'string'
                }
            ],
            'Jar': 'string',
        }
    }
]
```

```

        'MainClass': 'string',
        'Args': [
            'string',
        ]
    },
],
BootstrapActions=[
    {
        'Name': 'string',
        'ScriptBootstrapAction': {
            'Path': 'string',
            'Args': [
                'string',
            ]
        }
    },
],
SupportedProducts=[
    'string',
],
NewSupportedProducts=[
    {
        'Name': 'string',
        'Args': [
            'string',
        ]
    },
],
Applications=[
    {
        'Name': 'string',
        'Version': 'string',
        'Args': [
            'string',
        ],
        'AdditionalInfo': {
            'string': 'string'
        }
    },
],
Configurations=[
    {
        'Classification': 'string',
        'Configurations': {'... recursive ...'},
        'Properties': {
            'string': 'string'
        }
    },
],
VisibleToAllUsers=True|False,
JobFlowRole='string',
ServiceRole='string',
Tags=[
    {
        'Key': 'string',
        'Value': 'string'
    },
],

```

```
]
)
```

Parameters

- **Name** (*string*) – **[REQUIRED]**

The name of the job flow.

- **LogUri** (*string*) – The location in Amazon S3 to write the log files of the job flow. If a value is not provided, logs are not created.
- **AdditionalInfo** (*string*) – A JSON string for selecting additional features.
- **AmiVersion** (*string*) –

Note: For Amazon EMR releases 3.x and 2.x. For Amazon EMR releases 4.x and greater, use ReleaseLabel.

The version of the Amazon Machine Image (AMI) to use when launching Amazon EC2 instances in the job flow. The following values are valid:

- The version number of the AMI to use, for example, “2.0.”

If the AMI supports multiple versions of Hadoop (for example, AMI 1.0 supports both Hadoop 0.18 and 0.20) you can use the JobFlowInstancesConfig HadoopVersion parameter to modify the version of Hadoop from the defaults shown above.

For details about the AMI versions currently supported by Amazon Elastic MapReduce, go to [AMI Versions Supported in Elastic MapReduce](#) in the *Amazon Elastic MapReduce Developer’s Guide*.

- **ReleaseLabel** (*string*) –

Note: Amazon EMR releases 4.x or later.

The release label for the Amazon EMR release. For Amazon EMR 3.x and 2.x AMIs, use amiVersion instead instead of ReleaseLabel.

- **Instances** (*dict*) – **[REQUIRED]**

A specification of the number and type of Amazon EC2 instances on which to run the job flow.

- **MasterInstanceType** (*string*) –

The EC2 instance type of the master node.

- **SlaveInstanceType** (*string*) –

The EC2 instance type of the slave nodes.

- **InstanceCount** (*integer*) –

The number of Amazon EC2 instances used to execute the job flow.

- **InstanceGroups** (*list*) –

Configuration for the job flow’s instance groups.

* (*dict*) –

Configuration defining a new instance group.

- **Name** (*string*) –

Friendly name given to the instance group.

- **Market** (*string*) –

Market type of the Amazon EC2 instances used to create a cluster node.

- **InstanceRole** (*string*) – [REQUIRED]

The role of the instance group in the cluster.

- **BidPrice** (*string*) –

Bid price for each Amazon EC2 instance in the instance group when launching nodes as Spot Instances, expressed in USD.

- **InstanceType** (*string*) – [REQUIRED]

The Amazon EC2 instance type for all instances in the instance group.

- **InstanceCount** (*integer*) – [REQUIRED]

Target number of instances for the instance group.

- **Configurations** (*list*) –

Note: Amazon EMR releases 4.x or later.

The list of configurations supplied for an EMR cluster instance group. You can specify a separate configuration for each instance group (master, core, and task).

- (*dict*) –

Note: Amazon EMR releases 4.x or later.

Specifies a hardware and software configuration of the EMR cluster. This includes configurations for applications and software bundled with Amazon EMR. The Configuration object is a JSON object which is defined by a classification and a set of properties. Configurations can be nested, so a configuration may have its own Configuration objects listed.

- **Classification** (*string*) –

The classification of a configuration. For more information see, [Amazon EMR Configurations](#).

- **Configurations** (*list*) –

A list of configurations you apply to this configuration object.

- **Properties** (*dict*) –

A set of properties supplied to the Configuration object.

- (*string*) –

- (*string*) –

- **Ec2KeyName** (*string*) –

The name of the Amazon EC2 key pair that can be used to ssh to the master node as the user called “hadoop.”

- **Placement** (*dict*) –

The Availability Zone the job flow will run in.

- **AvailabilityZone** (*string*) – [REQUIRED]

The Amazon EC2 Availability Zone for the job flow.

- **KeepJobFlowAliveWhenNoSteps** (*boolean*) –

Specifies whether the job flow should be kept alive after completing all steps.

- **TerminationProtected** (*boolean*) –

Specifies whether to lock the job flow to prevent the Amazon EC2 instances from being terminated by API call, user intervention, or in the event of a job flow error.

– **HadoopVersion** (*string*) –

The Hadoop version for the job flow. Valid inputs are “0.18” (deprecated), “0.20” (deprecated), “0.20.205” (deprecated), “1.0.3”, “2.2.0”, or “2.4.0”. If you do not set this value, the default of 0.18 is used, unless the `AmiVersion` parameter is set in the `RunJobFlow` call, in which case the default version of Hadoop for that AMI version is used.

– **Ec2SubnetId** (*string*) –

To launch the job flow in Amazon Virtual Private Cloud (Amazon VPC), set this parameter to the identifier of the Amazon VPC subnet where you want the job flow to launch. If you do not specify this value, the job flow is launched in the normal Amazon Web Services cloud, outside of an Amazon VPC.

Amazon VPC currently does not support cluster compute quadruple extra large (cc1.4xlarge) instances. Thus you cannot specify the cc1.4xlarge instance type for nodes of a job flow launched in a Amazon VPC.

– **EmrManagedMasterSecurityGroup** (*string*) –

The identifier of the Amazon EC2 security group (managed by Amazon ElasticMapReduce) for the master node.

– **EmrManagedSlaveSecurityGroup** (*string*) –

The identifier of the Amazon EC2 security group (managed by Amazon ElasticMapReduce) for the slave nodes.

– **AdditionalMasterSecurityGroups** (*list*) –

A list of additional Amazon EC2 security group IDs for the master node.

* (*string*) –

– **AdditionalSlaveSecurityGroups** (*list*) –

A list of additional Amazon EC2 security group IDs for the slave nodes.

* (*string*) –

• **Steps** (*list*) – A list of steps to be executed by the job flow.

– (*dict*) –

Specification of a job flow step.

* **Name** (*string*) – **[REQUIRED]**

The name of the job flow step.

* **ActionOnFailure** (*string*) –

The action to take if the job flow step fails.

* **HadoopJarStep** (*dict*) – **[REQUIRED]**

The JAR file used for the job flow step.

· **Properties** (*list*) –

A list of Java properties that are set when the step runs. You can use these properties to pass key value pairs to your main function.

· (*dict*) –

A key value pair.

· **Key** (*string*) –

The unique identifier of a key value pair.

- **Value** (*string*) –
The value part of the identified key.
- **Jar** (*string*) – **[REQUIRED]**
A path to a JAR file run during the step.
- **MainClass** (*string*) –
The name of the main class in the specified Java file. If not specified, the JAR file should specify a Main-Class in its manifest file.
- **Args** (*list*) –
A list of command line arguments passed to the JAR file's main function when executed.
- (*string*) –
- **BootstrapActions** (*list*) – A list of bootstrap actions that will be run before Hadoop is started on the cluster nodes.
 - (*dict*) –
Configuration of a bootstrap action.
 - * **Name** (*string*) – **[REQUIRED]**
The name of the bootstrap action.
 - * **ScriptBootstrapAction** (*dict*) – **[REQUIRED]**
The script run by the bootstrap action.
 - **Path** (*string*) – **[REQUIRED]**
Location of the script to run during a bootstrap action. Can be either a location in Amazon S3 or on a local file system.
 - **Args** (*list*) –
A list of command line arguments to pass to the bootstrap action script.
 - (*string*) –
- **SupportedProducts** (*list*) –

Note: For Amazon EMR releases 3.x and 2.x. For Amazon EMR releases 4.x and greater, use Applications.

A list of strings that indicates third-party software to use with the job flow. For more information, go to [Use Third Party Applications with Amazon EMR](#) . Currently supported values are:

- “mapr-m3” - launch the job flow using MapR M3 Edition.
- “mapr-m5” - launch the job flow using MapR M5 Edition.
- (*string*) –
- **NewSupportedProducts** (*list*) –

Note: For Amazon EMR releases 3.x and 2.x. For Amazon EMR releases 4.x and greater, use Applications.

A list of strings that indicates third-party software to use with the job flow that accepts a user argument list. EMR accepts and forwards the argument list to the corresponding installation script as bootstrap action arguments. For more information, see [Launch a Job Flow on the MapR Distribution for Hadoop](#) . Currently supported values are:

- “mapr-m3” - launch the cluster using MapR M3 Edition.

- “mapr-m5” - launch the cluster using MapR M5 Edition.
- “mapr” with the user arguments specifying “-edition,m3” or “-edition,m5” - launch the job flow using MapR M3 or M5 Edition respectively.
- “mapr-m7” - launch the cluster using MapR M7 Edition.
- “hunk” - launch the cluster with the Hunk Big Data Analytics Platform.
- “hue” - launch the cluster with Hue installed.
- “spark” - launch the cluster with Apache Spark installed.
- “ganglia” - launch the cluster with the Ganglia Monitoring System installed.
- (*dict*) -

The list of supported product configurations which allow user-supplied arguments. EMR accepts these arguments and forwards them to the corresponding installation script as bootstrap action arguments.

- * **Name** (*string*) -

The name of the product configuration.

- * **Args** (*list*) -

The list of user-supplied arguments.

- (*string*) -

- **Applications** (*list*) -

Note: Amazon EMR releases 4.x or later.

A list of applications for the cluster. Valid values are: “Hadoop”, “Hive”, “Mahout”, “Pig”, and “Spark.” They are case insensitive.

- (*dict*) -

An application is any Amazon or third-party software that you can add to the cluster. This structure contains a list of strings that indicates the software to use with the cluster and accepts a user argument list. Amazon EMR accepts and forwards the argument list to the corresponding installation script as bootstrap action argument. For more information, see [Launch a Job Flow on the MapR Distribution for Hadoop](#) . Currently supported values are:

- * “mapr-m3” - launch the job flow using MapR M3 Edition.
- * “mapr-m5” - launch the job flow using MapR M5 Edition.
- * “mapr” with the user arguments specifying “-edition,m3” or “-edition,m5” - launch the job flow using MapR M3 or M5 Edition, respectively.

Note: In Amazon EMR releases 4.0 and greater, the only accepted parameter is the application name. To pass arguments to applications, you supply a configuration for each application.

- * **Name** (*string*) -

The name of the application.

- * **Version** (*string*) -

The version of the application.

- * **Args** (*list*) -

Arguments for Amazon EMR to pass to the application.

- (*string*) -

* **AdditionalInfo** (*dict*) –

This option is for advanced users only. This is meta information about third-party applications that third-party vendors use for testing purposes.

- (*string*) –
- (*string*) –

• **Configurations** (*list*) –

Note: Amazon EMR releases 4.x or later.

The list of configurations supplied for the EMR cluster you are creating.

– (*dict*) –

Note: Amazon EMR releases 4.x or later.

Specifies a hardware and software configuration of the EMR cluster. This includes configurations for applications and software bundled with Amazon EMR. The Configuration object is a JSON object which is defined by a classification and a set of properties. Configurations can be nested, so a configuration may have its own Configuration objects listed.

* **Classification** (*string*) –

The classification of a configuration. For more information see, [Amazon EMR Configurations](#) .

* **Configurations** (*list*) –

A list of configurations you apply to this configuration object.

* **Properties** (*dict*) –

A set of properties supplied to the Configuration object.

- (*string*) –
- (*string*) –

- **VisibleToAllUsers** (*boolean*) – Whether the job flow is visible to all IAM users of the AWS account associated with the job flow. If this value is set to `true` , all IAM users of that AWS account can view and (if they have the proper policy permissions set) manage the job flow. If it is set to `false` , only the IAM user that created the job flow can view and manage it.
- **JobFlowRole** (*string*) – An IAM role for the job flow. The EC2 instances of the job flow assume this role. The default role is `EMRJobflowDefault` . In order to use the default role, you must have already created it using the CLI.
- **ServiceRole** (*string*) – The IAM role that will be assumed by the Amazon EMR service to access AWS resources on your behalf.
- **Tags** (*list*) – A list of tags to associate with a cluster and propagate to Amazon EC2 instances.

– (*dict*) –

A key/value pair containing user-defined metadata that you can associate with an Amazon EMR resource. Tags make it easier to associate clusters in various ways, such as grouping clusters to track your Amazon EMR resource allocation costs. For more information, see [Tagging Amazon EMR Resources](#) .

* **Key** (*string*) –

A user-defined key, which is the minimum required information for a valid tag. For more information, see [Tagging Amazon EMR Resources](#) .

* **Value** (*string*) –

A user-defined value, which is optional in a tag. For more information, see [Tagging Amazon EMR Resources](#).

Return type dict

Returns

Response Syntax

```
{
    'JobFlowId': 'string'
}
```

Response Structure

- (*dict*) –

The result of the RunJobFlow operation.

- **JobFlowId** (*string*) –

An unique identifier for the job flow.

set_termination_protection (***kwargs*)

SetTerminationProtection locks a job flow so the Amazon EC2 instances in the cluster cannot be terminated by user intervention, an API call, or in the event of a job-flow error. The cluster still terminates upon successful completion of the job flow. Calling SetTerminationProtection on a job flow is analogous to calling the Amazon EC2 DisableAPITermination API on all of the EC2 instances in a cluster.

SetTerminationProtection is used to prevent accidental termination of a job flow and to ensure that in the event of an error, the instances will persist so you can recover any data stored in their ephemeral instance storage.

To terminate a job flow that has been locked by setting SetTerminationProtection to `true`, you must first unlock the job flow by a subsequent call to SetTerminationProtection in which you set the value to `false`.

For more information, go to [Protecting a Job Flow from Termination](#) in the *Amazon Elastic MapReduce Developer's Guide*.

Request Syntax

```
response = client.set_termination_protection(
    JobFlowIds=[
        'string',
    ],
    TerminationProtected=True|False
)
```

Parameters

- **JobFlowIds** (*list*) – **[REQUIRED]**

A list of strings that uniquely identify the job flows to protect. This identifier is returned by RunJobFlow and can also be obtained from DescribeJobFlows.

- (*string*) –

- **TerminationProtected** (*boolean*) – **[REQUIRED]**

A Boolean that indicates whether to protect the job flow and prevent the Amazon EC2 instances in the cluster from shutting down due to API calls, user intervention, or job-flow error.

Returns None

set_visible_to_all_users (***kwargs*)

Sets whether all AWS Identity and Access Management (IAM) users under your account can access the specified job flows. This action works on running job flows. You can also set the visibility of a job flow when you launch it using the `VisibleToAllUsers` parameter of `RunJobFlow`. The `SetVisibleToAllUsers` action can be called only by an IAM user who created the job flow or the AWS account that owns the job flow.

Request Syntax

```
response = client.set_visible_to_all_users(
    JobFlowIds=[
        'string',
    ],
    VisibleToAllUsers=True|False
)
```

Parameters

- **JobFlowIds** (*list*) – [REQUIRED]

Identifiers of the job flows to receive the new visibility setting.

- (*string*) –

- **VisibleToAllUsers** (*boolean*) – [REQUIRED]

Whether the specified job flows are visible to all IAM users of the AWS account associated with the job flow. If this value is set to `True`, all IAM users of that AWS account can view and, if they have the proper IAM policy permissions set, manage the job flows. If it is set to `False`, only the IAM user that created a job flow can view and manage it.

Returns None

terminate_job_flows (***kwargs*)

`TerminateJobFlows` shuts a list of job flows down. When a job flow is shut down, any step not yet completed is canceled and the EC2 instances on which the job flow is running are stopped. Any log files not already saved are uploaded to Amazon S3 if a `LogUri` was specified when the job flow was created.

The maximum number of `JobFlows` allowed is 10. The call to `TerminateJobFlows` is asynchronous. Depending on the configuration of the job flow, it may take up to 5-20 minutes for the job flow to completely terminate and release allocated resources, such as Amazon EC2 instances.

Request Syntax

```
response = client.terminate_job_flows(
    JobFlowIds=[
        'string',
    ]
)
```

Parameters **JobFlowIds** (*list*) – [REQUIRED]

A list of job flows to be shutdown.

- (*string*) –

Returns None

Paginators

The available paginators are:

- `EMR.Paginator.ListBootstrapActions`

- `EMR.Paginator.ListClusters`
- `EMR.Paginator.ListInstanceGroups`
- `EMR.Paginator.ListInstances`
- `EMR.Paginator.ListSteps`

class `EMR.Paginator.ListBootstrapActions`

```
paginator = client.get_paginator('list_bootstrap_actions')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `EMR.Client.list_bootstrap_actions()`.

Request Syntax

```
response_iterator = paginator.paginate(
    ClusterId='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **ClusterId** (*string*) – **[REQUIRED]**

The cluster identifier for the bootstrap actions to list .

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
    'BootstrapActions': [
        {
            'Name': 'string',
            'ScriptPath': 'string',
            'Args': [
                'string',
            ]
        },
    ],
}
```

```
    ],
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

This output contains the bootstrap actions detail .

- **BootstrapActions** (*list*) –

The bootstrap actions associated with the cluster .

- * (*dict*) –

An entity describing an executable that runs on a cluster.

- **Name** (*string*) –

The name of the command.

- **ScriptPath** (*string*) –

The Amazon S3 location of the command script.

- **Args** (*list*) –

Arguments for Amazon EMR to pass to the command for execution.

- (*string*) –

- **NextToken** (*string*) –

A token to resume pagination.

class EMR.Paginator.**ListClusters**

```
paginator = client.get_paginator('list_clusters')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `EMR.Client.list_clusters()`.

Request Syntax

```
response_iterator = paginator.paginate(
    CreatedAfter=datetime(2015, 1, 1),
    CreatedBefore=datetime(2015, 1, 1),
    ClusterStates=[
        'STARTING' | 'BOOTSTRAPPING' | 'RUNNING' | 'WAITING' | 'TERMINATING' | 'TERMINATED' | 'TERMINATING',
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **CreatedAfter** (*datetime*) – The creation date and time beginning value filter for listing clusters .
- **CreatedBefore** (*datetime*) – The creation date and time end value filter for listing clusters .
- **ClusterStates** (*list*) – The cluster state filters to apply when listing clusters.
 - (*string*) –

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Clusters': [
    {
      'Id': 'string',
      'Name': 'string',
      'Status': {
        'State': 'STARTING'|'BOOTSTRAPPING'|'RUNNING'|'WAITING'|'TERMINATING',
        'StateChangeReason': {
          'Code': 'INTERNAL_ERROR'|'VALIDATION_ERROR'|'INSTANCE_FAILURE',
          'Message': 'string'
        },
        'Timeline': {
          'CreationDateTime': datetime(2015, 1, 1),
          'ReadyDateTime': datetime(2015, 1, 1),
          'EndDateTime': datetime(2015, 1, 1)
        }
      },
      'NormalizedInstanceHours': 123
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
This contains a `ClusterSummaryList` with the cluster details; for example, the cluster IDs, names, and status.
 - **Clusters** (*list*) –
The list of clusters for the account based on the given filters.
 - * (*dict*) –
The summary description of the cluster.
 - **Id** (*string*) –
The unique identifier for the cluster.
 - **Name** (*string*) –
The name of the cluster.

- **Status** (*dict*) –
The details about the current status of the cluster.
- **State** (*string*) –
The current state of the cluster.
- **StateChangeReason** (*dict*) –
The reason for the cluster status change.
- **Code** (*string*) –
The programmatic code for the state change reason.
- **Message** (*string*) –
The descriptive message for the state change reason.
- **Timeline** (*dict*) –
A timeline that represents the status of a cluster over the life-time of the cluster.
- **CreationDateTime** (*datetime*) –
The creation date and time of the cluster.
- **ReadyDateTime** (*datetime*) –
The date and time when the cluster was ready to execute steps.
- **EndTime** (*datetime*) –
The date and time when the cluster was terminated.
- **NormalizedInstanceHours** (*integer*) –
An approximation of the cost of the job flow, represented in m1.small/hours. This value is incremented one time for every hour an m1.small instance runs. Larger instances are weighted more, so an EC2 instance that is roughly four times more expensive would result in the normalized instance hours being incremented by four. This result is only an approximation and does not reflect the actual billing rate.

– **NextToken** (*string*) –

A token to resume pagination.

class EMR.Paginator.**ListInstanceGroups**

```
paginator = client.get_paginator('list_instance_groups')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `EMR.Client.list_instance_groups()`.

Request Syntax

```
response_iterator = paginator.paginate(
    ClusterId='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **ClusterId** (*string*) – **[REQUIRED]**

The identifier of the cluster for which to list the instance groups.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns**Response Syntax**

```
{
  'InstanceGroups': [
    {
      'Id': 'string',
      'Name': 'string',
      'Market': 'ON_DEMAND' | 'SPOT',
      'InstanceGroupType': 'MASTER' | 'CORE' | 'TASK',
      'BidPrice': 'string',
      'InstanceType': 'string',
      'RequestedInstanceCount': 123,
      'RunningInstanceCount': 123,
      'Status': {
        'State': 'PROVISIONING' | 'BOOTSTRAPPING' | 'RUNNING' | 'RESIZING' | 'SUSPENDED',
        'StateChangeReason': {
          'Code': 'INTERNAL_ERROR' | 'VALIDATION_ERROR' | 'INSTANCE_FAILURE',
          'Message': 'string'
        },
      },
      'Timeline': {
        'CreationDateTime': datetime(2015, 1, 1),
        'ReadyDateTime': datetime(2015, 1, 1),
        'EndDateTime': datetime(2015, 1, 1)
      },
    },
    {
      'Configurations': [
        {
          'Classification': 'string',
          'Configurations': {'... recursive ...'},
          'Properties': {
            'string': 'string'
          },
        },
      ],
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

This input determines which instance groups to retrieve.

- **InstanceGroups** (*list*) –

The list of instance groups for the cluster and given filters.

- * (*dict*) –

This entity represents an instance group, which is a group of instances that have common purpose. For example, CORE instance group is used for HDFS.

- **Id** (*string*) –

The identifier of the instance group.

- **Name** (*string*) –

The name of the instance group.

- **Market** (*string*) –

The marketplace to provision instances for this group. Valid values are ON_DEMAND or SPOT.

- **InstanceGroupType** (*string*) –

The type of the instance group. Valid values are MASTER, CORE or TASK.

- **BidPrice** (*string*) –

The bid price for each EC2 instance in the instance group when launching nodes as Spot Instances, expressed in USD.

- **InstanceType** (*string*) –

The EC2 instance type for all instances in the instance group.

- **RequestedInstanceCount** (*integer*) –

The target number of instances for the instance group.

- **RunningInstanceCount** (*integer*) –

The number of instances currently running in this instance group.

- **Status** (*dict*) –

The current status of the instance group.

- **State** (*string*) –

The current state of the instance group.

- **StateChangeReason** (*dict*) –

The status change reason details for the instance group.

- **Code** (*string*) –

The programmable code for the state change reason.

- **Message** (*string*) –

The status change reason description.

- **Timeline** (*dict*) –

The timeline of the instance group status over time.

- **CreationDateTime** (*datetime*) –

The creation date and time of the instance group.

- **ReadyDateTime** (*datetime*) –

The date and time when the instance group became ready to perform tasks.

- **EndTime** (*datetime*) –

The date and time when the instance group terminated.

- **Configurations** (*list*) –

Note: Amazon EMR releases 4.x or later.

The list of configurations supplied for an EMR cluster instance group. You can specify a separate configuration for each instance group (master, core, and task).

- (*dict*) –

Note: Amazon EMR releases 4.x or later.

Specifies a hardware and software configuration of the EMR cluster. This includes configurations for applications and software bundled with Amazon EMR. The Configuration object is a JSON object which is defined by a classification and a set of properties. Configurations can be nested, so a configuration may have its own Configuration objects listed.

- **Classification** (*string*) –

The classification of a configuration. For more information see, [Amazon EMR Configurations](#).

- **Configurations** (*list*) –

A list of configurations you apply to this configuration object.

- **Properties** (*dict*) –

A set of properties supplied to the Configuration object.

- (*string*) –

- (*string*) –

- **NextToken** (*string*) –

A token to resume pagination.

class EMR.Paginator.**ListInstances**

```
paginator = client.get_paginator('list_instances')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `EMR.Client.list_instances()`.

Request Syntax

```
response_iterator = paginator.paginate(
    ClusterId='string',
    InstanceGroupId='string',
    InstanceGroupTypes=[
        'MASTER'|'CORE'|'TASK',
    ],
    PaginationConfig={
        'MaxItems': 123,
```

```

        'PageSize': 123,
        'StartingToken': 'string'
    }
)

```

Parameters

- **ClusterId** (*string*) – [REQUIRED]

The identifier of the cluster for which to list the instances.

- **InstanceGroupId** (*string*) – The identifier of the instance group for which to list the instances.

- **InstanceGroupTypes** (*list*) – The type of instance group for which to list the instances.

- (*string*) –

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```

{
    'Instances': [
        {
            'Id': 'string',
            'Ec2InstanceId': 'string',
            'PublicDnsName': 'string',
            'PublicIpAddress': 'string',
            'PrivateDnsName': 'string',
            'PrivateIpAddress': 'string',
            'Status': {
                'State': 'AWAITING_FULFILLMENT'|'PROVISIONING'|'BOOTSTRAPPING'|'RUNNING'|'STOPPING'|'TERMINATING'|'SHUTTING_DOWN'|'FAILED'|'PENDING'|'UNKNOWN',
                'StateChangeReason': {
                    'Code': 'INTERNAL_ERROR'|'VALIDATION_ERROR'|'INSTANCE_FAILURE'|'USER_REQUEST',
                    'Message': 'string'
                }
            },
            'Timeline': {
                'CreationDateTime': datetime(2015, 1, 1),
                'ReadyDateTime': datetime(2015, 1, 1),
                'EndDateTime': datetime(2015, 1, 1)
            }
        }
    ],
    'NextToken': 'string'
}

```

Response Structure

- (*dict*) –

This output contains the list of instances.

- **Instances** (*list*) –

The list of instances for the cluster and given filters.

- * (*dict*) –

Represents an EC2 instance provisioned as part of cluster.

- **Id** (*string*) –

The unique identifier for the instance in Amazon EMR.

- **Ec2InstanceId** (*string*) –

The unique identifier of the instance in Amazon EC2.

- **PublicDnsName** (*string*) –

The public DNS name of the instance.

- **PublicIpAddress** (*string*) –

The public IP address of the instance.

- **PrivateDnsName** (*string*) –

The private DNS name of the instance.

- **PrivateIpAddress** (*string*) –

The private IP address of the instance.

- **Status** (*dict*) –

The current status of the instance.

- **State** (*string*) –

The current state of the instance.

- **StateChangeReason** (*dict*) –

The details of the status change reason for the instance.

- **Code** (*string*) –

The programmable code for the state change reason.

- **Message** (*string*) –

The status change reason description.

- **Timeline** (*dict*) –

The timeline of the instance status over time.

- **CreationDateTime** (*datetime*) –

The creation date and time of the instance.

- **ReadyDateTime** (*datetime*) –

The date and time when the instance was ready to perform tasks.

- **EndDateTime** (*datetime*) –

The date and time when the instance was terminated.

- **NextToken** (*string*) –

A token to resume pagination.

class EMR.Paginator.**ListSteps**

```
paginator = client.get_paginator('list_steps')
```

paginate (**kwargs)

Creates an iterator that will paginate through responses from `EMR.Client.list_steps()`.

Request Syntax

```
response_iterator = paginator.paginate(
    ClusterId='string',
    StepStates=[
        'PENDING' | 'RUNNING' | 'COMPLETED' | 'CANCELLED' | 'FAILED' | 'INTERRUPTED',
    ],
    StepIds=[
        'string',
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **ClusterId** (*string*) – **[REQUIRED]**
The identifier of the cluster for which to list the steps.
- **StepStates** (*list*) – The filter to limit the step list based on certain states.
 - (*string*) –
- **StepIds** (*list*) – The filter to limit the step list based on the identifier of the steps.
 - (*string*) –
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Steps': [
    {
      'Id': 'string',
      'Name': 'string',
      'Config': {
        'Jar': 'string',
        'Properties': {
          'string': 'string'
        }
      }
    }
  ]
}
```

```
    },
    'MainClass': 'string',
    'Args': [
        'string',
    ]
},
'ActionOnFailure': 'TERMINATE_JOB_FLOW' | 'TERMINATE_CLUSTER' | 'CANCEL_AN
'Status': {
    'State': 'PENDING' | 'RUNNING' | 'COMPLETED' | 'CANCELLED' | 'FAILED' | 'INT
    'StateChangeReason': {
        'Code': 'NONE',
        'Message': 'string'
    },
    'Timeline': {
        'CreationDateTime': datetime(2015, 1, 1),
        'StartDateTime': datetime(2015, 1, 1),
        'EndDateTime': datetime(2015, 1, 1)
    }
}
    },
],
'NextToken': 'string'
}
```

Response Structure

- *(dict)* –

This output contains the list of steps.

- **Steps** (*list*) –

The filtered list of steps for the cluster.

- * *(dict)* –

The summary of the cluster step.

- **Id** (*string*) –

The identifier of the cluster step.

- **Name** (*string*) –

The name of the cluster step.

- **Config** (*dict*) –

The Hadoop job configuration of the cluster step.

- **Jar** (*string*) –

The path to the JAR file that runs during the step.

- **Properties** (*dict*) –

The list of Java properties that are set when the step runs. You can use these properties to pass key value pairs to your main function.

- (*string*) –

- (*string*) –

- **MainClass** (*string*) –

The name of the main class in the specified Java file. If not specified, the JAR file should specify a main class in its manifest file.

- **Args** (*list*) –

The list of command line arguments to pass to the JAR file's main function for execution.

- *(string)* –
- **ActionOnFailure** (*string*) –

This specifies what action to take when the cluster step fails. Possible values are TERMINATE_CLUSTER, CANCEL_AND_WAIT, and CONTINUE.

- **Status** (*dict*) –

The current execution status details of the cluster step.

- **State** (*string*) –

The execution state of the cluster step.

- **StateChangeReason** (*dict*) –

The reason for the step execution status change.

- **Code** (*string*) –

The programmable code for the state change reason. Note: Currently, the service provides no code for the state change.

- **Message** (*string*) –

The descriptive message for the state change reason.

- **Timeline** (*dict*) –

The timeline of the cluster step status over time.

- **CreationDateTime** (*datetime*) –

The date and time when the cluster step was created.

- **StartDateTime** (*datetime*) –

The date and time when the cluster step execution started.

- **EndDateTime** (*datetime*) –

The date and time when the cluster step execution completed or failed.

- **NextToken** (*string*) –

A token to resume pagination.

Waiters

The available waiters are:

- `EMR.Waiter.ClusterRunning`

class `EMR.Waiter.ClusterRunning`

```
waiter = client.get_waiter('cluster_running')
```

wait (***kwargs*)

Polls `EMR.Client.describe_cluster()` every 30 seconds until a successful state is reached. An error is returned after 60 failed checks.

Request Syntax

```
waiter.wait(  
    ClusterId='string'  
)
```

Parameters `ClusterId` (*string*) – [REQUIRED]

The identifier of the cluster to describe.

Returns None

Glacier

Table of Contents

- Glacier
 - Client
 - Service Resource
 - Account
 - Archive
 - Job
 - MultipartUpload
 - Notification
 - Vault

Client

`class` `Glacier.Client`

A low-level client representing Amazon Glacier:

```
import boto3  
  
client = boto3.client('glacier')
```

These are the available methods:

- `abort_multipart_upload()`
- `abort_vault_lock()`
- `add_tags_to_vault()`
- `can_paginate()`
- `complete_multipart_upload()`
- `complete_vault_lock()`
- `create_vault()`
- `delete_archive()`
- `delete_vault()`
- `delete_vault_access_policy()`
- `delete_vault_notifications()`
- `describe_job()`
- `describe_vault()`
- `generate_presigned_url()`
- `get_data_retrieval_policy()`
- `get_job_output()`
- `get_paginator()`
- `get_vault_access_policy()`
- `get_vault_lock()`

```

•get_vault_notifications()
•get_waiter()
•initiate_job()
•initiate_multipart_upload()
•initiate_vault_lock()
•list_jobs()
•list_multipart_uploads()
•list_parts()
•list_tags_for_vault()
•list_vaults()
•remove_tags_from_vault()
•set_data_retrieval_policy()
•set_vault_access_policy()
•set_vault_notifications()
•upload_archive()
•upload_multipart_part()
abort_multipart_upload (**kwargs)

```

This operation aborts a multipart upload identified by the upload ID.

After the Abort Multipart Upload request succeeds, you cannot upload any more parts to the multipart upload or complete the multipart upload. Aborting a completed upload fails. However, aborting an already-aborted upload will succeed, for a short time. For more information about uploading a part and completing a multipart upload, see [UploadMultipartPart](#) and [CompleteMultipartUpload](#).

This operation is idempotent.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#).

For conceptual information and underlying REST API, go to [Working with Archives in Amazon Glacier](#) and [Abort Multipart Upload](#) in the *Amazon Glacier Developer Guide*.

Request Syntax

```

response = client.abort_multipart_upload(
    vaultName='string',
    uploadId='string'
)

```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos“-'” apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **vaultName** (*string*) – **[REQUIRED]**

The name of the vault.

- **uploadId** (*string*) – **[REQUIRED]**

The upload ID of the multipart upload to delete.

Returns None

abort_vault_lock (**kwargs)

This operation aborts the vault locking process if the vault lock is not in the `Locked` state. If the vault lock is in the `Locked` state when this operation is requested, the operation returns an `AccessDeniedException` error. Aborting the vault locking process removes the vault lock policy from the specified vault.

A vault lock is put into the `InProgress` state by calling `InitiateVaultLock`. A vault lock is put into the `Locked` state by calling `CompleteVaultLock`. You can get the state of a vault lock by calling `GetVaultLock`. For more information about the vault locking process, see [Amazon Glacier Vault Lock](#). For more information about vault lock policies, see [Amazon Glacier Access Control with Vault Lock Policies](#).

This operation is idempotent. You can successfully invoke this operation multiple times, if the vault lock is in the `InProgress` state or if there is no policy associated with the vault.

Request Syntax

```
response = client.abort_vault_lock(
    vaultName='string'
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID. This value must match the AWS account ID associated with the credentials used to sign the request. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you specify your account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **vaultName** (*string*) – **[REQUIRED]**

The name of the vault.

Returns None

add_tags_to_vault (**kwargs)

This operation adds the specified tags to a vault. Each tag is composed of a key and a value. Each vault can have up to 10 tags. If your request would cause the tag limit for the vault to be exceeded, the operation throws the `LimitExceededException` error. If a tag already exists on the vault under a specified key, the existing key value will be overwritten. For more information about tags, see [Tagging Amazon Glacier Resources](#).

Request Syntax

```
response = client.add_tags_to_vault(
    vaultName='string',
    Tags={
        'string': 'string'
    }
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **vaultName** (*string*) – [REQUIRED]

The name of the vault.

- **Tags** (*dict*) – The tags to add to the vault. Each tag is composed of a key and a value. The value can be an empty string.
 - (*string*) –
 - * (*string*) –

Returns None

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

complete_multipart_upload (***kwargs*)

You call this operation to inform Amazon Glacier that all the archive parts have been uploaded and that Amazon Glacier can now assemble the archive from the uploaded parts. After assembling and saving the archive to the vault, Amazon Glacier returns the URI path of the newly created archive resource. Using the URI path, you can then access the archive. After you upload an archive, you should save the archive ID returned to retrieve the archive at a later point. You can also get the vault inventory to obtain a list of archive IDs in a vault. For more information, see `InitiateJob`.

In the request, you must include the computed SHA256 tree hash of the entire archive you have uploaded. For information about computing a SHA256 tree hash, see [Computing Checksums](#). On the server side, Amazon Glacier also constructs the SHA256 tree hash of the assembled archive. If the values match, Amazon Glacier saves the archive to the vault; otherwise, it returns an error, and the operation fails. The `ListParts` operation returns a list of parts uploaded for a specific multipart upload. It includes checksum information for each uploaded part that can be used to debug a bad checksum issue.

Additionally, Amazon Glacier also checks for any missing content ranges when assembling the archive, if missing content ranges are found, Amazon Glacier returns an error and the operation fails.

Complete Multipart Upload is an idempotent operation. After your first successful complete multipart upload, if you call the operation again within a short period, the operation will succeed and return the same archive ID. This is useful in the event you experience a network issue that causes an aborted connection or receive a 500 server error, in which case you can repeat your Complete Multipart Upload request and get the same archive ID without creating duplicate archives. Note, however, that after the multipart upload completes, you cannot call the `List Parts` operation and the multipart upload will not appear in `List Multipart Uploads` response, even if idempotent complete is possible.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#).

For conceptual information and underlying REST API, go to [Uploading Large Archives in Parts \(Multipart Upload\)](#) and [Complete Multipart Upload](#) in the *Amazon Glacier Developer Guide*.

Request Syntax

```
response = client.complete_multipart_upload(
    vaultName='string',
    uploadId='string',
    archiveSize='string',
```

)

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos`'-'` apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to `"-"` by default if no value is not specified.

- **vaultName** (*string*) – [REQUIRED]

The name of the vault.

- **uploadId** (*string*) – [REQUIRED]

The upload ID of the multipart upload.

- **archiveSize** (*string*) – The total size, in bytes, of the entire archive. This value should be the sum of all the sizes of the individual parts that you uploaded.
- **checksum** (*string*) – The SHA256 tree hash of the entire archive. It is the tree hash of SHA256 tree hash of the individual parts. If the value you specify in the request does not match the SHA256 tree hash of the final assembled archive as computed by Amazon Glacier, Amazon Glacier returns an error and the request fails.

Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required

Return type dict**Returns****Response Syntax**

```
{
    'location': 'string',
    'checksum': 'string',
    'archiveId': 'string'
}
```

Response Structure

- (*dict*) –

Contains the Amazon Glacier response to your request.

For information about the underlying REST API, go to [Upload Archive](#) . For conceptual information, go to [Working with Archives in Amazon Glacier](#) .

- **location** (*string*) –

The relative URI path of the newly added archive resource.

- **checksum** (*string*) –

The checksum of the archive computed by Amazon Glacier.

- **archiveId** (*string*) –

The ID of the archive. This value is also included as part of the location.

complete_vault_lock (***kwargs*)

This operation completes the vault locking process by transitioning the vault lock from the `InProgress` state to the `Locked` state, which causes the vault lock policy to become unchangeable. A vault lock is put into the `InProgress` state by calling `InitiateVaultLock` . You can obtain the state of the vault lock

by calling `GetVaultLock`. For more information about the vault locking process, see [Amazon Glacier Vault Lock](#).

This operation is idempotent. This request is always successful if the vault lock is in the `Locked` state and the provided lock ID matches the lock ID originally used to lock the vault.

If an invalid lock ID is passed in the request when the vault lock is in the `Locked` state, the operation returns an `AccessDeniedException` error. If an invalid lock ID is passed in the request when the vault lock is in the `InProgress` state, the operation throws an `InvalidParameter` error.

Request Syntax

```
response = client.complete_vault_lock(
    vaultName='string',
    lockId='string'
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID. This value must match the AWS account ID associated with the credentials used to sign the request. You can either specify an AWS account ID or optionally a single apos`“-“` apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you specify your account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to `“-”` by default if no value is not specified.

- **vaultName** (*string*) – **[REQUIRED]**

The name of the vault.

- **lockId** (*string*) – **[REQUIRED]**

The `lockId` value is the lock ID obtained from a `InitiateVaultLock` request.

Returns None

`create_vault` (***kwargs*)

This operation creates a new vault with the specified name. The name of the vault must be unique within a region for an AWS account. You can create up to 1,000 vaults per account. If you need to create more vaults, contact Amazon Glacier.

You must use the following guidelines when naming a vault.

- Names can be between 1 and 255 characters long.
- Allowed characters are a-z, A-Z, 0-9, `_` (underscore), `-` (hyphen), and `.` (period).

This operation is idempotent.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#).

For conceptual information and underlying REST API, go to [Creating a Vault in Amazon Glacier](#) and [Create Vault](#) in the *Amazon Glacier Developer Guide*.

Request Syntax

```
response = client.create_vault(
    vaultName='string'
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID. This value must match the AWS account ID associated with the credentials used to sign the request. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you specify your account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **vaultName** (*string*) – [REQUIRED]

The name of the vault.

Return type dict

Returns

Response Syntax

```
{
    'location': 'string'
}
```

Response Structure

- (*dict*) –

Contains the Amazon Glacier response to your request.

- **location** (*string*) –

The URI of the vault that was created.

delete_archive (***kwargs*)

This operation deletes an archive from a vault. Subsequent requests to initiate a retrieval of this archive will fail. Archive retrievals that are in progress for this archive ID may or may not succeed according to the following scenarios:

- If the archive retrieval job is actively preparing the data for download when Amazon Glacier receives the delete archive request, the archival retrieval operation might fail.
- If the archive retrieval job has successfully prepared the archive for download when Amazon Glacier receives the delete archive request, you will be able to download the output.

This operation is idempotent. Attempting to delete an already-deleted archive does not result in an error.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don’t have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#) .

For conceptual information and underlying REST API, go to [Deleting an Archive in Amazon Glacier](#) and [Delete Archive](#) in the *Amazon Glacier Developer Guide* .

Request Syntax

```
response = client.delete_archive(
    vaultName='string',
    archiveId='string'
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **vaultName** (*string*) – [REQUIRED]

The name of the vault.

- **archiveId** (*string*) – [REQUIRED]

The ID of the archive to delete.

Returns None

delete_vault (***kwargs*)

This operation deletes a vault. Amazon Glacier will delete a vault only if there are no archives in the vault as of the last inventory and there have been no writes to the vault since the last inventory. If either of these conditions is not satisfied, the vault deletion fails (that is, the vault is not removed) and Amazon Glacier returns an error. You can use [DescribeVault](#) to return the number of archives in a vault, and you can use [Initiate a Job \(POST jobs\)](#) to initiate a new inventory retrieval for a vault. The inventory contains the archive IDs you use to delete archives using [Delete Archive \(DELETE archive\)](#) .

This operation is idempotent.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don’t have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#) .

For conceptual information and underlying REST API, go to [Deleting a Vault in Amazon Glacier and Delete Vault](#) in the *Amazon Glacier Developer Guide* .

Request Syntax

```
response = client.delete_vault(
    vaultName='string'
)
```

Parameters

- **accountId** (*string*) – The AccountId value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **vaultName** (*string*) – [REQUIRED]

The name of the vault.

Returns None

delete_vault_access_policy (***kwargs*)

This operation deletes the access policy associated with the specified vault. The operation is eventually consistent; that is, it might take some time for Amazon Glacier to completely remove the access policy, and you might still see the effect of the policy for a short time after you send the delete request.

This operation is idempotent. You can invoke delete multiple times, even if there is no policy associated with the vault. For more information about vault access policies, see [Amazon Glacier Access Control with Vault Access Policies](#) .

Request Syntax

```
response = client.delete_vault_access_policy(  
    vaultName='string'  
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **vaultName** (*string*) – **[REQUIRED]**

The name of the vault.

Returns None

delete_vault_notifications (***kwargs*)

This operation deletes the notification configuration set for a vault. The operation is eventually consistent; that is, it might take some time for Amazon Glacier to completely disable the notifications and you might still receive some notifications for a short time after you send the delete request.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don’t have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#) .

For conceptual information and underlying REST API, go to [Configuring Vault Notifications in Amazon Glacier](#) and [Delete Vault Notification Configuration](#) in the Amazon Glacier Developer Guide.

Request Syntax

```
response = client.delete_vault_notifications(  
    vaultName='string'  
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **vaultName** (*string*) – **[REQUIRED]**

The name of the vault.

Returns None

describe_job (***kwargs*)

This operation returns information about a job you previously initiated, including the job initiation date, the user who initiated the job, the job status code/message and the Amazon SNS topic to notify after Amazon Glacier completes the job. For more information about initiating a job, see [InitiateJob](#) .

Note: This operation enables you to check the status of your job. However, it is strongly recommended that you set up an Amazon SNS topic and specify it in your initiate job request so that Amazon Glacier can notify the topic after it completes the job.

A job ID will not expire for at least 24 hours after Amazon Glacier completes the job.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#).

For information about the underlying REST API, go to [Working with Archives in Amazon Glacier](#) in the *Amazon Glacier Developer Guide*.

Request Syntax

```
response = client.describe_job(
    vaultName='string',
    jobId='string'
)
```

Parameters

- **accountId** (*string*) – The AccountId value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **vaultName** (*string*) – **[REQUIRED]**

The name of the vault.

- **jobId** (*string*) – **[REQUIRED]**

The ID of the job to describe.

Return type dict

Returns

Response Syntax

```
{
    'JobId': 'string',
    'JobDescription': 'string',
    'Action': 'ArchiveRetrieval'|'InventoryRetrieval',
    'ArchiveId': 'string',
    'VaultARN': 'string',
    'CreationDate': 'string',
    'Completed': True|False,
    'StatusCode': 'InProgress'|'Succeeded'|'Failed',
    'StatusMessage': 'string',
    'ArchiveSizeInBytes': 123,
    'InventorySizeInBytes': 123,
    'SNSTopic': 'string',
    'CompletionDate': 'string',
    'SHA256TreeHash': 'string',
    'ArchiveSHA256TreeHash': 'string',
    'RetrievalByteRange': 'string',
    'InventoryRetrievalParameters': {
        'Format': 'string',
        'StartDate': 'string',
        'EndDate': 'string',
        'Limit': 'string',
        'Marker': 'string'
    }
}
```

```
}  
}
```

Response Structure

- (*dict*) –

Describes an Amazon Glacier job.

- **JobId** (*string*) –

An opaque string that identifies an Amazon Glacier job.

- **JobDescription** (*string*) –

The job description you provided when you initiated the job.

- **Action** (*string*) –

The job type. It is either `ArchiveRetrieval` or `InventoryRetrieval`.

- **ArchiveId** (*string*) –

For an `ArchiveRetrieval` job, this is the archive ID requested for download. Otherwise, this field is null.

- **VaultARN** (*string*) –

The Amazon Resource Name (ARN) of the vault from which the archive retrieval was requested.

- **CreationDate** (*string*) –

The UTC date when the job was created. A string representation of ISO 8601 date format, for example, “2012-03-20T17:03:43.221Z”.

- **Completed** (*boolean*) –

The job status. When a job is completed, you get the job’s output.

- **StatusCode** (*string*) –

The status code can be `InProgress`, `Succeeded`, or `Failed`, and indicates the status of the job.

- **StatusMessage** (*string*) –

A friendly message that describes the job status.

- **ArchiveSizeInBytes** (*integer*) –

For an `ArchiveRetrieval` job, this is the size in bytes of the archive being requested for download. For the `InventoryRetrieval` job, the value is null.

- **InventorySizeInBytes** (*integer*) –

For an `InventoryRetrieval` job, this is the size in bytes of the inventory requested for download. For the `ArchiveRetrieval` job, the value is null.

- **SNSTopic** (*string*) –

An Amazon Simple Notification Service (Amazon SNS) topic that receives notification.

- **CompletionDate** (*string*) –

The UTC time that the archive retrieval request completed. While the job is in progress, the value will be null.

- **SHA256TreeHash** (*string*) –

For an `ArchiveRetrieval` job, it is the checksum of the archive. Otherwise, the value is null.

The SHA256 tree hash value for the requested range of an archive. If the `Initiate a Job` request for an archive specified a tree-hash aligned range, then this field returns a value.

For the specific case when the whole archive is retrieved, this value is the same as the `ArchiveSHA256TreeHash` value.

This field is null in the following situations:

- * Archive retrieval jobs that specify a range that is not tree-hash aligned.
- * Archival jobs that specify a range that is equal to the whole archive and the job status is `InProgress`.
- * Inventory jobs.

– **ArchiveSHA256TreeHash** (*string*) –

The SHA256 tree hash of the entire archive for an archive retrieval. For inventory retrieval jobs, this field is null.

– **RetrievalByteRange** (*string*) –

The retrieved byte range for archive retrieval jobs in the form “*StartByteValue -EndByteValue*”. If no range was specified in the archive retrieval, then the whole archive is retrieved and *StartByteValue* equals 0 and *EndByteValue* equals the size of the archive minus 1. For inventory retrieval jobs this field is null.

– **InventoryRetrievalParameters** (*dict*) –

Parameters used for range inventory retrieval.

* **Format** (*string*) –

The output format for the vault inventory list, which is set by the **InitiateJob** request when initiating a job to retrieve a vault inventory. Valid values are “CSV” and “JSON”.

* **StartDate** (*string*) –

The start of the date range in UTC for vault inventory retrieval that includes archives created on or after this date. A string representation of ISO 8601 date format, for example, 2013-03-20T17:03:43Z.

* **EndDate** (*string*) –

The end of the date range in UTC for vault inventory retrieval that includes archives created before this date. A string representation of ISO 8601 date format, for example, 2013-03-20T17:03:43Z.

* **Limit** (*string*) –

Specifies the maximum number of inventory items returned per vault inventory retrieval request. This limit is set when initiating the job with the a **InitiateJob** request.

* **Marker** (*string*) –

An opaque string that represents where to continue pagination of the vault inventory retrieval results. You use the marker in a new **InitiateJob** request to obtain additional inventory items. If there are no more inventory items, this value is `null`. For more information, see [Range Inventory Retrieval](#).

describe_vault (***kwargs*)

This operation returns information about a vault, including the vault’s Amazon Resource Name (ARN), the date the vault was created, the number of archives it contains, and the total size of all the archives in the vault. The number of archives and their total size are as of the last inventory generation. This means that if you add or remove an archive from a vault, and then immediately use `Describe Vault`, the change in contents will not be immediately reflected. If you want to retrieve the latest inventory of the vault, use `InitiateJob`. Amazon Glacier generates vault inventories approximately daily. For more information, see [Downloading a Vault Inventory in Amazon Glacier](#).

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#) .

For conceptual information and underlying REST API, go to [Retrieving Vault Metadata in Amazon Glacier](#) and [Describe Vault](#) in the *Amazon Glacier Developer Guide* .

Request Syntax

```
response = client.describe_vault(
    vaultName='string'
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apostrophe (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apostrophes) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **vaultName** (*string*) – **[REQUIRED]**

The name of the vault.

Return type dict

Returns

Response Syntax

```
{
    'VaultARN': 'string',
    'VaultName': 'string',
    'CreationDate': 'string',
    'LastInventoryDate': 'string',
    'NumberOfArchives': 123,
    'SizeInBytes': 123
}
```

Response Structure

- (*dict*) –

Contains the Amazon Glacier response to your request.

- **VaultARN** (*string*) –

The Amazon Resource Name (ARN) of the vault.

- **VaultName** (*string*) –

The name of the vault.

- **CreationDate** (*string*) –

The UTC date when the vault was created. A string representation of ISO 8601 date format, for example, “2012-03-20T17:03:43.221Z”.

- **LastInventoryDate** (*string*) –

The UTC date when Amazon Glacier completed the last vault inventory. A string representation of ISO 8601 date format, for example, “2012-03-20T17:03:43.221Z”.

– **NumberOfArchives** (*integer*) –

The number of archives in the vault as of the last inventory date. This field will return `null` if an inventory has not yet run on the vault, for example, if you just created the vault.

– **SizeInBytes** (*integer*) –

Total size, in bytes, of the archives in the vault as of the last inventory date. This field will return `null` if an inventory has not yet run on the vault, for example, if you just created the vault.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to *ClientMethod*.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_data_retrieval_policy (***kwargs*)

This operation returns the current data retrieval policy for the account and region specified in the GET request. For more information about data retrieval policies, see [Amazon Glacier Data Retrieval Policies](#).

Request Syntax

```
response = client.get_data_retrieval_policy(
    )
```

Parameters **accountId** (*string*) – The `AccountId` value is the AWS account ID. This value must match the AWS account ID associated with the credentials used to sign the request. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you specify your account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

Return type dict

Returns

Response Syntax

```
{
    'Policy': {
        'Rules': [
            {
                'Strategy': 'string',
                'BytesPerHour': 123
            },
        ],
    }
}
```

Response Structure

- (*dict*) –

Contains the Amazon Glacier response to the `GetDataRetrievalPolicy` request.

– **Policy** (*dict*) –

Contains the returned data retrieval policy in JSON format.

* **Rules** (*list*) –

The policy rule. Although this is a list type, currently there must be only one rule, which contains a `Strategy` field and optionally a `BytesPerHour` field.

· (*dict*) –

Data retrieval policy rule.

· **Strategy** (*string*) –

The type of data retrieval policy to set.

Valid values: `BytesPerHour`|`FreeTier`|`None`

· **BytesPerHour** (*integer*) –

The maximum number of bytes that can be retrieved in an hour.

This field is required only if the value of the `Strategy` field is `BytesPerHour`. Your `PUT` operation will be rejected if the `Strategy` field is not set to `BytesPerHour` and you set this field.

get_job_output (***kwargs*)

This operation downloads the output of the job you initiated using `InitiateJob`. Depending on the job type you specified when you initiated the job, the output will be either the content of an archive or a vault inventory.

A job ID will not expire for at least 24 hours after Amazon Glacier completes the job. That is, you can download the job output within the 24 hours period after Amazon Glacier completes the job.

If the job output is large, then you can use the `Range` request header to retrieve a portion of the output. This allows you to download the entire output in smaller chunks of bytes. For example, suppose you have 1 GB of job output you want to download and you decide to download 128 MB chunks of data at a time, which is a total of eight Get Job Output requests. You use the following process to download the job output:

- Download a 128 MB chunk of output by specifying the appropriate byte range using the `Range` header.
- Along with the data, the response includes a SHA256 tree hash of the payload. You compute the checksum of the payload on the client and compare it with the checksum you received in the response to ensure you received all the expected data.
- Repeat steps 1 and 2 for all the eight 128 MB chunks of output data, each time specifying the appropriate byte range.
- After downloading all the parts of the job output, you have a list of eight checksum values. Compute the tree hash of these values to find the checksum of the entire output. Using the `DescribeJob` API, obtain job information of the job that provided you the output. The response includes the checksum of the entire archive stored in Amazon Glacier. You compare this value with the checksum you computed to ensure you have downloaded the entire archive content with no errors.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#).

For conceptual information and the underlying REST API, go to [Downloading a Vault Inventory](#) , [Downloading an Archive](#) , and [Get Job Output](#)

Request Syntax

```
response = client.get_job_output(
    vaultName='string',
    jobId='string',
    range='string'
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **vaultName** (*string*) – **[REQUIRED]**

The name of the vault.

- **jobId** (*string*) – **[REQUIRED]**

The job ID whose data is downloaded.

- **range** (*string*) – The range of bytes to retrieve from the output. For example, if you want to download the first 1,048,576 bytes, specify “Range: bytes=0-1048575”. By default, this operation downloads the entire output.

Return type dict

Returns

Response Syntax

```
{
    'body': b'bytes',
    'checksum': 'string',
    'status': 123,
    'contentRange': 'string',
    'acceptRanges': 'string',
    'contentType': 'string',
    'archiveDescription': 'string'
}
```

Response Structure

- (*dict*) –

Contains the Amazon Glacier response to your request.

- **body** (*bytes*) –

The job data, either archive data or inventory data.

- **checksum** (*string*) –

The checksum of the data in the response. This header is returned only when retrieving the output for an archive retrieval job. Furthermore, this header appears only under the following conditions:

- * You get the entire range of the archive.
- * You request a range to return of the archive that starts and ends on a multiple of 1 MB. For example, if you have an 3.1 MB archive and

you specify a range to return that starts at 1 MB and ends at 2 MB, then the `x-amz-sha256-tree-hash` is returned as a response header.

- * You request a range of the archive to return that starts on a multiple of 1 MB and goes to the end of the archive. For example, if you have a 3.1 MB archive and you specify a range that starts at 2 MB and ends at 3.1 MB (the end of the archive), then the `x-amz-sha256-tree-hash` is returned as a response header.

– **status** (*integer*) –

The HTTP response code for a job output request. The value depends on whether a range was specified in the request.

– **contentRange** (*string*) –

The range of bytes returned by Amazon Glacier. If only partial output is downloaded, the response provides the range of bytes Amazon Glacier returned. For example, bytes 0-1048575/8388608 returns the first 1 MB from 8 MB.

– **acceptRanges** (*string*) –

Indicates the range units accepted. For more information, go to [RFC2616](#).

– **contentType** (*string*) –

The Content-Type depends on whether the job output is an archive or a vault inventory. For archive data, the Content-Type is `application/octet-stream`. For vault inventory, if you requested CSV format when you initiated the job, the Content-Type is `text/csv`. Otherwise, by default, vault inventory is returned as JSON, and the Content-Type is `application/json`.

– **archiveDescription** (*string*) –

The description of an archive.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_vault_access_policy (***kwargs*)

This operation retrieves the `access-policy` subresource set on the vault; for more information on setting this subresource, see [Set Vault Access Policy \(PUT access-policy\)](#). If there is no access policy set on the vault, the operation returns a 404 Not found error. For more information about vault access policies, see [Amazon Glacier Access Control with Vault Access Policies](#).

Request Syntax

```
response = client.get_vault_access_policy(
    vaultName='string'
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos`“-“` apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to `“-”` by default if no value is not specified.

- **vaultName** (*string*) – [REQUIRED]

The name of the vault.

Return type dict

Returns

Response Syntax

```
{
  'policy': {
    'Policy': 'string'
  }
}
```

Response Structure

- (*dict*) –

Output for `GetVaultAccessPolicy`.

- **policy** (*dict*) –

Contains the returned vault access policy as a JSON string.

- * **Policy** (*string*) –

The vault access policy.

get_vault_lock (***kwargs*)

This operation retrieves the following attributes from the `lock-policy` subresource set on the specified vault:

- The vault lock policy set on the vault.
- The state of the vault lock, which is either `InProgress` or `Locked`.
- When the lock ID expires. The lock ID is used to complete the vault locking process.
- When the vault lock was initiated and put into the `InProgress` state.

A vault lock is put into the `InProgress` state by calling `InitiateVaultLock`. A vault lock is put into the `Locked` state by calling `CompleteVaultLock`. You can abort the vault locking process by calling `AbortVaultLock`. For more information about the vault locking process, [Amazon Glacier Vault Lock](#).

If there is no vault lock policy set on the vault, the operation returns a 404 `Not found` error. For more information about vault lock policies, [Amazon Glacier Access Control with Vault Lock Policies](#).

Request Syntax

```
response = client.get_vault_lock(
    vaultName='string'
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos`“-“` apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **vaultName** (*string*) – [REQUIRED]

The name of the vault.

Return type dict

Returns

Response Syntax

```
{
    'Policy': 'string',
    'State': 'string',
    'ExpirationDate': 'string',
    'CreationDate': 'string'
}
```

Response Structure

- (*dict*) –

Contains the Amazon Glacier response to your request.

- **Policy** (*string*) –

The vault lock policy as a JSON string, which uses “” as an escape character.

- **State** (*string*) –

The state of the vault lock. `InProgress` or `Locked`.

- **ExpirationDate** (*string*) –

The UTC date and time at which the lock ID expires. This value can be null if the vault lock is in a `Locked` state.

- **CreationDate** (*string*) –

The UTC date and time at which the vault lock was put into the `InProgress` state.

get_vault_notifications (**kwargs)

This operation retrieves the `notification-configuration` subresource of the specified vault.

For information about setting a notification configuration on a vault, see `SetVaultNotifications`. If a notification configuration for a vault is not set, the operation returns a `404 Not Found` error. For more information about vault notifications, see [Configuring Vault Notifications in Amazon Glacier](#).

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don’t have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#).

For conceptual information and underlying REST API, go to [Configuring Vault Notifications in Amazon Glacier](#) and [Get Vault Notification Configuration](#) in the *Amazon Glacier Developer Guide*.

Request Syntax

```
response = client.get_vault_notifications(
    vaultName='string'
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **vaultName** (*string*) – **[REQUIRED]**

The name of the vault.

Return type dict

Returns

Response Syntax

```
{
  'vaultNotificationConfig': {
    'SNSTopic': 'string',
    'Events': [
      'string',
    ]
  }
}
```

Response Structure

- (*dict*) –

Contains the Amazon Glacier response to your request.

- **vaultNotificationConfig** (*dict*) –

Returns the notification configuration set on the vault.

- * **SNSTopic** (*string*) –

The Amazon Simple Notification Service (Amazon SNS) topic Amazon Resource Name (ARN).

- * **Events** (*list*) –

A list of one or more events for which Amazon Glacier will send a notification to the specified Amazon SNS topic.

- (*string*) –

get_waiter (*waiter_name*)

initiate_job (***kwargs*)

This operation initiates a job of the specified type. In this release, you can initiate a job to retrieve either an archive or a vault inventory (a list of archives in a vault).

Retrieving data from Amazon Glacier is a two-step process:

- Initiate a retrieval job. .. note::A data retrieval policy can cause your initiate retrieval job request to fail with a `PolicyEnforcedException` exception. For more information about data retrieval policies, see [Amazon Glacier Data Retrieval Policies](#) . For more information about the `PolicyEnforcedException` exception, see [Error Responses](#) .
- After the job completes, download the bytes.

The retrieval request is executed asynchronously. When you initiate a retrieval job, Amazon Glacier creates a job and returns a job ID in the response. When Amazon Glacier completes the job, you can get the job output (archive or inventory data). For information about getting job output, see `GetJobOutput` operation.

The job must complete before you can get its output. To determine when a job is complete, you have the following options:

- Use Amazon SNS Notification** You can specify an Amazon Simple Notification Service (Amazon SNS) topic to which Amazon Glacier can post a notification after the job is completed. You can specify an SNS topic per job request. The notification is sent only after Amazon Glacier completes the job. In addition to specifying an SNS topic per job request, you can configure vault notifications for a vault so that job notifications are always sent. For more information, see [SetVaultNotifications](#).
- Get job details** You can make a `DescribeJob` request to obtain job status information while a job is in progress. However, it is more efficient to use an Amazon SNS notification to determine when a job is complete.

Note: The information you get via notification is same that you get by calling `DescribeJob`.

If for a specific event, you add both the notification configuration on the vault and also specify an SNS topic in your initiate job request, Amazon Glacier sends both notifications. For more information, see [SetVaultNotifications](#).

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#).

About the Vault Inventory

Amazon Glacier prepares an inventory for each vault periodically, every 24 hours. When you initiate a job for a vault inventory, Amazon Glacier returns the last inventory for the vault. The inventory data you get might be up to a day or two days old. Also, the initiate inventory job might take some time to complete before you can download the vault inventory. So you do not want to retrieve a vault inventory for each vault operation. However, in some scenarios, you might find the vault inventory useful. For example, when you upload an archive, you can provide an archive description but not an archive name. Amazon Glacier provides you a unique archive ID, an opaque string of characters. So, you might maintain your own database that maps archive names to their corresponding Amazon Glacier assigned archive IDs. You might find the vault inventory useful in the event you need to reconcile information in your database with the actual vault inventory.

Range Inventory Retrieval

You can limit the number of inventory items retrieved by filtering on the archive creation date or by setting a limit.

Filtering by Archive Creation Date

You can retrieve inventory items for archives created between `StartDate` and `EndDate` by specifying values for these parameters in the **InitiateJob** request. Archives created on or after the `StartDate` and before the `EndDate` will be returned. If you only provide the `StartDate` without the `EndDate`, you will retrieve the inventory for all archives created on or after the `StartDate`. If you only provide the `EndDate` without the `StartDate`, you will get back the inventory for all archives created before the `EndDate`.

Limiting Inventory Items per Retrieval

You can limit the number of inventory items returned by setting the `Limit` parameter in the **InitiateJob** request. The inventory job output will contain inventory items up to the specified `Limit`. If there are more inventory items available, the result is paginated. After a job is complete you can use the `DescribeJob` operation to get a marker that you use in a subsequent **InitiateJob** request. The marker will indicate the starting point to retrieve the next set of inventory items. You can page through your entire inventory by repeatedly making **InitiateJob** requests with the marker from the previous **DescribeJob** output, until you get a marker from **DescribeJob** that returns null, indicating that there are no more inventory items available.

You can use the `Limit` parameter together with the date range parameters.

About Ranged Archive Retrieval

You can initiate an archive retrieval for the whole archive or a range of the archive. In the case of ranged archive retrieval, you specify a byte range to return or the whole archive. The range specified must be megabyte (MB) aligned, that is the range start value must be divisible by 1 MB and range end value plus 1 must be divisible by 1 MB or equal the end of the archive. If the ranged archive retrieval is not megabyte aligned, this operation returns a 400 response. Furthermore, to ensure you get checksum values for data you download using Get Job Output API, the range must be tree hash aligned.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#).

For conceptual information and the underlying REST API, go to [Initiate a Job](#) and [Downloading a Vault Inventory](#)

Request Syntax

```
response = client.initiate_job(
    vaultName='string',
    jobParameters={
        'Format': 'string',
        'Type': 'string',
        'ArchiveId': 'string',
        'Description': 'string',
        'SNSTopic': 'string',
        'RetrievalByteRange': 'string',
        'InventoryRetrievalParameters': {
            'StartDate': 'string',
            'EndDate': 'string',
            'Limit': 'string',
            'Marker': 'string'
        }
    }
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **vaultName** (*string*) – [REQUIRED]

The name of the vault.

- **jobParameters** (*dict*) – Provides options for specifying job information.

- **Format** (*string*) –

When initiating a job to retrieve a vault inventory, you can optionally add this parameter to your request to specify the output format. If you are initiating an inventory job and do not specify a `Format` field, JSON is the default format. Valid values are “CSV” and “JSON”.

- **Type** (*string*) –

The job type. You can initiate a job to retrieve an archive or get an inventory of a vault. Valid values are “archive-retrieval” and “inventory-retrieval”.

– **ArchiveId** (*string*) –

The ID of the archive that you want to retrieve. This field is required only if `Type` is set to archive-retrieval. An error occurs if you specify this request parameter for an inventory retrieval job request.

– **Description** (*string*) –

The optional description for the job. The description must be less than or equal to 1,024 bytes. The allowable characters are 7-bit ASCII without control codes—specifically, ASCII values 32-126 decimal or 0x20-0x7E hexadecimal.

– **SNSTopic** (*string*) –

The Amazon SNS topic ARN to which Amazon Glacier sends a notification when the job is completed and the output is ready for you to download. The specified topic publishes the notification to its subscribers. The SNS topic must exist.

– **RetrievalByteRange** (*string*) –

The byte range to retrieve for an archive retrieval, in the form “*StartByteValue* -*EndByteValue*”. If not specified, the whole archive is retrieved. If specified, the byte range must be megabyte (1024*1024) aligned which means that *StartByteValue* must be divisible by 1 MB and *EndByteValue* plus 1 must be divisible by 1 MB or be the end of the archive specified as the archive byte size value minus 1. If `RetrievalByteRange` is not megabyte aligned, this operation returns a 400 response.

An error occurs if you specify this field for an inventory retrieval job request.

– **InventoryRetrievalParameters** (*dict*) –

Input parameters used for range inventory retrieval.

* **StartDate** (*string*) –

The start of the date range in UTC for vault inventory retrieval that includes archives created on or after this date. A string representation of ISO 8601 date format, for example, 2013-03-20T17:03:43Z.

* **EndDate** (*string*) –

The end of the date range in UTC for vault inventory retrieval that includes archives created before this date. A string representation of ISO 8601 date format, for example, 2013-03-20T17:03:43Z.

* **Limit** (*string*) –

Specifies the maximum number of inventory items returned per vault inventory retrieval request. Valid values are greater than or equal to 1.

* **Marker** (*string*) –

An opaque string that represents where to continue pagination of the vault inventory retrieval results. You use the marker in a new **InitiateJob** request to obtain additional inventory items. If there are no more inventory items, this value is `null`.

Return type dict
Returns

Response Syntax

```
{
    'location': 'string',
    'jobId': 'string'
}
```

Response Structure

- (*dict*) –

Contains the Amazon Glacier response to your request.

- **location** (*string*) –

The relative URI path of the job.

- **jobId** (*string*) –

The ID of the job.

`initiate_multipart_upload` (**kwargs)

This operation initiates a multipart upload. Amazon Glacier creates a multipart upload resource and returns its ID in the response. The multipart upload ID is used in subsequent requests to upload parts of an archive (see `UploadMultipartPart`).

When you initiate a multipart upload, you specify the part size in number of bytes. The part size must be a megabyte (1024 KB) multiplied by a power of 2—for example, 1048576 (1 MB), 2097152 (2 MB), 4194304 (4 MB), 8388608 (8 MB), and so on. The minimum allowable part size is 1 MB, and the maximum is 4 GB.

Every part you upload to this resource (see `UploadMultipartPart`), except the last one, must have the same size. The last one can be the same size or smaller. For example, suppose you want to upload a 16.2 MB file. If you initiate the multipart upload with a part size of 4 MB, you will upload four parts of 4 MB each and one part of 0.2 MB.

Note: You don't need to know the size of the archive when you start a multipart upload because Amazon Glacier does not require you to specify the overall archive size.

After you complete the multipart upload, Amazon Glacier removes the multipart upload resource referenced by the ID. Amazon Glacier also removes the multipart upload resource if you cancel the multipart upload or it may be removed if there is no activity for a period of 24 hours.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#).

For conceptual information and underlying REST API, go to [Uploading Large Archives in Parts \(Multipart Upload\)](#) and [Initiate Multipart Upload](#) in the *Amazon Glacier Developer Guide*.

Request Syntax

```
response = client.initiate_multipart_upload(
    vaultName='string',
    archiveDescription='string',
    partSize='string'
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos`'-'` apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to `-` by default if no value is not specified.

- **vaultName** (*string*) – [REQUIRED]

The name of the vault.

- **archiveDescription** (*string*) – The archive description that you are uploading in parts.

The part size must be a megabyte (1024 KB) multiplied by a power of 2, for example 1048576 (1 MB), 2097152 (2 MB), 4194304 (4 MB), 8388608 (8 MB), and so on. The minimum allowable part size is 1 MB, and the maximum is 4 GB (4096 MB).

- **partSize** (*string*) – The size of each part except the last, in bytes. The last part can be smaller than this part size.

Return type dict

Returns

Response Syntax

```
{
    'location': 'string',
    'uploadId': 'string'
}
```

Response Structure

- (*dict*) –

The Amazon Glacier response to your request.

- **location** (*string*) –

The relative URI path of the multipart upload ID Amazon Glacier created.

- **uploadId** (*string*) –

The ID of the multipart upload. This value is also included as part of the location.

initiate_vault_lock (***kwargs*)

This operation initiates the vault locking process by doing the following:

- Installing a vault lock policy on the specified vault.
- Setting the lock state of vault lock to `InProgress`.
- Returning a lock ID, which is used to complete the vault locking process.

You can set one vault lock policy for each vault and this policy can be up to 20 KB in size. For more information about vault lock policies, see [Amazon Glacier Access Control with Vault Lock Policies](#).

You must complete the vault locking process within 24 hours after the vault lock enters the `InProgress` state. After the 24 hour window ends, the lock ID expires, the vault automatically exits the `InProgress` state, and the vault lock policy is removed from the vault. You call `CompleteVaultLock` to complete the vault locking process by setting the state of the vault lock to `Locked`.

After a vault lock is in the `Locked` state, you cannot initiate a new vault lock for the vault.

You can abort the vault locking process by calling `AbortVaultLock`. You can get the state of the vault lock by calling `GetVaultLock`. For more information about the vault locking process, see [Amazon Glacier Vault Lock](#).

If this operation is called when the vault lock is in the `InProgress` state, the operation returns an `AccessDeniedException` error. When the vault lock is in the `InProgress` state you must call `AbortVaultLock` before you can initiate a new vault lock policy.

Request Syntax

```
response = client.initiate_vault_lock(
    vaultName='string',
    policy={
        'Policy': 'string'
    }
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID. This value must match the AWS account ID associated with the credentials used to sign the request. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you specify your account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **vaultName** (*string*) – **[REQUIRED]**

The name of the vault.

- **policy** (*dict*) – The vault lock policy as a JSON string, which uses “” as an escape character.

– **Policy** (*string*) –

The vault lock policy.

Return type dict

Returns

Response Syntax

```
{
    'lockId': 'string'
}
```

Response Structure

- (*dict*) –

Contains the Amazon Glacier response to your request.

– **lockId** (*string*) –

The lock ID, which is used to complete the vault locking process.

list_jobs (***kwargs*)

This operation lists jobs for a vault, including jobs that are in-progress and jobs that have recently finished.

Note: Amazon Glacier retains recently completed jobs for a period before deleting them; however, it eventually removes completed jobs. The output of completed jobs can be retrieved. Retaining completed jobs for a period of time after they have completed enables you to get a job output in the event you miss the job completion notification or your first attempt to download it fails. For example, suppose you start an archive retrieval job to download an archive. After the job completes, you start to download the archive but encounter a network error. In this scenario, you can retry and download the archive while the job exists.

To retrieve an archive or retrieve a vault inventory from Amazon Glacier, you first initiate a job, and after the job completes, you download the data. For an archive retrieval, the output is the archive data, and for an inventory retrieval, it is the inventory list. The List Job operation returns a list of these jobs sorted by job initiation time.

This List Jobs operation supports pagination. By default, this operation returns up to 1,000 jobs in the response. You should always check the response for a `marker` at which to continue the list; if there are no more items the `marker` is `null`. To return a list of jobs that begins at a specific job, set the `marker` request parameter to the value you obtained from a previous List Jobs request. You can also limit the number of jobs returned in the response by specifying the `limit` parameter in the request.

Additionally, you can filter the jobs list returned by specifying an optional `statuscode` (InProgress, Succeeded, or Failed) and `completed` (true, false) parameter. The `statuscode` allows you to specify that only jobs that match a specified status are returned. The `completed` parameter allows you to specify that only jobs in a specific completion state are returned.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#).

For the underlying REST API, go to [List Jobs](#)

Request Syntax

```
response = client.list_jobs(
    vaultName='string',
    limit='string',
    marker='string',
    statuscode='string',
    completed='string'
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **vaultName** (*string*) – **[REQUIRED]**

The name of the vault.

- **limit** (*string*) – Specifies that the response be limited to the specified number of items or fewer. If not specified, the List Jobs operation returns up to 1,000 jobs.
- **marker** (*string*) – An opaque string used for pagination. This value specifies the job at which the listing of jobs should begin. Get the marker value from a previous List Jobs response. You need only include the marker if you are continuing the pagination of results started in a previous List Jobs request.
- **statuscode** (*string*) – Specifies the type of job status to return. You can specify the following values: “InProgress”, “Succeeded”, or “Failed”.
- **completed** (*string*) – Specifies the state of the jobs to return. You can specify `true` or `false`.

Return type dict

Returns

Response Syntax

```

{
  'JobList': [
    {
      'JobId': 'string',
      'JobDescription': 'string',
      'Action': 'ArchiveRetrieval'|'InventoryRetrieval',
      'ArchiveId': 'string',
      'VaultARN': 'string',
      'CreationDate': 'string',
      'Completed': True|False,
      'StatusCode': 'InProgress'|'Succeeded'|'Failed',
      'StatusMessage': 'string',
      'ArchiveSizeInBytes': 123,
      'InventorySizeInBytes': 123,
      'SNSTopic': 'string',
      'CompletionDate': 'string',
      'SHA256TreeHash': 'string',
      'ArchiveSHA256TreeHash': 'string',
      'RetrievalByteRange': 'string',
      'InventoryRetrievalParameters': {
        'Format': 'string',
        'StartDate': 'string',
        'EndDate': 'string',
        'Limit': 'string',
        'Marker': 'string'
      }
    },
  ],
  'Marker': 'string'
}

```

Response Structure

- (*dict*) –

Contains the Amazon Glacier response to your request.

- **JobList** (*list*) –

A list of job objects. Each job object contains metadata describing the job.

- * (*dict*) –

Describes an Amazon Glacier job.

- **JobId** (*string*) –

An opaque string that identifies an Amazon Glacier job.

- **JobDescription** (*string*) –

The job description you provided when you initiated the job.

- **Action** (*string*) –

The job type. It is either ArchiveRetrieval or InventoryRetrieval.

- **ArchiveId** (*string*) –

For an ArchiveRetrieval job, this is the archive ID requested for download. Otherwise, this field is null.

- **VaultARN** (*string*) –

The Amazon Resource Name (ARN) of the vault from which the archive retrieval was requested.

- **CreationDate** (*string*) –

The UTC date when the job was created. A string representation of ISO 8601 date format, for example, “2012-03-20T17:03:43.221Z”.

- **Completed** (*boolean*) –

The job status. When a job is completed, you get the job’s output.

- **StatusCode** (*string*) –

The status code can be InProgress, Succeeded, or Failed, and indicates the status of the job.

- **StatusMessage** (*string*) –

A friendly message that describes the job status.

- **ArchiveSizeInBytes** (*integer*) –

For an ArchiveRetrieval job, this is the size in bytes of the archive being requested for download. For the InventoryRetrieval job, the value is null.

- **InventorySizeInBytes** (*integer*) –

For an InventoryRetrieval job, this is the size in bytes of the inventory requested for download. For the ArchiveRetrieval job, the value is null.

- **SNSTopic** (*string*) –

An Amazon Simple Notification Service (Amazon SNS) topic that receives notification.

- **CompletionDate** (*string*) –

The UTC time that the archive retrieval request completed. While the job is in progress, the value will be null.

- **SHA256TreeHash** (*string*) –

For an ArchiveRetrieval job, it is the checksum of the archive. Otherwise, the value is null.

The SHA256 tree hash value for the requested range of an archive. If the Initiate a Job request for an archive specified a tree-hash aligned range, then this field returns a value.

For the specific case when the whole archive is retrieved, this value is the same as the ArchiveSHA256TreeHash value.

This field is null in the following situations:

- Archive retrieval jobs that specify a range that is not tree-hash aligned.
- Archival jobs that specify a range that is equal to the whole archive and the job status is InProgress.
- Inventory jobs.

- **ArchiveSHA256TreeHash** (*string*) –

The SHA256 tree hash of the entire archive for an archive retrieval. For inventory retrieval jobs, this field is null.

- **RetrievalByteRange** (*string*) –

The retrieved byte range for archive retrieval jobs in the form “StartByteValue -EndByteValue ” If no range was specified in the archive retrieval, then the whole archive is retrieved and

StartByteValue equals 0 and *EndByteValue* equals the size of the archive minus 1. For inventory retrieval jobs this field is null.

- **InventoryRetrievalParameters** (*dict*) –

Parameters used for range inventory retrieval.

- **Format** (*string*) –

The output format for the vault inventory list, which is set by the **InitiateJob** request when initiating a job to retrieve a vault inventory. Valid values are “CSV” and “JSON”.

- **StartDate** (*string*) –

The start of the date range in UTC for vault inventory retrieval that includes archives created on or after this date. A string representation of ISO 8601 date format, for example, 2013-03-20T17:03:43Z.

- **EndDate** (*string*) –

The end of the date range in UTC for vault inventory retrieval that includes archives created before this date. A string representation of ISO 8601 date format, for example, 2013-03-20T17:03:43Z.

- **Limit** (*string*) –

Specifies the maximum number of inventory items returned per vault inventory retrieval request. This limit is set when initiating the job with the a **InitiateJob** request.

- **Marker** (*string*) –

An opaque string that represents where to continue pagination of the vault inventory retrieval results. You use the marker in a new **InitiateJob** request to obtain additional inventory items. If there are no more inventory items, this value is `null`. For more information, see [Range Inventory Retrieval](#).

- **Marker** (*string*) –

An opaque string that represents where to continue pagination of the results. You use this value in a new List Jobs request to obtain more jobs in the list. If there are no more jobs, this value is `null`.

list_multipart_uploads (***kwargs*)

This operation lists in-progress multipart uploads for the specified vault. An in-progress multipart upload is a multipart upload that has been initiated by an `InitiateMultipartUpload` request, but has not yet been completed or aborted. The list returned in the List Multipart Upload response has no guaranteed order.

The List Multipart Uploads operation supports pagination. By default, this operation returns up to 1,000 multipart uploads in the response. You should always check the response for a `marker` at which to continue the list; if there are no more items the `marker` is `null`. To return a list of multipart uploads that begins at a specific upload, set the `marker` request parameter to the value you obtained from a previous List Multipart Upload request. You can also limit the number of uploads returned in the response by specifying the `limit` parameter in the request.

Note the difference between this operation and listing parts (`ListParts`). The List Multipart Uploads operation lists all multipart uploads for a vault and does not require a multipart upload ID. The List Parts operation requires a multipart upload ID since parts are associated with a single upload.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don’t have any permissions by default. You must grant them explicit

permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#) .

For conceptual information and the underlying REST API, go to [Working with Archives in Amazon Glacier](#) and [List Multipart Uploads](#) in the *Amazon Glacier Developer Guide* .

Request Syntax

```
response = client.list_multipart_uploads(
    vaultName='string',
    marker='string',
    limit='string'
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **vaultName** (*string*) – **[REQUIRED]**

The name of the vault.

- **marker** (*string*) – An opaque string used for pagination. This value specifies the upload at which the listing of uploads should begin. Get the marker value from a previous List Uploads response. You need only include the marker if you are continuing the pagination of results started in a previous List Uploads request.
- **limit** (*string*) – Specifies the maximum number of uploads returned in the response body. If this value is not specified, the List Uploads operation returns up to 1,000 uploads.

Return type dict

Returns

Response Syntax

```
{
    'UploadsList': [
        {
            'MultipartUploadId': 'string',
            'VaultARN': 'string',
            'ArchiveDescription': 'string',
            'PartSizeInBytes': 123,
            'CreationDate': 'string'
        },
    ],
    'Marker': 'string'
}
```

Response Structure

- (*dict*) –
 - Contains the Amazon Glacier response to your request.
 - **UploadsList** (*list*) –
 - A list of in-progress multipart uploads.

- * (*dict*) –
 - A list of in-progress multipart uploads for a vault.
 - **MultipartUploadId** (*string*) –
 - The ID of a multipart upload.
 - **VaultARN** (*string*) –
 - The Amazon Resource Name (ARN) of the vault that contains the archive.
 - **ArchiveDescription** (*string*) –
 - The description of the archive that was specified in the Initiate Multipart Upload request.
 - **PartSizeInBytes** (*integer*) –
 - The part size, in bytes, specified in the Initiate Multipart Upload request. This is the size of all the parts in the upload except the last part, which may be smaller than this size.
 - **CreationDate** (*string*) –
 - The UTC time at which the multipart upload was initiated.
 - **Marker** (*string*) –
 - An opaque string that represents where to continue pagination of the results. You use the marker in a new List Multipart Uploads request to obtain more uploads in the list. If there are no more uploads, this value is `null`.

list_parts (***kwargs*)

This operation lists the parts of an archive that have been uploaded in a specific multipart upload. You can make this request at any time during an in-progress multipart upload before you complete the upload (see `CompleteMultipartUpload`). List Parts returns an error for completed uploads. The list returned in the List Parts response is sorted by part range.

The List Parts operation supports pagination. By default, this operation returns up to 1,000 uploaded parts in the response. You should always check the response for a `marker` at which to continue the list; if there are no more items the `marker` is `null`. To return a list of parts that begins at a specific part, set the `marker` request parameter to the value you obtained from a previous List Parts request. You can also limit the number of parts returned in the response by specifying the `limit` parameter in the request.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#).

For conceptual information and the underlying REST API, go to [Working with Archives in Amazon Glacier](#) and [List Parts in the Amazon Glacier Developer Guide](#).

Request Syntax

```
response = client.list_parts(
    vaultName='string',
    uploadId='string',
    marker='string',
    limit='string'
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **vaultName** (*string*) – [REQUIRED]

The name of the vault.

- **uploadId** (*string*) – [REQUIRED]

The upload ID of the multipart upload.

- **marker** (*string*) – An opaque string used for pagination. This value specifies the part at which the listing of parts should begin. Get the marker value from the response of a previous List Parts response. You need only include the marker if you are continuing the pagination of results started in a previous List Parts request.
- **limit** (*string*) – Specifies the maximum number of parts returned in the response body. If this value is not specified, the List Parts operation returns up to 1,000 uploads.

Return type dict

Returns

Response Syntax

```
{
    'MultipartUploadId': 'string',
    'VaultARN': 'string',
    'ArchiveDescription': 'string',
    'PartSizeInBytes': 123,
    'CreationDate': 'string',
    'Parts': [
        {
            'RangeInBytes': 'string',
            'SHA256TreeHash': 'string'
        },
    ],
    'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Contains the Amazon Glacier response to your request.

- **MultipartUploadId** (*string*) –

The ID of the upload to which the parts are associated.

- **VaultARN** (*string*) –

The Amazon Resource Name (ARN) of the vault to which the multipart upload was initiated.

- **ArchiveDescription** (*string*) –

The description of the archive that was specified in the Initiate Multipart Upload request.

- **PartSizeInBytes** (*integer*) –

The part size in bytes.

- **CreationDate** (*string*) –

The UTC time at which the multipart upload was initiated.

- **Parts** (*list*) –

A list of the part sizes of the multipart upload.

- * (*dict*) –

A list of the part sizes of the multipart upload.

- **RangeInBytes** (*string*) –

The byte range of a part, inclusive of the upper value of the range.

- **SHA256TreeHash** (*string*) –

The SHA256 tree hash value that Amazon Glacier calculated for the part. This field is never `null`.

- **Marker** (*string*) –

An opaque string that represents where to continue pagination of the results. You use the marker in a new List Parts request to obtain more jobs in the list. If there are no more parts, this value is `null`.

`list_tags_for_vault` (**kwargs)

This operation lists all the tags attached to a vault. The operation returns an empty map if there are no tags. For more information about tags, see [Tagging Amazon Glacier Resources](#).

Request Syntax

```
response = client.list_tags_for_vault(
    vaultName='string'
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **vaultName** (*string*) – **[REQUIRED]**

The name of the vault.

Return type dict

Returns

Response Syntax

```
{
  'Tags': {
    'string': 'string'
  }
}
```

Response Structure

- (*dict*) –

Contains the Amazon Glacier response to your request.

– **Tags** (*dict*) –

The tags attached to the vault. Each tag is composed of a key and a value.

- * (*string*) –
- (*string*) –

list_vaults (***kwargs*)

This operation lists all vaults owned by the calling user’s account. The list returned in the response is ASCII-sorted by vault name.

By default, this operation returns up to 1,000 items. If there are more vaults to list, the response `marker` field contains the vault Amazon Resource Name (ARN) at which to continue the list with a new List Vaults request; otherwise, the `marker` field is `null`. To return a list of vaults that begins at a specific vault, set the `marker` request parameter to the vault ARN you obtained from a previous List Vaults request. You can also limit the number of vaults returned in the response by specifying the `limit` parameter in the request.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don’t have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#).

For conceptual information and underlying REST API, go to [Retrieving Vault Metadata in Amazon Glacier](#) and [List Vaults](#) in the *Amazon Glacier Developer Guide*.

Request Syntax

```
response = client.list_vaults(
    marker='string',
    limit='string'
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID. This value must match the AWS account ID associated with the credentials used to sign the request. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you specify your account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **marker** (*string*) – A string used for pagination. The marker specifies the vault ARN after which the listing of vaults should begin.
- **limit** (*string*) – The maximum number of items returned in the response. If you don’t specify a value, the List Vaults operation returns up to 1,000 items.

Return type dict

Returns

Response Syntax

```
{
  'VaultList': [
    {
      'VaultARN': 'string',
      'VaultName': 'string',
      'CreationDate': 'string',
      'LastInventoryDate': 'string',
      'NumberOfArchives': 123,
```

```

        'SizeInBytes': 123
    },
],
'Marker': 'string'
}

```

Response Structure

- *(dict)* –

Contains the Amazon Glacier response to your request.

- **VaultList** (*list*) –

List of vaults.

- * *(dict)* –

Contains the Amazon Glacier response to your request.

- **VaultARN** (*string*) –

The Amazon Resource Name (ARN) of the vault.

- **VaultName** (*string*) –

The name of the vault.

- **CreationDate** (*string*) –

The UTC date when the vault was created. A string representation of ISO 8601 date format, for example, “2012-03-20T17:03:43.221Z”.

- **LastInventoryDate** (*string*) –

The UTC date when Amazon Glacier completed the last vault inventory. A string representation of ISO 8601 date format, for example, “2012-03-20T17:03:43.221Z”.

- **NumberOfArchives** (*integer*) –

The number of archives in the vault as of the last inventory date. This field will return `null` if an inventory has not yet run on the vault, for example, if you just created the vault.

- **SizeInBytes** (*integer*) –

Total size, in bytes, of the archives in the vault as of the last inventory date. This field will return `null` if an inventory has not yet run on the vault, for example, if you just created the vault.

- **Marker** (*string*) –

The vault ARN at which to continue pagination of the results. You use the marker in another List Vaults request to obtain more vaults in the list.

remove_tags_from_vault (***kwargs*)

This operation removes one or more tags from the set of tags attached to a vault. For more information about tags, see [Tagging Amazon Glacier Resources](#). This operation is idempotent. The operation will be successful, even if there are no tags attached to the vault.

Request Syntax

```

response = client.remove_tags_from_vault(
    vaultName='string',
    TagKeys=[
        'string',

```

```
]
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **vaultName** (*string*) – [REQUIRED]

The name of the vault.

- **TagKeys** (*list*) – A list of tag keys. Each corresponding tag is removed from the vault.

– (*string*) –

Returns

None

set_data_retrieval_policy (***kwargs*)

This operation sets and then enacts a data retrieval policy in the region specified in the PUT request. You can set one policy per region for an AWS account. The policy is enacted within a few minutes of a successful PUT operation.

The set policy operation does not affect retrieval jobs that were in progress before the policy was enacted. For more information about data retrieval policies, see [Amazon Glacier Data Retrieval Policies](#).

Request Syntax

```
response = client.set_data_retrieval_policy(
    Policy={
        'Rules': [
            {
                'Strategy': 'string',
                'BytesPerHour': 123
            },
        ]
    }
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID. This value must match the AWS account ID associated with the credentials used to sign the request. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you specify your account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **Policy** (*dict*) – The data retrieval policy in JSON format.

– **Rules** (*list*) –

The policy rule. Although this is a list type, currently there must be only one rule, which contains a `Strategy` field and optionally a `BytesPerHour` field.

* (*dict*) –

Data retrieval policy rule.

- **Strategy** (*string*) –
The type of data retrieval policy to set.
Valid values: BytesPerHour|FreeTier|None
- **BytesPerHour** (*integer*) –
The maximum number of bytes that can be retrieved in an hour.
This field is required only if the value of the Strategy field is BytesPerHour . Your PUT operation will be rejected if the Strategy field is not set to BytesPerHour and you set this field.

Returns None

set_vault_access_policy (***kwargs*)

This operation configures an access policy for a vault and will overwrite an existing policy. To configure a vault access policy, send a PUT request to the `access-policy` subresource of the vault. An access policy is specific to a vault and is also called a vault subresource. You can set one access policy per vault and the policy can be up to 20 KB in size. For more information about vault access policies, see [Amazon Glacier Access Control with Vault Access Policies](#) .

Request Syntax

```
response = client.set_vault_access_policy(
    vaultName='string',
    policy={
        'Policy': 'string'
    }
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.
Note: this parameter is set to “-” by default if no value is not specified.
- **vaultName** (*string*) – **[REQUIRED]**
The name of the vault.
- **policy** (*dict*) – The vault access policy as a JSON string.
 - **Policy** (*string*) –
The vault access policy.

Returns None

set_vault_notifications (***kwargs*)

This operation configures notifications that will be sent when specific events happen to a vault. By default, you don't get any notifications.

To configure vault notifications, send a PUT request to the `notification-configuration` subresource of the vault. The request should include a JSON document that provides an Amazon SNS topic and specific events for which you want Amazon Glacier to send notifications to the topic.

Amazon SNS topics must grant permission to the vault to be allowed to publish notifications to the topic. You can configure a vault to publish a notification for the following vault events:

- **ArchiveRetrievalCompleted** This event occurs when a job that was initiated for an archive retrieval is completed (`InitiateJob`). The status of the completed job can be “Succeeded” or “Failed”. The notification sent to the SNS topic is the same output as returned from `DescribeJob` .

- **InventoryRetrievalCompleted** This event occurs when a job that was initiated for an inventory retrieval is completed (`InitiateJob`). The status of the completed job can be “Succeeded” or “Failed”. The notification sent to the SNS topic is the same output as returned from `DescribeJob` .

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don’t have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#) .

For conceptual information and underlying REST API, go to [Configuring Vault Notifications in Amazon Glacier](#) and [Set Vault Notification Configuration in the Amazon Glacier Developer Guide](#) .

Request Syntax

```
response = client.set_vault_notifications(
    vaultName='string',
    vaultNotificationConfig={
        'SNSTopic': 'string',
        'Events': [
            'string',
        ]
    }
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **vaultName** (*string*) – **[REQUIRED]**

The name of the vault.

- **vaultNotificationConfig** (*dict*) – Provides options for specifying notification configuration.

- **SNSTopic** (*string*) –

The Amazon Simple Notification Service (Amazon SNS) topic Amazon Resource Name (ARN).

- **Events** (*list*) –

A list of one or more events for which Amazon Glacier will send a notification to the specified Amazon SNS topic.

- * (*string*) –

Returns None

`upload_archive` (***kwargs*)

This operation adds an archive to a vault. This is a synchronous operation, and for a successful upload, your data is durably persisted. Amazon Glacier returns the archive ID in the `x-amz-archive-id` header of the response.

You must use the archive ID to access your data in Amazon Glacier. After you upload an archive, you should save the archive ID returned so that you can retrieve or delete the archive later. Besides saving the archive ID, you can also index it and give it a friendly name to allow for better searching. You can also

use the optional archive description field to specify how the archive is referred to in an external index of archives, such as you might create in Amazon DynamoDB. You can also get the vault inventory to obtain a list of archive IDs in a vault. For more information, see [InitiateJob](#) .

You must provide a SHA256 tree hash of the data you are uploading. For information about computing a SHA256 tree hash, see [Computing Checksums](#) .

You can optionally specify an archive description of up to 1,024 printable ASCII characters. You can get the archive description when you either retrieve the archive or get the vault inventory. For more information, see [InitiateJob](#) . Amazon Glacier does not interpret the description in any way. An archive description does not need to be unique. You cannot use the description to retrieve or sort the archive list.

Archives are immutable. After you upload an archive, you cannot edit the archive or its description.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#) .

For conceptual information and underlying REST API, go to [Uploading an Archive in Amazon Glacier](#) and [Upload Archive](#) in the *Amazon Glacier Developer Guide* .

Request Syntax

```
response = client.upload_archive(
    vaultName='string',
    archiveDescription='string',
    body=b'bytes'
)
```

Parameters

- **vaultName** (*string*) – [REQUIRED]
The name of the vault.
- **accountId** (*string*) – The `AccountId` value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.
Note: this parameter is set to “-” by default if no value is not specified.
- **archiveDescription** (*string*) – The optional description of the archive you are uploading.
- **checksum** (*string*) – The SHA256 tree hash of the data being uploaded.
Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **body** (*bytes*) – The data to upload.

Return type dict

Returns

Response Syntax

```
{
    'location': 'string',
    'checksum': 'string',
    'archiveId': 'string'
}
```

Response Structure

- (*dict*) –

Contains the Amazon Glacier response to your request.

For information about the underlying REST API, go to [Upload Archive](#) . For conceptual information, go to [Working with Archives in Amazon Glacier](#) .

- **location** (*string*) –

The relative URI path of the newly added archive resource.

- **checksum** (*string*) –

The checksum of the archive computed by Amazon Glacier.

- **archiveId** (*string*) –

The ID of the archive. This value is also included as part of the location.

upload_multipart_part (***kwargs*)

This operation uploads a part of an archive. You can upload archive parts in any order. You can also upload them in parallel. You can upload up to 10,000 parts for a multipart upload.

Amazon Glacier rejects your upload part request if any of the following conditions is true:

- SHA256 tree hash does not match** To ensure that part data is not corrupted in transmission, you compute a SHA256 tree hash of the part and include it in your request. Upon receiving the part data, Amazon Glacier also computes a SHA256 tree hash. If these hash values don't match, the operation fails. For information about computing a SHA256 tree hash, see [Computing Checksums](#) .

- Part size does not match** The size of each part except the last must match the size specified in the corresponding InitiateMultipartUpload request. The size of the last part must be the same size as, or smaller than, the specified size. .. note::If you upload a part whose size is smaller than the part size you specified in your initiate multipart upload request and that part is not the last part, then the upload part request will succeed. However, the subsequent Complete Multipart Upload request will fail.

- Range does not align** The byte range value in the request does not align with the part size specified in the corresponding initiate request. For example, if you specify a part size of 4194304 bytes (4 MB), then 0 to 4194303 bytes (4 MB - 1) and 4194304 (4 MB) to 8388607 (8 MB - 1) are valid part ranges. However, if you set a range value of 2 MB to 6 MB, the range does not align with the part size and the upload will fail.

This operation is idempotent. If you upload the same part multiple times, the data included in the most recent request overwrites the previously uploaded data.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#) .

For conceptual information and underlying REST API, go to [Uploading Large Archives in Parts \(Multipart Upload\)](#) and [Upload Part](#) in the *Amazon Glacier Developer Guide* .

Request Syntax

```
response = client.upload_multipart_part(  
    vaultName='string',  
    uploadId='string',  
    range='string',  
    body=b'bytes'  
)
```

Parameters

- **accountId** (*string*) – The `AccountId` value is the AWS account ID of the account that owns the vault. You can either specify an AWS account ID or optionally a single apos“-“ apos (hyphen), in which case Amazon Glacier uses the AWS account ID associated with the credentials used to sign the request. If you use an account ID, do not include any hyphens (apos-apos) in the ID.

Note: this parameter is set to “-” by default if no value is not specified.

- **vaultName** (*string*) – [REQUIRED]

The name of the vault.

- **uploadId** (*string*) – [REQUIRED]

The upload ID of the multipart upload.

- **checksum** (*string*) – The SHA256 tree hash of the data being uploaded.

Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required

- **range** (*string*) – Identifies the range of bytes in the assembled archive that will be uploaded in this part. Amazon Glacier uses this information to assemble the archive in the proper sequence. The format of this header follows RFC 2616. An example header is `Content-Range:bytes 0-4194303/*`.

- **body** (*bytes*) – The data to upload.

Return type dict

Returns

Response Syntax

```
{
    'checksum': 'string'
}
```

Response Structure

- (*dict*) –

Contains the Amazon Glacier response to your request.

- **checksum** (*string*) –

The SHA256 tree hash that Amazon Glacier computed for the uploaded part.

Service Resource

class `Glacier.ServiceResource`

A resource representing Amazon Glacier:

```
import boto3
```

```
glacier = boto3.resource('glacier')
```

These are the resource’s available actions:

- `create_vault()`

These are the resource’s available sub-resources:

- `Account()`
- `Archive()`
- `Job()`
- `MultipartUpload()`
- `Notification()`

- `Vault()`

These are the resource's available collections:

- `vaults`

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

create_vault (***kwargs*)

This operation creates a new vault with the specified name. The name of the vault must be unique within a region for an AWS account. You can create up to 1,000 vaults per account. If you need to create more vaults, contact Amazon Glacier.

You must use the following guidelines when naming a vault.

- Names can be between 1 and 255 characters long.
- Allowed characters are a-z, A-Z, 0-9, '_' (underscore), '-' (hyphen), and '.' (period).

This operation is idempotent.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#).

For conceptual information and underlying REST API, go to [Creating a Vault in Amazon Glacier](#) and [Create Vault in the Amazon Glacier Developer Guide](#).

Request Syntax

```
vault = glacier.create_vault(
    vaultName='string'
)
```

Parameters `vaultName` (*string*) – [REQUIRED]

The name of the vault.

Return type `glacier.Vault`

Returns Vault resource

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Account (*id*)

Creates a Account resource.:

```
account = glacier.Account('id')
```

Parameters `id` (*string*) – The Account's id identifier. This **must** be set.

Return type `Glacier.Account`

Returns A Account resource

Archive (*account_id, vault_name, id*)

Creates a Archive resource.:

```
archive = glacier.Archive('account_id', 'vault_name', 'id')
```

Parameters

- **account_id** (*string*) – The Archive's account_id identifier. This **must** be set.
- **vault_name** (*string*) – The Archive's vault_name identifier. This **must** be set.

- **id** (*string*) – The Archive’s id identifier. This **must** be set.

Return type `Glacier.Archive`

Returns A Archive resource

Job (*account_id*, *vault_name*, *id*)

Creates a Job resource.:

```
job = glacier.Job('account_id', 'vault_name', 'id')
```

Parameters

- **account_id** (*string*) – The Job’s account_id identifier. This **must** be set.
- **vault_name** (*string*) – The Job’s vault_name identifier. This **must** be set.
- **id** (*string*) – The Job’s id identifier. This **must** be set.

Return type `Glacier.Job`

Returns A Job resource

MultipartUpload (*account_id*, *vault_name*, *id*)

Creates a MultipartUpload resource.:

```
multipart_upload = glacier.MultipartUpload('account_id', 'vault_name', 'id')
```

Parameters

- **account_id** (*string*) – The MultipartUpload’s account_id identifier. This **must** be set.
- **vault_name** (*string*) – The MultipartUpload’s vault_name identifier. This **must** be set.
- **id** (*string*) – The MultipartUpload’s id identifier. This **must** be set.

Return type `Glacier.MultipartUpload`

Returns A MultipartUpload resource

Notification (*account_id*, *vault_name*)

Creates a Notification resource.:

```
notification = glacier.Notification('account_id', 'vault_name')
```

Parameters

- **account_id** (*string*) – The Notification’s account_id identifier. This **must** be set.
- **vault_name** (*string*) – The Notification’s vault_name identifier. This **must** be set.

Return type `Glacier.Notification`

Returns A Notification resource

Vault (*account_id*, *name*)

Creates a Vault resource.:

```
vault = glacier.Vault('account_id', 'name')
```

Parameters

- **account_id** (*string*) – The Vault’s account_id identifier. This **must** be set.
- **name** (*string*) – The Vault’s name identifier. This **must** be set.

Return type `Glacier.Vault`

Returns A Vault resource

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

vaults**all()**

Creates an iterable of all Vault resources in the collection.

Request Syntax

```
vault_iterator = glacier.vaults.all()
```

Return type list(glacier.Vault)

Returns A list of Vault resources

filter(kwargs)**

Creates an iterable of all Vault resources in the collection filtered by kwargs passed to method.

Request Syntax

```
vault_iterator = glacier.vaults.filter(
    marker='string',
    limit='string'
)
```

Parameters

- **marker** (*string*) – A string used for pagination. The marker specifies the vault ARN after which the listing of vaults should begin.
- **limit** (*string*) – The maximum number of items returned in the response. If you don't specify a value, the List Vaults operation returns up to 1,000 items.

Return type list(glacier.Vault)

Returns A list of Vault resources

limit(kwargs)**

Creates an iterable up to a specified amount of Vault resources in the collection.

Request Syntax

```
vault_iterator = glacier.vaults.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type list(glacier.Vault)

Returns A list of Vault resources

page_size(kwargs)**

Creates an iterable of all Vault resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
vault_iterator = glacier.vaults.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type list(glacier.Vault)

Returns A list of Vault resources

Account

class `Glacier.Account(id)`

A resource representing an Amazon Glacier Account:

```
import boto3

glacier = boto3.resource('glacier')
account = glacier.Account('id')
```

Parameters `id` (*string*) – The Account’s id identifier. This **must** be set.

These are the resource’s available identifiers:

- `id`

These are the resource’s available actions:

- `create_vault()`

These are the resource’s available sub-resources:

- `Vault()`

These are the resource’s available collections:

- `vaults`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

id

(*string*) The Account’s id identifier. This **must** be set.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

create_vault (***kwargs*)

This operation creates a new vault with the specified name. The name of the vault must be unique within a region for an AWS account. You can create up to 1,000 vaults per account. If you need to create more vaults, contact Amazon Glacier.

You must use the following guidelines when naming a vault.

- Names can be between 1 and 255 characters long.
- Allowed characters are a-z, A-Z, 0-9, ‘_’ (underscore), ‘-’ (hyphen), and ‘.’ (period).

This operation is idempotent.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don’t have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#).

For conceptual information and underlying REST API, go to [Creating a Vault in Amazon Glacier](#) and [Create Vault](#) in the *Amazon Glacier Developer Guide*.

Request Syntax

```
vault = account.create_vault(
    vaultName='string'
)
```

Parameters `vaultName` (*string*) – [REQUIRED]

The name of the vault.

Return type `glacier.Vault`

Returns Vault resource

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Vault (*name*)

Creates a Vault resource.:

```
vault = account.Vault('name')
```

Parameters *name* (*string*) – The Vault's name identifier. This **must** be set.

Return type `Glacier.Vault`

Returns A Vault resource

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

vaults

all ()

Creates an iterable of all Vault resources in the collection.

Request Syntax

```
vault_iterator = account.vaults.all()
```

Return type `list(glacier.Vault)`

Returns A list of Vault resources

filter (***kwargs*)

Creates an iterable of all Vault resources in the collection filtered by kwargs passed to method.

Request Syntax

```
vault_iterator = account.vaults.filter(
    marker='string',
    limit='string'
)
```

Parameters

- **marker** (*string*) – A string used for pagination. The marker specifies the vault ARN after which the listing of vaults should begin.
- **limit** (*string*) – The maximum number of items returned in the response. If you don't specify a value, the List Vaults operation returns up to 1,000 items.

Return type `list(glacier.Vault)`

Returns A list of Vault resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Vault resources in the collection.

Request Syntax


```

vault_iterator = account.vaults.limit(
    count=123
)

```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(glacier.Vault)`

Returns A list of Vault resources

page_size (***kwargs*)

Creates an iterable of all Vault resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```

vault_iterator = account.vaults.page_size(
    count=123
)

```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(glacier.Vault)`

Returns A list of Vault resources

Archive

class `Glacier.Archive` (*account_id*, *vault_name*, *id*)

A resource representing an Amazon Glacier Archive:

```
import boto3
```

```
glacier = boto3.resource('glacier')
archive = glacier.Archive('account_id', 'vault_name', 'id')
```

Parameters

- **account_id** (*string*) – The Archive’s `account_id` identifier. This **must** be set.
- **vault_name** (*string*) – The Archive’s `vault_name` identifier. This **must** be set.
- **id** (*string*) – The Archive’s `id` identifier. This **must** be set.

These are the resource’s available identifiers:

- `account_id`
- `vault_name`
- `id`

These are the resource’s available actions:

- `delete()`
- `initiate_archive_retrieval()`

These are the resource’s available sub-resources:

- `Vault()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

`account_id`

(*string*) The Archive’s `account_id` identifier. This **must** be set.

`vault_name`

(*string*) The Archive’s `vault_name` identifier. This **must** be set.

id

(*string*) The Archive's id identifier. This **must** be set.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

delete()

This operation deletes an archive from a vault. Subsequent requests to initiate a retrieval of this archive will fail. Archive retrievals that are in progress for this archive ID may or may not succeed according to the following scenarios:

- If the archive retrieval job is actively preparing the data for download when Amazon Glacier receives the delete archive request, the archival retrieval operation might fail.
- If the archive retrieval job has successfully prepared the archive for download when Amazon Glacier receives the delete archive request, you will be able to download the output.

This operation is idempotent. Attempting to delete an already-deleted archive does not result in an error.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#).

For conceptual information and underlying REST API, go to [Deleting an Archive in Amazon Glacier](#) and [Delete Archive](#) in the *Amazon Glacier Developer Guide*.

Request Syntax

```
response = archive.delete()
```

Returns None

initiate_archive_retrieval()

This operation initiates a job of the specified type. In this release, you can initiate a job to retrieve either an archive or a vault inventory (a list of archives in a vault).

Retrieving data from Amazon Glacier is a two-step process:

- Initiate a retrieval job. .. note::A data retrieval policy can cause your initiate retrieval job request to fail with a `PolicyEnforcedException` exception. For more information about data retrieval policies, see [Amazon Glacier Data Retrieval Policies](#). For more information about the `PolicyEnforcedException` exception, see [Error Responses](#).
- After the job completes, download the bytes.

The retrieval request is executed asynchronously. When you initiate a retrieval job, Amazon Glacier creates a job and returns a job ID in the response. When Amazon Glacier completes the job, you can get the job output (archive or inventory data). For information about getting job output, see `GetJobOutput` operation.

The job must complete before you can get its output. To determine when a job is complete, you have the following options:

- Use Amazon SNS Notification** You can specify an Amazon Simple Notification Service (Amazon SNS) topic to which Amazon Glacier can post a notification after the job is completed. You can specify an SNS topic per job request. The notification is sent only after Amazon Glacier completes the job. In addition to specifying an SNS topic per job request, you can configure vault notifications for a vault so that job notifications are always sent. For more information, see `SetVaultNotifications`.
- Get job details** You can make a `DescribeJob` request to obtain job status information while a job is in progress. However, it is more efficient to use an Amazon SNS notification to determine when a job is complete.

Note: The information you get via notification is same that you get by calling `DescribeJob` .

If for a specific event, you add both the notification configuration on the vault and also specify an SNS topic in your initiate job request, Amazon Glacier sends both notifications. For more information, see `SetVaultNotifications` .

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#) .

About the Vault Inventory

Amazon Glacier prepares an inventory for each vault periodically, every 24 hours. When you initiate a job for a vault inventory, Amazon Glacier returns the last inventory for the vault. The inventory data you get might be up to a day or two days old. Also, the initiate inventory job might take some time to complete before you can download the vault inventory. So you do not want to retrieve a vault inventory for each vault operation. However, in some scenarios, you might find the vault inventory useful. For example, when you upload an archive, you can provide an archive description but not an archive name. Amazon Glacier provides you a unique archive ID, an opaque string of characters. So, you might maintain your own database that maps archive names to their corresponding Amazon Glacier assigned archive IDs. You might find the vault inventory useful in the event you need to reconcile information in your database with the actual vault inventory.

Range Inventory Retrieval

You can limit the number of inventory items retrieved by filtering on the archive creation date or by setting a limit.

Filtering by Archive Creation Date

You can retrieve inventory items for archives created between `StartDate` and `EndDate` by specifying values for these parameters in the **InitiateJob** request. Archives created on or after the `StartDate` and before the `EndDate` will be returned. If you only provide the `StartDate` without the `EndDate` , you will retrieve the inventory for all archives created on or after the `StartDate` . If you only provide the `EndDate` without the `StartDate` , you will get back the inventory for all archives created before the `EndDate` .

Limiting Inventory Items per Retrieval

You can limit the number of inventory items returned by setting the `Limit` parameter in the **InitiateJob** request. The inventory job output will contain inventory items up to the specified `Limit` . If there are more inventory items available, the result is paginated. After a job is complete you can use the `DescribeJob` operation to get a marker that you use in a subsequent **InitiateJob** request. The marker will indicate the starting point to retrieve the next set of inventory items. You can page through your entire inventory by repeatedly making **InitiateJob** requests with the marker from the previous **DescribeJob** output, until you get a marker from **DescribeJob** that returns null, indicating that there are no more inventory items available.

You can use the `Limit` parameter together with the date range parameters.

About Ranged Archive Retrieval

You can initiate an archive retrieval for the whole archive or a range of the archive. In the case of ranged archive retrieval, you specify a byte range to return or the whole archive. The range specified must be megabyte (MB) aligned, that is the range start value must be divisible by 1 MB and range end value plus 1 must be divisible by 1 MB or equal the end of the archive. If the ranged archive retrieval is not megabyte aligned, this operation returns a 400 response. Furthermore, to ensure you get checksum values for data you download using `Get Job Output API`, the range must be tree hash aligned.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#) .

For conceptual information and the underlying REST API, go to [Initiate a Job and Downloading a Vault Inventory](#)

Request Syntax

```
job = archive.initiate_archive_retrieval()
```

Return type `glacier.Job`

Returns Job resource

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

`Vault()`

Creates a Vault resource.:

```
vault = archive.Vault()
```

Return type `Glacier.Vault`

Returns A Vault resource

Job

class `Glacier.Job(account_id, vault_name, id)`

A resource representing an Amazon Glacier Job:

```
import boto3
```

```
glacier = boto3.resource('glacier')
job = glacier.Job('account_id', 'vault_name', 'id')
```

Parameters

- **account_id** (*string*) – The Job's account_id identifier. This **must** be set.
- **vault_name** (*string*) – The Job's vault_name identifier. This **must** be set.
- **id** (*string*) – The Job's id identifier. This **must** be set.

These are the resource's available identifiers:

- `account_id`
- `vault_name`
- `id`

These are the resource's available attributes:

- `action`
- `archive_id`
- `archive_sha256_tree_hash`
- `archive_size_in_bytes`
- `completed`
- `completion_date`
- `creation_date`
- `inventory_retrieval_parameters`
- `inventory_size_in_bytes`

- `job_description`
- `job_id`
- `retrieval_byte_range`
- `sha256_tree_hash`
- `sns_topic`
- `status_code`
- `status_message`
- `vault_arn`

These are the resource's available actions:

- `get_output()`
- `load()`
- `reload()`

These are the resource's available sub-resources:

- `Vault()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

account_id

(*string*) The Job's `account_id` identifier. This **must** be set.

vault_name

(*string*) The Job's `vault_name` identifier. This **must** be set.

id

(*string*) The Job's `id` identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

action

(*string*)

The job type. It is either `ArchiveRetrieval` or `InventoryRetrieval`.

archive_id

(*string*)

For an `ArchiveRetrieval` job, this is the archive ID requested for download. Otherwise, this field is null.

archive_sha256_tree_hash

(*string*)

The SHA256 tree hash of the entire archive for an archive retrieval. For inventory retrieval jobs, this field is null.

archive_size_in_bytes

(*integer*)

For an `ArchiveRetrieval` job, this is the size in bytes of the archive being requested for download. For the `InventoryRetrieval` job, the value is null.

completed

(*boolean*)

The job status. When a job is completed, you get the job's output.

completion_date

(*string*)

The UTC time that the archive retrieval request completed. While the job is in progress, the value will be null.

creation_date

(string)

The UTC date when the job was created. A string representation of ISO 8601 date format, for example, “2012-03-20T17:03:43.221Z”.

inventory_retrieval_parameters

(dict)

Parameters used for range inventory retrieval.

inventory_size_in_bytes

(integer)

For an InventoryRetrieval job, this is the size in bytes of the inventory requested for download. For the ArchiveRetrieval job, the value is null.

job_description

(string)

The job description you provided when you initiated the job.

job_id

(string)

An opaque string that identifies an Amazon Glacier job.

retrieval_byte_range

(string)

The retrieved byte range for archive retrieval jobs in the form “*StartByteValue* - *EndByteValue*”. If no range was specified in the archive retrieval, then the whole archive is retrieved and *StartByteValue* equals 0 and *EndByteValue* equals the size of the archive minus 1. For inventory retrieval jobs this field is null.

sha256_tree_hash

(string)

For an ArchiveRetrieval job, it is the checksum of the archive. Otherwise, the value is null.

The SHA256 tree hash value for the requested range of an archive. If the Initiate a Job request for an archive specified a tree-hash aligned range, then this field returns a value.

For the specific case when the whole archive is retrieved, this value is the same as the Archive-SHA256TreeHash value.

This field is null in the following situations:

- Archive retrieval jobs that specify a range that is not tree-hash aligned.
- Archival jobs that specify a range that is equal to the whole archive and the job status is InProgress.
- Inventory jobs.

sns_topic

(string)

An Amazon Simple Notification Service (Amazon SNS) topic that receives notification.

status_code

(string)

The status code can be InProgress, Succeeded, or Failed, and indicates the status of the job.

status_message*(string)*

A friendly message that describes the job status.

vault_arn*(string)*

The Amazon Resource Name (ARN) of the vault from which the archive retrieval was requested.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

get_output (***kwargs*)

This operation downloads the output of the job you initiated using `InitiateJob`. Depending on the job type you specified when you initiated the job, the output will be either the content of an archive or a vault inventory.

A job ID will not expire for at least 24 hours after Amazon Glacier completes the job. That is, you can download the job output within the 24 hours period after Amazon Glacier completes the job.

If the job output is large, then you can use the `Range` request header to retrieve a portion of the output. This allows you to download the entire output in smaller chunks of bytes. For example, suppose you have 1 GB of job output you want to download and you decide to download 128 MB chunks of data at a time, which is a total of eight Get Job Output requests. You use the following process to download the job output:

- Download a 128 MB chunk of output by specifying the appropriate byte range using the `Range` header.
- Along with the data, the response includes a SHA256 tree hash of the payload. You compute the checksum of the payload on the client and compare it with the checksum you received in the response to ensure you received all the expected data.
- Repeat steps 1 and 2 for all the eight 128 MB chunks of output data, each time specifying the appropriate byte range.
- After downloading all the parts of the job output, you have a list of eight checksum values. Compute the tree hash of these values to find the checksum of the entire output. Using the `DescribeJob` API, obtain job information of the job that provided you the output. The response includes the checksum of the entire archive stored in Amazon Glacier. You compare this value with the checksum you computed to ensure you have downloaded the entire archive content with no errors.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#).

For conceptual information and the underlying REST API, go to [Downloading a Vault Inventory](#), [Downloading an Archive](#), and [Get Job Output](#).

Request Syntax

```
response = job.get_output(
    range='string'
)
```

Parameters `range` (*string*) – The range of bytes to retrieve from the output. For example, if you want to download the first 1,048,576 bytes, specify “Range: bytes=0-1048575”. By default, this operation downloads the entire output.

Return type dict

Returns

Response Syntax

```
{
    'body': b'bytes',
    'checksum': 'string',
    'status': 123,
    'contentRange': 'string',
    'acceptRanges': 'string',
    'contentType': 'string',
    'archiveDescription': 'string'
}
```

Response Structure

- (*dict*) –

Contains the Amazon Glacier response to your request.

- **body** (*bytes*) –

The job data, either archive data or inventory data.

- **checksum** (*string*) –

The checksum of the data in the response. This header is returned only when retrieving the output for an archive retrieval job. Furthermore, this header appears only under the following conditions:

- * You get the entire range of the archive.
- * You request a range to return of the archive that starts and ends on a multiple of 1 MB. For example, if you have an 3.1 MB archive and you specify a range to return that starts at 1 MB and ends at 2 MB, then the x-amz-sha256-tree-hash is returned as a response header.
- * You request a range of the archive to return that starts on a multiple of 1 MB and goes to the end of the archive. For example, if you have a 3.1 MB archive and you specify a range that starts at 2 MB and ends at 3.1 MB (the end of the archive), then the x-amz-sha256-tree-hash is returned as a response header.

- **status** (*integer*) –

The HTTP response code for a job output request. The value depends on whether a range was specified in the request.

- **contentRange** (*string*) –

The range of bytes returned by Amazon Glacier. If only partial output is downloaded, the response provides the range of bytes Amazon Glacier returned. For example, bytes 0-1048575/8388608 returns the first 1 MB from 8 MB.

- **acceptRanges** (*string*) –

Indicates the range units accepted. For more information, go to [RFC2616](#).

- **contentType** (*string*) –

The Content-Type depends on whether the job output is an archive or a vault inventory. For archive data, the Content-Type is application/octet-stream. For vault inventory, if you requested CSV format when you initiated the job, the Content-Type is text/csv. Otherwise, by default, vault inventory is returned as JSON, and the Content-Type is application/json.

- **archiveDescription** (*string*) –

The description of an archive.

load()

Calls `glacier.Client.describe_job()` to update the attributes of the Job resource

Request Syntax

```
job.load()
```

Returns None

reload()

Calls `glacier.Client.describe_job()` to update the attributes of the Job resource

Request Syntax

```
job.reload()
```

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Vault()

Creates a Vault resource.:

```
vault = job.Vault()
```

Return type `Glacier.Vault`

Returns A Vault resource

MultipartUpload

class `Glacier.MultipartUpload(account_id, vault_name, id)`

A resource representing an Amazon Glacier MultipartUpload:

```
import boto3
```

```
glacier = boto3.resource('glacier')
```

```
multipart_upload = glacier.MultipartUpload('account_id', 'vault_name', 'id')
```

Parameters

- **account_id** (*string*) – The MultipartUpload's `account_id` identifier. This **must** be set.
- **vault_name** (*string*) – The MultipartUpload's `vault_name` identifier. This **must** be set.
- **id** (*string*) – The MultipartUpload's `id` identifier. This **must** be set.

These are the resource's available identifiers:

- `account_id`
- `vault_name`
- `id`

These are the resource's available attributes:

- `archive_description`
- `creation_date`
- `multipart_upload_id`
- `part_size_in_bytes`
- `vault_arn`

These are the resource's available actions:

- `abort()`
- `complete()`
- `parts()`
- `upload_part()`

These are the resource's available sub-resources:

- `Vault()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

account_id

(*string*) The MultipartUpload's account_id identifier. This **must** be set.

vault_name

(*string*) The MultipartUpload's vault_name identifier. This **must** be set.

id

(*string*) The MultipartUpload's id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

archive_description

(*string*)

The description of the archive that was specified in the Initiate Multipart Upload request.

creation_date

(*string*)

The UTC time at which the multipart upload was initiated.

multipart_upload_id

(*string*)

The ID of a multipart upload.

part_size_in_bytes

(*integer*)

The part size, in bytes, specified in the Initiate Multipart Upload request. This is the size of all the parts in the upload except the last part, which may be smaller than this size.

vault_arn

(*string*)

The Amazon Resource Name (ARN) of the vault that contains the archive.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

abort()

This operation aborts a multipart upload identified by the upload ID.

After the Abort Multipart Upload request succeeds, you cannot upload any more parts to the multipart upload or complete the multipart upload. Aborting a completed upload fails. However, aborting an already-aborted upload will succeed, for a short time. For more information about uploading a part and completing a multipart upload, see `UploadMultipartPart` and `CompleteMultipartUpload`.

This operation is idempotent.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#) .

For conceptual information and underlying REST API, go to [Working with Archives in Amazon Glacier](#) and [Abort Multipart Upload](#) in the *Amazon Glacier Developer Guide* .

Request Syntax

```
response = multipart_upload.abort()
```

Returns None

complete (**kwargs)

You call this operation to inform Amazon Glacier that all the archive parts have been uploaded and that Amazon Glacier can now assemble the archive from the uploaded parts. After assembling and saving the archive to the vault, Amazon Glacier returns the URI path of the newly created archive resource. Using the URI path, you can then access the archive. After you upload an archive, you should save the archive ID returned to retrieve the archive at a later point. You can also get the vault inventory to obtain a list of archive IDs in a vault. For more information, see [InitiateJob](#) .

In the request, you must include the computed SHA256 tree hash of the entire archive you have uploaded. For information about computing a SHA256 tree hash, see [Computing Checksums](#) . On the server side, Amazon Glacier also constructs the SHA256 tree hash of the assembled archive. If the values match, Amazon Glacier saves the archive to the vault; otherwise, it returns an error, and the operation fails. The [ListParts](#) operation returns a list of parts uploaded for a specific multipart upload. It includes checksum information for each uploaded part that can be used to debug a bad checksum issue.

Additionally, Amazon Glacier also checks for any missing content ranges when assembling the archive, if missing content ranges are found, Amazon Glacier returns an error and the operation fails.

Complete Multipart Upload is an idempotent operation. After your first successful complete multipart upload, if you call the operation again within a short period, the operation will succeed and return the same archive ID. This is useful in the event you experience a network issue that causes an aborted connection or receive a 500 server error, in which case you can repeat your Complete Multipart Upload request and get the same archive ID without creating duplicate archives. Note, however, that after the multipart upload completes, you cannot call the [List Parts](#) operation and the multipart upload will not appear in [List Multipart Uploads](#) response, even if idempotent complete is possible.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#) .

For conceptual information and underlying REST API, go to [Uploading Large Archives in Parts \(Multipart Upload\)](#) and [Complete Multipart Upload](#) in the *Amazon Glacier Developer Guide* .

Request Syntax

```
response = multipart_upload.complete(
    archiveSize='string',
)
```

Parameters

- **archiveSize** (*string*) – The total size, in bytes, of the entire archive. This value should be the sum of all the sizes of the individual parts that you uploaded.
- **checksum** (*string*) – The SHA256 tree hash of the entire archive. It is the tree hash of SHA256 tree hash of the individual parts. If the value you specify in the request does not match the SHA256 tree hash of the final assembled archive as computed by Amazon Glacier, Amazon Glacier returns an error and the request fails.

Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required

Return type dict

Returns

Response Syntax

```
{
    'location': 'string',
    'checksum': 'string',
    'archiveId': 'string'
}
```

Response Structure

- (*dict*) –

Contains the Amazon Glacier response to your request.

For information about the underlying REST API, go to [Upload Archive](#) . For conceptual information, go to [Working with Archives in Amazon Glacier](#) .

- **location** (*string*) –

The relative URI path of the newly added archive resource.

- **checksum** (*string*) –

The checksum of the archive computed by Amazon Glacier.

- **archiveId** (*string*) –

The ID of the archive. This value is also included as part of the location.

parts (***kwargs*)

This operation lists the parts of an archive that have been uploaded in a specific multipart upload. You can make this request at any time during an in-progress multipart upload before you complete the upload (see [CompleteMultipartUpload](#) . List Parts returns an error for completed uploads. The list returned in the List Parts response is sorted by part range.

The List Parts operation supports pagination. By default, this operation returns up to 1,000 uploaded parts in the response. You should always check the response for a `marker` at which to continue the list; if there are no more items the `marker` is `null` . To return a list of parts that begins at a specific part, set the `marker` request parameter to the value you obtained from a previous List Parts request. You can also limit the number of parts returned in the response by specifying the `limit` parameter in the request.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#) .

For conceptual information and the underlying REST API, go to [Working with Archives in Amazon Glacier](#) and [List Parts](#) in the *Amazon Glacier Developer Guide* .

Request Syntax

```
response = multipart_upload.parts(
    marker='string',
    limit='string'
)
```

Parameters

- **marker** (*string*) – An opaque string used for pagination. This value specifies the part at which the listing of parts should begin. Get the marker value from the response of a previous List Parts response. You need only include the marker if you are continuing the pagination of results started in a previous List Parts request.
- **limit** (*string*) – Specifies the maximum number of parts returned in the response body. If this value is not specified, the List Parts operation returns up to 1,000 uploads.

Return type dict

Returns

Response Syntax

```
{
  'MultipartUploadId': 'string',
  'VaultARN': 'string',
  'ArchiveDescription': 'string',
  'PartSizeInBytes': 123,
  'CreationDate': 'string',
  'Parts': [
    {
      'RangeInBytes': 'string',
      'SHA256TreeHash': 'string'
    },
  ],
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Contains the Amazon Glacier response to your request.

- **MultipartUploadId** (*string*) –

The ID of the upload to which the parts are associated.

- **VaultARN** (*string*) –

The Amazon Resource Name (ARN) of the vault to which the multipart upload was initiated.

- **ArchiveDescription** (*string*) –

The description of the archive that was specified in the Initiate Multipart Upload request.

- **PartSizeInBytes** (*integer*) –

The part size in bytes.

- **CreationDate** (*string*) –

The UTC time at which the multipart upload was initiated.

- **Parts** (*list*) –

A list of the part sizes of the multipart upload.

* (*dict*) –

A list of the part sizes of the multipart upload.

- **RangeInBytes** (*string*) –

The byte range of a part, inclusive of the upper value of the range.

- **SHA256TreeHash** (*string*) –

The SHA256 tree hash value that Amazon Glacier calculated for the part. This field is never `null`.

- **Marker** (*string*) –

An opaque string that represents where to continue pagination of the results. You use the marker in a new List Parts request to obtain more jobs in the list. If there are no more parts, this value is `null`.

upload_part (***kwargs*)

This operation uploads a part of an archive. You can upload archive parts in any order. You can also upload them in parallel. You can upload up to 10,000 parts for a multipart upload.

Amazon Glacier rejects your upload part request if any of the following conditions is true:

- **SHA256 tree hash does not match** To ensure that part data is not corrupted in transmission, you compute a SHA256 tree hash of the part and include it in your request. Upon receiving the part data, Amazon Glacier also computes a SHA256 tree hash. If these hash values don't match, the operation fails. For information about computing a SHA256 tree hash, see [Computing Checksums](#).

- **Part size does not match** The size of each part except the last must match the size specified in the corresponding InitiateMultipartUpload request. The size of the last part must be the same size as, or smaller than, the specified size. .. note::If you upload a part whose size is smaller than the part size you specified in your initiate multipart upload request and that part is not the last part, then the upload part request will succeed. However, the subsequent Complete Multipart Upload request will fail.

- **Range does not align** The byte range value in the request does not align with the part size specified in the corresponding initiate request. For example, if you specify a part size of 4194304 bytes (4 MB), then 0 to 4194303 bytes (4 MB - 1) and 4194304 (4 MB) to 8388607 (8 MB - 1) are valid part ranges. However, if you set a range value of 2 MB to 6 MB, the range does not align with the part size and the upload will fail.

This operation is idempotent. If you upload the same part multiple times, the data included in the most recent request overwrites the previously uploaded data.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#).

For conceptual information and underlying REST API, go to [Uploading Large Archives in Parts \(Multipart Upload\)](#) and [Upload Part](#) in the *Amazon Glacier Developer Guide*.

Request Syntax

```
response = multipart_upload.upload_part(  
    range='string',  
    body=b'bytes'  
)
```

Parameters

- **checksum** (*string*) – The SHA256 tree hash of the data being uploaded.

Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required

- **range** (*string*) – Identifies the range of bytes in the assembled archive that will be uploaded in this part. Amazon Glacier uses this information to assemble the archive in the proper sequence. The format of this header follows RFC 2616. An example header is Content-Range:bytes 0-4194303/*.
- **body** (*bytes*) – The data to upload.

Return type dict

Returns

Response Syntax

```
{
    'checksum': 'string'
}
```

Response Structure

- (*dict*) –

Contains the Amazon Glacier response to your request.

- **checksum** (*string*) –

The SHA256 tree hash that Amazon Glacier computed for the uploaded part.

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Vault ()

Creates a Vault resource.:

```
vault = multipart_upload.Vault()
```

Return type `Glacier.Vault`

Returns A Vault resource

Notification

class `Glacier.Notification` (*account_id*, *vault_name*)

A resource representing an Amazon Glacier Notification:

```
import boto3
```

```
glacier = boto3.resource('glacier')
```

```
notification = glacier.Notification('account_id', 'vault_name')
```

Parameters

- **account_id** (*string*) – The Notification's `account_id` identifier. This **must** be set.
- **vault_name** (*string*) – The Notification's `vault_name` identifier. This **must** be set.

These are the resource's available identifiers:

- `account_id`
- `vault_name`

These are the resource's available attributes:

- `events`

- `sns_topic`

These are the resource's available actions:

- `delete()`
- `load()`
- `reload()`
- `set()`

These are the resource's available sub-resources:

- `Vault()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

account_id

(*string*) The Notification's `account_id` identifier. This **must** be set.

vault_name

(*string*) The Notification's `vault_name` identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

events

(*list*)

A list of one or more events for which Amazon Glacier will send a notification to the specified Amazon SNS topic.

sns_topic

(*string*)

The Amazon Simple Notification Service (Amazon SNS) topic Amazon Resource Name (ARN).

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

delete()

This operation deletes the notification configuration set for a vault. The operation is eventually consistent; that is, it might take some time for Amazon Glacier to completely disable the notifications and you might still receive some notifications for a short time after you send the delete request.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#).

For conceptual information and underlying REST API, go to [Configuring Vault Notifications in Amazon Glacier](#) and [Delete Vault Notification Configuration](#) in the Amazon Glacier Developer Guide.

Request Syntax

```
response = notification.delete()
```

Returns None

load()

Calls `glacier.Client.get_vault_notifications()` to update the attributes of the Notification resource

Request Syntax

```
notification.load()
```

Returns None**reload()**

Calls `glacier.Client.get_vault_notifications()` to update the attributes of the Notification resource

Request Syntax

```
notification.reload()
```

Returns None**set (**kwargs)**

This operation configures notifications that will be sent when specific events happen to a vault. By default, you don't get any notifications.

To configure vault notifications, send a PUT request to the `notification-configuration` subresource of the vault. The request should include a JSON document that provides an Amazon SNS topic and specific events for which you want Amazon Glacier to send notifications to the topic.

Amazon SNS topics must grant permission to the vault to be allowed to publish notifications to the topic. You can configure a vault to publish a notification for the following vault events:

- **ArchiveRetrievalCompleted** This event occurs when a job that was initiated for an archive retrieval is completed (`InitiateJob`). The status of the completed job can be “Succeeded” or “Failed”. The notification sent to the SNS topic is the same output as returned from `DescribeJob` .
- **InventoryRetrievalCompleted** This event occurs when a job that was initiated for an inventory retrieval is completed (`InitiateJob`). The status of the completed job can be “Succeeded” or “Failed”. The notification sent to the SNS topic is the same output as returned from `DescribeJob` .

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#) .

For conceptual information and underlying REST API, go to [Configuring Vault Notifications in Amazon Glacier](#) and [Set Vault Notification Configuration](#) in the *Amazon Glacier Developer Guide* .

Request Syntax

```
response = notification.set(
    vaultNotificationConfig={
        'SNSTopic': 'string',
        'Events': [
            'string',
        ]
    }
)
```

Parameters `vaultNotificationConfig` (*dict*) – Provides options for specifying notification configuration.

- **SNSTopic** (*string*) –

The Amazon Simple Notification Service (Amazon SNS) topic Amazon Resource Name (ARN).

- **Events** (*list*) –

A list of one or more events for which Amazon Glacier will send a notification to the specified Amazon SNS topic.

– (*string*) –

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource’s identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Vault ()

Creates a Vault resource.:

```
vault = notification.Vault()
```

Return type `Glacier.Vault`

Returns A Vault resource

Vault

class `Glacier.Vault` (*account_id*, *name*)

A resource representing an Amazon Glacier Vault:

```
import boto3
```

```
glacier = boto3.resource('glacier')
vault = glacier.Vault('account_id', 'name')
```

Parameters

- **account_id** (*string*) – The Vault’s `account_id` identifier. This **must** be set.
- **name** (*string*) – The Vault’s `name` identifier. This **must** be set.

These are the resource’s available identifiers:

- `account_id`
- `name`

These are the resource’s available attributes:

- `creation_date`
- `last_inventory_date`
- `number_of_archives`
- `size_in_bytes`
- `vault_arn`
- `vault_name`

These are the resource’s available actions:

- `create()`
- `delete()`
- `initiate_inventory_retrieval()`
- `initiate_multipart_upload()`
- `load()`
- `reload()`
- `upload_archive()`

These are the resource’s available sub-resources:

- `Account()`
- `Archive()`
- `Job()`

- `MultipartUpload()`
- `Notification()`

These are the resource's available collections:

- `completed_jobs`
- `failed_jobs`
- `jobs`
- `jobs_in_progress`
- `multipart_uplaods`
- `succeeded_jobs`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

account_id

(*string*) The Vault's `account_id` identifier. This **must** be set.

name

(*string*) The Vault's `name` identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

creation_date

(*string*)

The UTC date when the vault was created. A string representation of ISO 8601 date format, for example, "2012-03-20T17:03:43.221Z".

last_inventory_date

(*string*)

The UTC date when Amazon Glacier completed the last vault inventory. A string representation of ISO 8601 date format, for example, "2012-03-20T17:03:43.221Z".

number_of_archives

(*integer*)

The number of archives in the vault as of the last inventory date. This field will return `null` if an inventory has not yet run on the vault, for example, if you just created the vault.

size_in_bytes

(*integer*)

Total size, in bytes, of the archives in the vault as of the last inventory date. This field will return `null` if an inventory has not yet run on the vault, for example, if you just created the vault.

vault_arn

(*string*)

The Amazon Resource Name (ARN) of the vault.

vault_name

(*string*)

The name of the vault.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

create()

This operation creates a new vault with the specified name. The name of the vault must be unique within a region for an AWS account. You can create up to 1,000 vaults per account. If you need to create more vaults, contact Amazon Glacier.

You must use the following guidelines when naming a vault.

- Names can be between 1 and 255 characters long.
- Allowed characters are a-z, A-Z, 0-9, '_' (underscore), '-' (hyphen), and '.' (period).

This operation is idempotent.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#).

For conceptual information and underlying REST API, go to [Creating a Vault in Amazon Glacier](#) and [Create Vault](#) in the *Amazon Glacier Developer Guide*.

Request Syntax

```
response = vault.create()
```

Return type dict

Returns**Response Syntax**

```
{
    'location': 'string'
}
```

Response Structure

- (*dict*) –

Contains the Amazon Glacier response to your request.

- **location** (*string*) –

The URI of the vault that was created.

delete()

This operation deletes a vault. Amazon Glacier will delete a vault only if there are no archives in the vault as of the last inventory and there have been no writes to the vault since the last inventory. If either of these conditions is not satisfied, the vault deletion fails (that is, the vault is not removed) and Amazon Glacier returns an error. You can use `DescribeVault` to return the number of archives in a vault, and you can use [Initiate a Job \(POST jobs\)](#) to initiate a new inventory retrieval for a vault. The inventory contains the archive IDs you use to delete archives using [Delete Archive \(DELETE archive\)](#).

This operation is idempotent.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#).

For conceptual information and underlying REST API, go to [Deleting a Vault in Amazon Glacier](#) and [Delete Vault](#) in the *Amazon Glacier Developer Guide*.

Request Syntax

```
response = vault.delete()
```

Returns None

`initiate_inventory_retrieval()`

This operation initiates a job of the specified type. In this release, you can initiate a job to retrieve either an archive or a vault inventory (a list of archives in a vault).

Retrieving data from Amazon Glacier is a two-step process:

- Initiate a retrieval job. .. note::A data retrieval policy can cause your initiate retrieval job request to fail with a `PolicyEnforcedException` exception. For more information about data retrieval policies, see [Amazon Glacier Data Retrieval Policies](#) . For more information about the `PolicyEnforcedException` exception, see [Error Responses](#) .
- After the job completes, download the bytes.

The retrieval request is executed asynchronously. When you initiate a retrieval job, Amazon Glacier creates a job and returns a job ID in the response. When Amazon Glacier completes the job, you can get the job output (archive or inventory data). For information about getting job output, see `GetJobOutput` operation.

The job must complete before you can get its output. To determine when a job is complete, you have the following options:

- Use Amazon SNS Notification** You can specify an Amazon Simple Notification Service (Amazon SNS) topic to which Amazon Glacier can post a notification after the job is completed. You can specify an SNS topic per job request. The notification is sent only after Amazon Glacier completes the job. In addition to specifying an SNS topic per job request, you can configure vault notifications for a vault so that job notifications are always sent. For more information, see `SetVaultNotifications` .
- Get job details** You can make a `DescribeJob` request to obtain job status information while a job is in progress. However, it is more efficient to use an Amazon SNS notification to determine when a job is complete.

Note: The information you get via notification is same that you get by calling `DescribeJob` .

If for a specific event, you add both the notification configuration on the vault and also specify an SNS topic in your initiate job request, Amazon Glacier sends both notifications. For more information, see `SetVaultNotifications` .

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#) .

About the Vault Inventory

Amazon Glacier prepares an inventory for each vault periodically, every 24 hours. When you initiate a job for a vault inventory, Amazon Glacier returns the last inventory for the vault. The inventory data you get might be up to a day or two days old. Also, the initiate inventory job might take some time to complete before you can download the vault inventory. So you do not want to retrieve a vault inventory for each vault operation. However, in some scenarios, you might find the vault inventory useful. For example, when you upload an archive, you can provide an archive description but not an archive name. Amazon Glacier provides you a unique archive ID, an opaque string of characters. So, you might maintain your own database that maps archive names to their corresponding Amazon Glacier assigned archive IDs. You might find the vault inventory useful in the event you need to reconcile information in your database with the actual vault inventory.

Range Inventory Retrieval

You can limit the number of inventory items retrieved by filtering on the archive creation date or by setting a limit.

Filtering by Archive Creation Date

You can retrieve inventory items for archives created between `StartDate` and `EndDate` by specifying values for these parameters in the **InitiateJob** request. Archives created on or after the `StartDate` and before the `EndDate` will be returned. If you only provide the `StartDate` without the `EndDate`, you will retrieve the inventory for all archives created on or after the `StartDate`. If you only provide the `EndDate` without the `StartDate`, you will get back the inventory for all archives created before the `EndDate`.

Limiting Inventory Items per Retrieval

You can limit the number of inventory items returned by setting the `Limit` parameter in the **InitiateJob** request. The inventory job output will contain inventory items up to the specified `Limit`. If there are more inventory items available, the result is paginated. After a job is complete you can use the `DescribeJob` operation to get a marker that you use in a subsequent **InitiateJob** request. The marker will indicate the starting point to retrieve the next set of inventory items. You can page through your entire inventory by repeatedly making **InitiateJob** requests with the marker from the previous **DescribeJob** output, until you get a marker from **DescribeJob** that returns null, indicating that there are no more inventory items available.

You can use the `Limit` parameter together with the date range parameters.

About Ranged Archive Retrieval

You can initiate an archive retrieval for the whole archive or a range of the archive. In the case of ranged archive retrieval, you specify a byte range to return or the whole archive. The range specified must be megabyte (MB) aligned, that is the range start value must be divisible by 1 MB and range end value plus 1 must be divisible by 1 MB or equal the end of the archive. If the ranged archive retrieval is not megabyte aligned, this operation returns a 400 response. Furthermore, to ensure you get checksum values for data you download using Get Job Output API, the range must be tree hash aligned.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#).

For conceptual information and the underlying REST API, go to [Initiate a Job](#) and [Downloading a Vault Inventory](#)

Request Syntax

```
job = vault.initiate_inventory_retrieval()
```

Return type `glacier.Job`

Returns Job resource

initiate_multipart_upload(kwargs)**

This operation initiates a multipart upload. Amazon Glacier creates a multipart upload resource and returns its ID in the response. The multipart upload ID is used in subsequent requests to upload parts of an archive (see `UploadMultipartPart`).

When you initiate a multipart upload, you specify the part size in number of bytes. The part size must be a megabyte (1024 KB) multiplied by a power of 2—for example, 1048576 (1 MB), 2097152 (2 MB), 4194304 (4 MB), 8388608 (8 MB), and so on. The minimum allowable part size is 1 MB, and the maximum is 4 GB.

Every part you upload to this resource (see `UploadMultipartPart`), except the last one, must have the same size. The last one can be the same size or smaller. For example, suppose you want to upload a 16.2 MB

file. If you initiate the multipart upload with a part size of 4 MB, you will upload four parts of 4 MB each and one part of 0.2 MB.

Note: You don't need to know the size of the archive when you start a multipart upload because Amazon Glacier does not require you to specify the overall archive size.

After you complete the multipart upload, Amazon Glacier removes the multipart upload resource referenced by the ID. Amazon Glacier also removes the multipart upload resource if you cancel the multipart upload or it may be removed if there is no activity for a period of 24 hours.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#).

For conceptual information and underlying REST API, go to [Uploading Large Archives in Parts \(Multipart Upload\)](#) and [Initiate Multipart Upload](#) in the *Amazon Glacier Developer Guide*.

Request Syntax

```
multipart_upload = vault.initiate_multipart_upload(
    archiveDescription='string',
    partSize='string'
)
```

Parameters

- **archiveDescription** (*string*) – The archive description that you are uploading in parts.
The part size must be a megabyte (1024 KB) multiplied by a power of 2, for example 1048576 (1 MB), 2097152 (2 MB), 4194304 (4 MB), 8388608 (8 MB), and so on. The minimum allowable part size is 1 MB, and the maximum is 4 GB (4096 MB).
- **partSize** (*string*) – The size of each part except the last, in bytes. The last part can be smaller than this part size.

Return type `glacier.MultipartUpload`

Returns MultipartUpload resource

load()

Calls `glacier.Client.describe_vault()` to update the attributes of the Vault resource

Request Syntax

```
vault.load()
```

Returns None

reload()

Calls `glacier.Client.describe_vault()` to update the attributes of the Vault resource

Request Syntax

```
vault.reload()
```

Returns None

upload_archive(**kwargs)

This operation adds an archive to a vault. This is a synchronous operation, and for a successful upload,

your data is durably persisted. Amazon Glacier returns the archive ID in the `x-amz-archive-id` header of the response.

You must use the archive ID to access your data in Amazon Glacier. After you upload an archive, you should save the archive ID returned so that you can retrieve or delete the archive later. Besides saving the archive ID, you can also index it and give it a friendly name to allow for better searching. You can also use the optional archive description field to specify how the archive is referred to in an external index of archives, such as you might create in Amazon DynamoDB. You can also get the vault inventory to obtain a list of archive IDs in a vault. For more information, see `InitiateJob` .

You must provide a SHA256 tree hash of the data you are uploading. For information about computing a SHA256 tree hash, see [Computing Checksums](#) .

You can optionally specify an archive description of up to 1,024 printable ASCII characters. You can get the archive description when you either retrieve the archive or get the vault inventory. For more information, see `InitiateJob` . Amazon Glacier does not interpret the description in any way. An archive description does not need to be unique. You cannot use the description to retrieve or sort the archive list.

Archives are immutable. After you upload an archive, you cannot edit the archive or its description.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [Access Control Using AWS Identity and Access Management \(IAM\)](#) .

For conceptual information and underlying REST API, go to [Uploading an Archive in Amazon Glacier](#) and [Upload Archive](#) in the *Amazon Glacier Developer Guide* .

Request Syntax

```
archive = vault.upload_archive(  
    archiveDescription='string',  
    body=b'bytes'  
)
```

Parameters

- **archiveDescription** (*string*) – The optional description of the archive you are uploading.
- **checksum** (*string*) – The SHA256 tree hash of the data being uploaded.
Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **body** (*bytes*) – The data to upload.

Return type `glacier.Archive`

Returns Archive resource

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Account ()

Creates a Account resource.:

```
account = vault.Account()
```

Return type `Glacier.Account`

Returns A Account resource

Archive (*id*)

Creates a Archive resource.:


```
archive = vault.Archive('id')
```

Parameters `id` (*string*) – The Archive’s id identifier. This **must** be set.

Return type `Glacier.Archive`

Returns A Archive resource

Job (*id*)

Creates a Job resource.:

```
job = vault.Job('id')
```

Parameters `id` (*string*) – The Job’s id identifier. This **must** be set.

Return type `Glacier.Job`

Returns A Job resource

MultipartUpload (*id*)

Creates a MultipartUpload resource.:

```
multipart_upload = vault.MultipartUpload('id')
```

Parameters `id` (*string*) – The MultipartUpload’s id identifier. This **must** be set.

Return type `Glacier.MultipartUpload`

Returns A MultipartUpload resource

Notification ()

Creates a Notification resource.:

```
notification = vault.Notification()
```

Return type `Glacier.Notification`

Returns A Notification resource

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

`completed_jobs`

all ()

Creates an iterable of all Job resources in the collection.

Request Syntax

```
job_iterator = vault.completed_jobs.all()
```

Return type `list(glacier.Job)`

Returns A list of Job resources

filter (***kwargs*)

Creates an iterable of all Job resources in the collection filtered by kwargs passed to method.

Request Syntax

```
job_iterator = vault.completed_jobs.filter(
    limit='string',
    marker='string',
    statuscode='string',
```

)

Parameters

- **limit** (*string*) – Specifies that the response be limited to the specified number of items or fewer. If not specified, the List Jobs operation returns up to 1,000 jobs.
- **marker** (*string*) – An opaque string used for pagination. This value specifies the job at which the listing of jobs should begin. Get the marker value from a previous List Jobs response. You need only include the marker if you are continuing the pagination of results started in a previous List Jobs request.
- **statuscode** (*string*) – Specifies the type of job status to return. You can specify the following values: “InProgress”, “Succeeded”, or “Failed”.

Return type list(glacier.Job)

Returns A list of Job resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Job resources in the collection.

Request Syntax

```
job_iterator = vault.completed_jobs.limit(  
    count=123  
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type list(glacier.Job)

Returns A list of Job resources

page_size (***kwargs*)

Creates an iterable of all Job resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
job_iterator = vault.completed_jobs.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type list(glacier.Job)

Returns A list of Job resources

failed_jobs

all ()

Creates an iterable of all Job resources in the collection.

Request Syntax

```
job_iterator = vault.failed_jobs.all()
```

Return type list(glacier.Job)

Returns A list of Job resources

filter (***kwargs*)

Creates an iterable of all Job resources in the collection filtered by kwargs passed to method.

Request Syntax

```
job_iterator = vault.failed_jobs.filter(
    limit='string',
    marker='string',
    completed='string'
)
```

Parameters

- **limit** (*string*) – Specifies that the response be limited to the specified number of items or fewer. If not specified, the List Jobs operation returns up to 1,000 jobs.
- **marker** (*string*) – An opaque string used for pagination. This value specifies the job at which the listing of jobs should begin. Get the marker value from a previous List Jobs response. You need only include the marker if you are continuing the pagination of results started in a previous List Jobs request.
- **completed** (*string*) – Specifies the state of the jobs to return. You can specify `true` or `false`.

Return type `list(glacier.Job)`

Returns A list of Job resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Job resources in the collection.

Request Syntax

```
job_iterator = vault.failed_jobs.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(glacier.Job)`

Returns A list of Job resources

page_size (***kwargs*)

Creates an iterable of all Job resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
job_iterator = vault.failed_jobs.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(glacier.Job)`

Returns A list of Job resources

jobs

all ()

Creates an iterable of all Job resources in the collection.

Request Syntax

```
job_iterator = vault.jobs.all()
```

Return type `list(glacier.Job)`

Returns A list of Job resources

filter (***kwargs*)

Creates an iterable of all Job resources in the collection filtered by kwargs passed to method.

Request Syntax

```
job_iterator = vault.jobs.filter(  
    limit='string',  
    marker='string',  
    statuscode='string',  
    completed='string'  
)
```

Parameters

- **limit** (*string*) – Specifies that the response be limited to the specified number of items or fewer. If not specified, the List Jobs operation returns up to 1,000 jobs.
- **marker** (*string*) – An opaque string used for pagination. This value specifies the job at which the listing of jobs should begin. Get the marker value from a previous List Jobs response. You need only include the marker if you are continuing the pagination of results started in a previous List Jobs request.
- **statuscode** (*string*) – Specifies the type of job status to return. You can specify the following values: “InProgress”, “Succeeded”, or “Failed”.
- **completed** (*string*) – Specifies the state of the jobs to return. You can specify `true` or `false`.

Return type `list(glacier.Job)`

Returns A list of Job resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Job resources in the collection.

Request Syntax

```
job_iterator = vault.jobs.limit(  
    count=123  
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(glacier.Job)`

Returns A list of Job resources

page_size (***kwargs*)

Creates an iterable of all Job resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
job_iterator = vault.jobs.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(glacier.Job)`

Returns A list of Job resources

jobs_in_progress**all()**

Creates an iterable of all Job resources in the collection.

Request Syntax

```
job_iterator = vault.jobs_in_progress.all()
```

Return type list(glacier.Job)**Returns** A list of Job resources**filter(**kwargs)**

Creates an iterable of all Job resources in the collection filtered by kwargs passed to method.

Request Syntax

```
job_iterator = vault.jobs_in_progress.filter(
    limit='string',
    marker='string',
    completed='string'
)
```

Parameters

- **limit** (*string*) – Specifies that the response be limited to the specified number of items or fewer. If not specified, the List Jobs operation returns up to 1,000 jobs.
- **marker** (*string*) – An opaque string used for pagination. This value specifies the job at which the listing of jobs should begin. Get the marker value from a previous List Jobs response. You need only include the marker if you are continuing the pagination of results started in a previous List Jobs request.
- **completed** (*string*) – Specifies the state of the jobs to return. You can specify `true` or `false`.

Return type list(glacier.Job)**Returns** A list of Job resources**limit(**kwargs)**

Creates an iterable up to a specified amount of Job resources in the collection.

Request Syntax

```
job_iterator = vault.jobs_in_progress.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.**Return type** list(glacier.Job)**Returns** A list of Job resources**page_size(**kwargs)**

Creates an iterable of all Job resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
job_iterator = vault.jobs_in_progress.page_size(  
    count=123  
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(glacier.Job)`

Returns A list of Job resources

multipart_uploads

all()

Creates an iterable of all MultipartUpload resources in the collection.

Request Syntax

```
multipart_upload_iterator = vault.multipart_uploads.all()
```

Return type `list(glacier.MultipartUpload)`

Returns A list of MultipartUpload resources

filter (***kwargs*)

Creates an iterable of all MultipartUpload resources in the collection filtered by kwargs passed to method.

Request Syntax

```
multipart_upload_iterator = vault.multipart_uploads.filter(  
    marker='string',  
    limit='string'  
)
```

Parameters

- **marker** (*string*) – An opaque string used for pagination. This value specifies the upload at which the listing of uploads should begin. Get the marker value from a previous List Uploads response. You need only include the marker if you are continuing the pagination of results started in a previous List Uploads request.
- **limit** (*string*) – Specifies the maximum number of uploads returned in the response body. If this value is not specified, the List Uploads operation returns up to 1,000 uploads.

Return type `list(glacier.MultipartUpload)`

Returns A list of MultipartUpload resources

limit (***kwargs*)

Creates an iterable up to a specified amount of MultipartUpload resources in the collection.

Request Syntax

```
multipart_upload_iterator = vault.multipart_uploads.limit(  
    count=123  
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(glacier.MultipartUpload)`

Returns A list of MultipartUpload resources

page_size (**kwargs)

Creates an iterable of all MultipartUpload resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
multipart_upload_iterator = vault.multipart_uploads.page_size(
    count=123
)
```

Parameters *count* (*integer*) – The number of items returned by each service call

Return type list(glacier.MultipartUpload)

Returns A list of MultipartUpload resources

succeeded_jobs**all** ()

Creates an iterable of all Job resources in the collection.

Request Syntax

```
job_iterator = vault.succeeded_jobs.all()
```

Return type list(glacier.Job)

Returns A list of Job resources

filter (**kwargs)

Creates an iterable of all Job resources in the collection filtered by kwargs passed to method.

Request Syntax

```
job_iterator = vault.succeeded_jobs.filter(
    limit='string',
    marker='string',
    completed='string'
)
```

Parameters

- **limit** (*string*) – Specifies that the response be limited to the specified number of items or fewer. If not specified, the List Jobs operation returns up to 1,000 jobs.
- **marker** (*string*) – An opaque string used for pagination. This value specifies the job at which the listing of jobs should begin. Get the marker value from a previous List Jobs response. You need only include the marker if you are continuing the pagination of results started in a previous List Jobs request.
- **completed** (*string*) – Specifies the state of the jobs to return. You can specify `true` or `false`.

Return type list(glacier.Job)

Returns A list of Job resources

limit (**kwargs)

Creates an iterable up to a specified amount of Job resources in the collection.

Request Syntax

```
job_iterator = vault.succeeded_jobs.limit(  
    count=123  
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(glacier.Job)`

Returns A list of Job resources

page_size (***kwargs*)

Creates an iterable of all Job resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
job_iterator = vault.succeeded_jobs.page_size(  
    count=123  
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(glacier.Job)`

Returns A list of Job resources

IAM

Table of Contents

- IAM
 - Client
 - Paginators
 - Waiters
 - Service Resource
 - AccessKey
 - AccessKeyPair
 - AccountPasswordPolicy
 - AccountSummary
 - AssumeRolePolicy
 - CurrentUser
 - Group
 - GroupPolicy
 - InstanceProfile
 - LoginProfile
 - MfaDevice
 - Policy
 - PolicyVersion
 - Role
 - RolePolicy
 - SamlProvider
 - ServerCertificate
 - SigningCertificate
 - User
 - UserPolicy
 - VirtualMfaDevice

Client

`class IAM.Client`

A low-level client representing AWS Identity and Access Management (IAM):

```
import boto3
```

```
client = boto3.client('iam')
```

These are the available methods:

- `add_client_id_to_open_id_connect_provider()`
- `add_role_to_instance_profile()`
- `add_user_to_group()`
- `attach_group_policy()`
- `attach_role_policy()`
- `attach_user_policy()`
- `can_paginate()`
- `change_password()`
- `create_access_key()`
- `create_account_alias()`
- `create_group()`
- `create_instance_profile()`
- `create_login_profile()`
- `create_open_id_connect_provider()`
- `create_policy()`
- `create_policy_version()`
- `create_role()`
- `create_saml_provider()`
- `create_user()`
- `create_virtual_mfa_device()`
- `deactivate_mfa_device()`
- `delete_access_key()`
- `delete_account_alias()`
- `delete_account_password_policy()`
- `delete_group()`
- `delete_group_policy()`
- `delete_instance_profile()`
- `delete_login_profile()`
- `delete_open_id_connect_provider()`
- `delete_policy()`
- `delete_policy_version()`
- `delete_role()`
- `delete_role_policy()`
- `delete_saml_provider()`
- `delete_server_certificate()`
- `delete_signing_certificate()`
- `delete_ssh_public_key()`
- `delete_user()`
- `delete_user_policy()`
- `delete_virtual_mfa_device()`
- `detach_group_policy()`
- `detach_role_policy()`
- `detach_user_policy()`
- `enable_mfa_device()`

- generate_credential_report()
- generate_presigned_url()
- get_access_key_last_used()
- get_account_authorization_details()
- get_account_password_policy()
- get_account_summary()
- get_context_keys_for_custom_policy()
- get_context_keys_for_principal_policy()
- get_credential_report()
- get_group()
- get_group_policy()
- get_instance_profile()
- get_login_profile()
- get_open_id_connect_provider()
- get_paginator()
- get_policy()
- get_policy_version()
- get_role()
- get_role_policy()
- get_saml_provider()
- get_server_certificate()
- get_ssh_public_key()
- get_user()
- get_user_policy()
- get_waiter()
- list_access_keys()
- list_account_aliases()
- list_attached_group_policies()
- list_attached_role_policies()
- list_attached_user_policies()
- list_entities_for_policy()
- list_group_policies()
- list_groups()
- list_groups_for_user()
- list_instance_profiles()
- list_instance_profiles_for_role()
- list_mfa_devices()
- list_open_id_connect_providers()
- list_policies()
- list_policy_versions()
- list_role_policies()
- list_roles()
- list_saml_providers()
- list_server_certificates()
- list_signing_certificates()
- list_ssh_public_keys()
- list_user_policies()
- list_users()
- list_virtual_mfa_devices()
- put_group_policy()
- put_role_policy()
- put_user_policy()
- remove_client_id_from_open_id_connect_provider()
- remove_role_from_instance_profile()

- `remove_user_from_group()`
- `resync_mfa_device()`
- `set_default_policy_version()`
- `simulate_custom_policy()`
- `simulate_principal_policy()`
- `update_access_key()`
- `update_account_password_policy()`
- `update_assume_role_policy()`
- `update_group()`
- `update_login_profile()`
- `update_open_id_connect_provider_thumbprint()`
- `update_saml_provider()`
- `update_server_certificate()`
- `update_signing_certificate()`
- `update_ssh_public_key()`
- `update_user()`
- `upload_server_certificate()`
- `upload_signing_certificate()`
- `upload_ssh_public_key()`

`add_client_id_to_open_id_connect_provider(**kwargs)`
 Adds a new client ID (also known as audience) to the list of client IDs already registered for the specified IAM OpenID Connect provider.

This action is idempotent; it does not fail or return an error if you add an existing client ID to the provider.

Request Syntax

```
response = client.add_client_id_to_open_id_connect_provider(
    OpenIDConnectProviderArn='string',
    ClientID='string'
)
```

Parameters

- **`OpenIDConnectProviderArn` (*string*) – [REQUIRED]**
 The Amazon Resource Name (ARN) of the IAM OpenID Connect (OIDC) provider to add the client ID to. You can get a list of OIDC provider ARNs by using the `ListOpenIDConnectProviders` action.
- **`ClientID` (*string*) – [REQUIRED]**
 The client ID (also known as audience) to add to the IAM OpenID Connect provider.

Returns None

`add_role_to_instance_profile(**kwargs)`
 Adds the specified role to the specified instance profile. For more information about roles, go to [Working with Roles](#) . For more information about instance profiles, go to [About Instance Profiles](#) .

Request Syntax

```
response = client.add_role_to_instance_profile(
    InstanceProfileName='string',
    RoleName='string'
)
```

Parameters

- **InstanceProfileName** (*string*) – [REQUIRED]

The name of the instance profile to update.

- **RoleName** (*string*) – [REQUIRED]

The name of the role to add.

Returns None

add_user_to_group (***kwargs*)

Adds the specified user to the specified group.

Request Syntax

```
response = client.add_user_to_group(
    GroupName='string',
    UserName='string'
)
```

Parameters

- **GroupName** (*string*) – [REQUIRED]

The name of the group to update.

- **UserName** (*string*) – [REQUIRED]

The name of the user to add.

Returns None

attach_group_policy (***kwargs*)

Attaches the specified managed policy to the specified group.

You use this API to attach a managed policy to a group. To embed an inline policy in a group, use `PutGroupPolicy`.

For more information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = client.attach_group_policy(
    GroupName='string',
    PolicyArn='string'
)
```

Parameters

- **GroupName** (*string*) – [REQUIRED]

The name (friendly name, not ARN) of the group to attach the policy to.

- **PolicyArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

Returns None

attach_role_policy (***kwargs*)

Attaches the specified managed policy to the specified role.

When you attach a managed policy to a role, the managed policy is used as the role's access (permissions) policy. You cannot use a managed policy as the role's trust policy. The role's trust policy is created at the

same time as the role, using `CreateRole` . You can update a role's trust policy using `UpdateAssumeRolePolicy` .

Use this API to attach a managed policy to a role. To embed an inline policy in a role, use `PutRolePolicy` . For more information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide* .

Request Syntax

```
response = client.attach_role_policy(
    RoleName='string',
    PolicyArn='string'
)
```

Parameters

- **RoleName** (*string*) – [REQUIRED]

The name (friendly name, not ARN) of the role to attach the policy to.

- **PolicyArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference* .

Returns None

attach_user_policy (***kwargs*)

Attaches the specified managed policy to the specified user.

You use this API to attach a managed policy to a user. To embed an inline policy in a user, use `PutUserPolicy` .

For more information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide* .

Request Syntax

```
response = client.attach_user_policy(
    UserName='string',
    PolicyArn='string'
)
```

Parameters

- **UserName** (*string*) – [REQUIRED]

The name (friendly name, not ARN) of the user to attach the policy to.

- **PolicyArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference* .

Returns None

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`,

if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

change_password (**kwargs)

Changes the password of the IAM user who is calling this action. The root account password is not affected by this action.

To change the password for a different user, see `UpdateLoginProfile` . For more information about modifying passwords, see [Managing Passwords](#) in the *IAM User Guide* .

Request Syntax

```
response = client.change_password(
    OldPassword='string',
    NewPassword='string'
)
```

Parameters

- **OldPassword** (*string*) – [REQUIRED]

The IAM user's current password.

- **NewPassword** (*string*) – [REQUIRED]

The new password. The new password must conform to the AWS account's password policy, if one exists.

Returns None

create_access_key (**kwargs)

Creates a new AWS secret access key and corresponding AWS access key ID for the specified user. The default status for new keys is `Active` .

If you do not specify a user name, IAM determines the user name implicitly based on the AWS access key ID signing the request. Because this action works for access keys under the AWS account, you can use this action to manage root credentials even if the AWS account has no associated users.

For information about limits on the number of keys you can create, see [Limitations on IAM Entities](#) in the *IAM User Guide* .

Warning: To ensure the security of your AWS account, the secret access key is accessible only during key and user creation. You must save the key (for example, in a text file) if you want to be able to access it again. If a secret key is lost, you can delete the access keys for the associated user and then create new keys.

Request Syntax

```
response = client.create_access_key(
    UserName='string'
)
```

Parameters **UserName** (*string*) – The user name that the new key will belong to.

Return type dict

Returns

Response Syntax

```
{
    'AccessKey': {
        'UserName': 'string',
```

```

        'AccessKeyId': 'string',
        'Status': 'Active'|'Inactive',
        'SecretAccessKey': 'string',
        'CreateDate': datetime(2015, 1, 1)
    }
}

```

Response Structure

- (*dict*) –

Contains the response to a successful `CreateAccessKey` request.

- **AccessKey** (*dict*) –

Information about the access key.

- * **UserName** (*string*) –

The name of the IAM user that the access key is associated with.

- * **AccessKeyId** (*string*) –

The ID for this access key.

- * **Status** (*string*) –

The status of the access key. `Active` means the key is valid for API calls, while `Inactive` means it is not.

- * **SecretAccessKey** (*string*) –

The secret key used to sign requests.

- * **CreateDate** (*datetime*) –

The date when the access key was created.

create_account_alias (***kwargs*)

Creates an alias for your AWS account. For information about using an AWS account alias, see [Using an Alias for Your AWS Account ID in the IAM User Guide](#).

Request Syntax

```

response = client.create_account_alias(
    AccountAlias='string'
)

```

Parameters **AccountAlias** (*string*) – [REQUIRED]

The account alias to create.

Returns `None`

create_group (***kwargs*)

Creates a new group.

For information about the number of groups you can create, see [Limitations on IAM Entities in the IAM User Guide](#).

Request Syntax

```

response = client.create_group(
    Path='string',
    GroupName='string'
)

```

Parameters

- **Path** (*string*) – The path to the group. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

This parameter is optional. If it is not included, it defaults to a slash (/).

- **GroupName** (*string*) – **[REQUIRED]**

The name of the group to create. Do not include the path in this value.

Return type dict

Returns

Response Syntax

```
{
  'Group': {
    'Path': 'string',
    'GroupName': 'string',
    'GroupId': 'string',
    'Arn': 'string',
    'CreateDate': datetime(2015, 1, 1)
  }
}
```

Response Structure

- (*dict*) –

Contains the response to a successful CreateGroup request.

– **Group** (*dict*) –

Information about the group.

- * **Path** (*string*) –

The path to the group. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **GroupName** (*string*) –

The friendly name that identifies the group.

- * **GroupId** (*string*) –

The stable and unique string identifying the group. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **Arn** (*string*) –

The Amazon Resource Name (ARN) specifying the group. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#) , when the group was created.

create_instance_profile (**kwargs)

Creates a new instance profile. For information about instance profiles, go to [About Instance Profiles](#) .

For information about the number of instance profiles you can create, see [Limitations on IAM Entities](#) in the *IAM User Guide* .

Request Syntax

```
response = client.create_instance_profile(
    InstanceProfileName='string',
```



```
Path='string'
)
```

Parameters

- **InstanceProfileName** (*string*) – **[REQUIRED]**

The name of the instance profile to create.

- **Path** (*string*) – The path to the instance profile. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

This parameter is optional. If it is not included, it defaults to a slash (/).

Return type dict

Returns

Response Syntax

```
{
  'InstanceProfile': {
    'Path': 'string',
    'InstanceProfileName': 'string',
    'InstanceProfileId': 'string',
    'Arn': 'string',
    'CreateDate': datetime(2015, 1, 1),
    'Roles': [
      {
        'Path': 'string',
        'RoleName': 'string',
        'RoleId': 'string',
        'Arn': 'string',
        'CreateDate': datetime(2015, 1, 1),
        'AssumeRolePolicyDocument': 'string'
      },
    ]
  }
}
```

Response Structure

- (*dict*) –

Contains the response to a successful CreateInstanceProfile request.

- **InstanceProfile** (*dict*) –

Information about the instance profile.

- * **Path** (*string*) –

The path to the instance profile. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **InstanceProfileName** (*string*) –

The name identifying the instance profile.

- * **InstanceProfileId** (*string*) –

The stable and unique string identifying the instance profile. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **Arn** (*string*) –

The Amazon Resource Name (ARN) specifying the instance profile. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **CreateDate** (*datetime*) –

The date when the instance profile was created.

- * **Roles** (*list*) –

The role associated with the instance profile.

- (*dict*) –

Contains information about an IAM role.

This data type is used as a response element in the following actions:

- CreateRole
 - GetRole
 - ListRoles
 - **Path** (*string*) –

The path to the role. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **RoleName** (*string*) –

The friendly name that identifies the role.

- **RoleId** (*string*) –

The stable and unique string identifying the role. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) specifying the role. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#) , when the role was created.

- **AssumeRolePolicyDocument** (*string*) –

The policy that grants an entity permission to assume the role.

create_login_profile (***kwargs*)

Creates a password for the specified user, giving the user the ability to access AWS services through the AWS Management Console. For more information about managing passwords, see [Managing Passwords](#) in the *Using IAM* guide.

Request Syntax

```
response = client.create_login_profile(
    Username='string',
    Password='string',
    PasswordResetRequired=True|False
)
```

Parameters

- **Username** (*string*) – **[REQUIRED]**

The name of the user to create a password for.

- **Password** (*string*) – **[REQUIRED]**

The new password for the user.

- **PasswordResetRequired** (*boolean*) – Specifies whether the user is required to set a new password on next sign-in.

Return type dict

Returns

Response Syntax

```
{
  'LoginProfile': {
    'UserName': 'string',
    'CreateDate': datetime(2015, 1, 1),
    'PasswordResetRequired': True|False
  }
}
```

Response Structure

- (*dict*) –

Contains the response to a successful CreateLoginProfile request.

- **LoginProfile** (*dict*) –

The user name and password create date.

- * **UserName** (*string*) –

The name of the user, which can be used for signing in to the AWS Management Console.

- * **CreateDate** (*datetime*) –

The date when the password for the user was created.

- * **PasswordResetRequired** (*boolean*) –

Specifies whether the user is required to set a new password on next sign-in.

create_open_id_connect_provider (***kwargs*)

Creates an IAM entity to describe an identity provider (IdP) that supports [OpenID Connect \(OIDC\)](#).

The OIDC provider that you create with this operation can be used as a principal in a role's trust policy to establish a trust relationship between AWS and the OIDC provider.

When you create the IAM OIDC provider, you specify the URL of the OIDC identity provider (IdP) to trust, a list of client IDs (also known as audiences) that identify the application or applications that are allowed to authenticate using the OIDC provider, and a list of thumbprints of the server certificate(s) that the IdP uses. You get all of this information from the OIDC IdP that you want to use for access to AWS.

Note: Because trust for the OIDC provider is ultimately derived from the IAM provider that this action creates, it is a best practice to limit access to the CreateOpenIDConnectProvider action to highly-privileged users.

Request Syntax

```
response = client.create_open_id_connect_provider(
    Url='string',
    ClientIDList=[
        'string',
    ],
    ThumbprintList=[
        'string',
    ],
)
```

```
]
)
```

Parameters

- **Url** (*string*) – [REQUIRED]

The URL of the identity provider. The URL must begin with “https://” and should correspond to the `iss` claim in the provider’s OpenID Connect ID tokens. Per the OIDC standard, path components are allowed but query parameters are not. Typically the URL consists of only a host name, like “https://server.example.org” or “https://example.com”.

You cannot register the same provider multiple times in a single AWS account. If you try to submit a URL that has already been used for an OpenID Connect provider in the AWS account, you will get an error.

- **ClientIDList** (*list*) – A list of client IDs (also known as audiences). When a mobile or web app registers with an OpenID Connect provider, they establish a value that identifies the application. (This is the value that’s sent as the `client_id` parameter on OAuth requests.)

You can register multiple client IDs with the same provider. For example, you might have multiple applications that use the same OIDC provider. You cannot register more than 100 client IDs with a single IAM OIDC provider.

There is no defined format for a client ID. The `CreateOpenIDConnectProviderRequest` action accepts client IDs up to 255 characters long.

– (*string*) –

- **ThumbprintList** (*list*) – [REQUIRED]

A list of server certificate thumbprints for the OpenID Connect (OIDC) identity provider’s server certificate(s). Typically this list includes only one entry. However, IAM lets you have up to five thumbprints for an OIDC provider. This lets you maintain multiple thumbprints if the identity provider is rotating certificates.

The server certificate thumbprint is the hex-encoded SHA-1 hash value of the X.509 certificate used by the domain where the OpenID Connect provider makes its keys available. It is always a 40-character string.

You must provide at least one thumbprint when creating an IAM OIDC provider. For example, if the OIDC provider is `server.example.com` and the provider stores its keys at “https://keys.server.example.com/openid-connect”, the thumbprint string would be the hex-encoded SHA-1 hash value of the certificate used by `https://keys.server.example.com`.

For more information about obtaining the OIDC provider’s thumbprint, see [Obtaining the Thumbprint for an OpenID Connect Provider](#) in the *IAM User Guide*.

– (*string*) –

Contains a thumbprint for an identity provider’s server certificate.

The identity provider’s server certificate thumbprint is the hex-encoded SHA-1 hash value of the self-signed X.509 certificate used by the domain where the OpenID Connect provider makes its keys available. It is always a 40-character string.

Return type dict

Returns

Response Syntax

```
{
    'OpenIDConnectProviderArn': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful CreateOpenIDConnectProvider request.

- **OpenIDConnectProviderArn** (*string*) –

The Amazon Resource Name (ARN) of the IAM OpenID Connect provider that was created. For more information, see [OpenIDConnect-ProviderListEntry](#).

create_policy (***kwargs*)

Creates a new managed policy for your AWS account.

This operation creates a policy version with a version identifier of v1 and sets v1 as the policy’s default version. For more information about policy versions, see [Versioning for Managed Policies](#) in the *IAM User Guide*.

For more information about managed policies in general, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = client.create_policy(
    PolicyName='string',
    Path='string',
    PolicyDocument='string',
    Description='string'
)
```

Parameters

- **PolicyName** (*string*) – **[REQUIRED]**

The name of the policy document.

- **Path** (*string*) – The path for the policy.

For more information about paths, see [IAM Identifiers](#) in the *IAM User Guide*.

This parameter is optional. If it is not included, it defaults to a slash (/).

- **PolicyDocument** (*string*) – **[REQUIRED]**

The policy document.

- **Description** (*string*) – A friendly description of the policy.

Typically used to store information about the permissions defined in the policy. For example, “Grants access to production DynamoDB tables.”

The policy description is immutable. After a value is assigned, it cannot be changed.

Return type dict

Returns**Response Syntax**

```
{
  'Policy': {
    'PolicyName': 'string',
    'PolicyId': 'string',
    'Arn': 'string',
    'Path': 'string',
    'DefaultVersionId': 'string',
    'AttachmentCount': 123,
    'IsAttachable': True|False,
    'Description': 'string',
    'CreateDate': datetime(2015, 1, 1),
    'UpdateDate': datetime(2015, 1, 1)
  }
}
```

Response Structure

- (*dict*) –

Contains the response to a successful CreatePolicy request.

- **Policy** (*dict*) –

Information about the policy.

- * **PolicyName** (*string*) –

The friendly name (not ARN) identifying the policy.

- * **PolicyId** (*string*) –

The stable and unique string identifying the policy.

For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **Arn** (*string*) –

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

- * **Path** (*string*) –

The path to the policy.

For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **DefaultVersionId** (*string*) –

The identifier for the version of the policy that is set as the default version.

- * **AttachmentCount** (*integer*) –

The number of entities (users, groups, and roles) that the policy is attached to.

- * **IsAttachable** (*boolean*) –

Specifies whether the policy can be attached to an IAM user, group, or role.

- * **Description** (*string*) –

A friendly description of the policy.

This element is included in the response to the `GetPolicy` operation. It is not included in the response to the `ListPolicies` operation.

* **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#) , when the policy was created.

* **UpdateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#) , when the policy was last updated.

When a policy has only one version, this field contains the date and time when the policy was created. When a policy has more than one version, this field contains the date and time when the most recent policy version was created.

create_policy_version (***kwargs*)

Creates a new version of the specified managed policy. To update a managed policy, you create a new policy version. A managed policy can have up to five versions. If the policy has five versions, you must delete an existing version using `DeletePolicyVersion` before you create a new version.

Optionally, you can set the new version as the policy's default version. The default version is the operative version; that is, the version that is in effect for the IAM users, groups, and roles that the policy is attached to.

For more information about managed policy versions, see [Versioning for Managed Policies](#) in the *IAM User Guide* .

Request Syntax

```
response = client.create_policy_version(
    PolicyArn='string',
    PolicyDocument='string',
    SetAsDefault=True|False
)
```

Parameters

• **PolicyArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference* .

• **PolicyDocument** (*string*) – [REQUIRED]

The policy document.

• **SetAsDefault** (*boolean*) – Specifies whether to set this version as the policy's default version.

When this parameter is `true` , the new policy version becomes the operative version; that is, the version that is in effect for the IAM users, groups, and roles that the policy is attached to.

For more information about managed policy versions, see [Versioning for Managed Policies](#) in the *IAM User Guide* .

Return type dict

Returns

Response Syntax

```
{
  'PolicyVersion': {
    'Document': 'string',
    'VersionId': 'string',
    'IsDefaultVersion': True|False,
    'CreateDate': datetime(2015, 1, 1)
  }
}
```

Response Structure

- (*dict*) –

Contains the response to a successful CreatePolicyVersion request.

- **PolicyVersion** (*dict*) –

Information about the policy version.

- * **Document** (*string*) –

The policy document.

The policy document is returned in the response to the GetPolicyVersion and GetAccountAuthorizationDetails operations. It is not returned in the response to the CreatePolicyVersion or ListPolicyVersions operations.

- * **VersionId** (*string*) –

The identifier for the policy version.

Policy version identifiers always begin with `v` (always lowercase). When a policy is created, the first policy version is `v1`.

- * **IsDefaultVersion** (*boolean*) –

Specifies whether the policy version is set as the policy's default version.

- * **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the policy version was created.

create_role (**kwargs)

Creates a new role for your AWS account. For more information about roles, go to [Working with Roles](#). For information about limitations on role names and the number of roles you can create, go to [Limitations on IAM Entities](#) in the *IAM User Guide*.

The policy in the following example grants permission to an EC2 instance to assume the role.

Request Syntax

```
response = client.create_role(
    Path='string',
    RoleName='string',
    AssumeRolePolicyDocument='string'
)
```

Parameters

- **Path** (*string*) – The path to the role. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

This parameter is optional. If it is not included, it defaults to a slash (/).

- **RoleName** (*string*) – [REQUIRED]
The name of the role to create.
- **AssumeRolePolicyDocument** (*string*) – [REQUIRED]
The policy that grants an entity permission to assume the role.

Return type dict

Returns

Response Syntax

```
{
  'Role': {
    'Path': 'string',
    'RoleName': 'string',
    'RoleId': 'string',
    'Arn': 'string',
    'CreateDate': datetime(2015, 1, 1),
    'AssumeRolePolicyDocument': 'string'
  }
}
```

Response Structure

- (*dict*) –

Contains the response to a successful CreateRole request.

– **Role** (*dict*) –

Information about the role.

* **Path** (*string*) –

The path to the role. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

* **RoleName** (*string*) –

The friendly name that identifies the role.

* **RoleId** (*string*) –

The stable and unique string identifying the role. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

* **Arn** (*string*) –

The Amazon Resource Name (ARN) specifying the role. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

* **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the role was created.

* **AssumeRolePolicyDocument** (*string*) –

The policy that grants an entity permission to assume the role.

create_saml_provider (***kwargs*)

Creates an IAM entity to describe an identity provider (IdP) that supports SAML 2.0.

The SAML provider that you create with this operation can be used as a principal in a role's trust policy to establish a trust relationship between AWS and a SAML identity provider. You can create an IAM role that supports Web-based single sign-on (SSO) to the AWS Management Console or one that supports API access to AWS.

When you create the SAML provider, you upload an a SAML metadata document that you get from your IdP and that includes the issuer's name, expiration information, and keys that can be used to validate the SAML authentication response (assertions) that are received from the IdP. You must generate the metadata document using the identity management software that is used as your organization's IdP.

Note: This operation requires [Signature Version 4](#) .

For more information, see [Enabling SAML 2.0 Federated Users to Access the AWS Management Console](#) and [About SAML 2.0-based Federation](#) in the *IAM User Guide* .

Request Syntax

```
response = client.create_saml_provider(
    SAMLMetadataDocument='string',
    Name='string'
)
```

Parameters

- **SAMLMetadataDocument** (*string*) – **[REQUIRED]**

An XML document generated by an identity provider (IdP) that supports SAML 2.0. The document includes the issuer's name, expiration information, and keys that can be used to validate the SAML authentication response (assertions) that are received from the IdP. You must generate the metadata document using the identity management software that is used as your organization's IdP.

For more information, see [About SAML 2.0-based Federation](#) in the *IAM User Guide*

- **Name** (*string*) – **[REQUIRED]**

The name of the provider to create.

Return type dict

Returns

Response Syntax

```
{
    'SAMLProviderArn': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful CreateSAMLProvider request.

- **SAMLProviderArn** (*string*) –

The Amazon Resource Name (ARN) of the SAML provider.

create_user (***kwargs*)

Creates a new user for your AWS account.

For information about limitations on the number of users you can create, see [Limitations on IAM Entities](#) in the *IAM User Guide* .

Request Syntax

```
response = client.create_user(
    Path='string',
    UserName='string'
)
```

Parameters

- **Path** (*string*) – The path for the user name. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

This parameter is optional. If it is not included, it defaults to a slash (/).

- **UserName** (*string*) – **[REQUIRED]**

The name of the user to create.

Return type dict

Returns**Response Syntax**

```
{
  'User': {
    'Path': 'string',
    'UserName': 'string',
    'UserId': 'string',
    'Arn': 'string',
    'CreateDate': datetime(2015, 1, 1),
    'PasswordLastUsed': datetime(2015, 1, 1)
  }
}
```

Response Structure

- (*dict*) –

Contains the response to a successful CreateUser request.

- **User** (*dict*) –

Information about the user.

- * **Path** (*string*) –

The path to the user. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **UserName** (*string*) –

The friendly name identifying the user.

- * **UserId** (*string*) –

The stable and unique string identifying the user. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **Arn** (*string*) –

The Amazon Resource Name (ARN) that identifies the user. For more information about ARNs and how to use ARNs in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#) , when the user was created.

- * **PasswordLastUsed** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#) , when the user's password was last used to sign in to an AWS website. For a list of AWS websites that capture a user's last sign-in time, see the [Credential Reports](#) topic in the *Using IAM* guide. If a password is used more than once in a five-minute span, only the first use is returned in this field. This field is null (not present) when:

- The user does not have a password
- The password exists but has never been used (at least not since IAM started tracking this information on October 20th, 2014)
- there is no sign-in data associated with the user

This value is returned only in the GetUser and ListUsers actions.

`create_virtual_mfa_device` (**kwargs)

Creates a new virtual MFA device for the AWS account. After creating the virtual MFA, use `EnableMFADevice` to attach the MFA device to an IAM user. For more information about creating and working with virtual MFA devices, go to [Using a Virtual MFA Device](#) in the *Using IAM* guide.

For information about limits on the number of MFA devices you can create, see [Limitations on Entities](#) in the *Using IAM* guide.

Warning: The seed information contained in the QR code and the Base32 string should be treated like any other secret access information, such as your AWS access keys or your passwords. After you provision your virtual device, you should ensure that the information is destroyed following secure procedures.

Request Syntax

```
response = client.create_virtual_mfa_device(
    Path='string',
    VirtualMFADeviceName='string'
)
```

Parameters

- **Path** (*string*) – The path for the virtual MFA device. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

This parameter is optional. If it is not included, it defaults to a slash (/).

- **VirtualMFADeviceName** (*string*) – **[REQUIRED]**

The name of the virtual MFA device. Use with path to uniquely identify a virtual MFA device.

Return type dict

Returns

Response Syntax

```
{
  'VirtualMFADevice': {
    'SerialNumber': 'string',
    'Base32StringSeed': b'bytes',
    'QRCodePNG': b'bytes',
    'User': {
      'Path': 'string',
      'UserName': 'string',
      'UserId': 'string',
      'Arn': 'string',
      'CreateDate': datetime(2015, 1, 1),
      'PasswordLastUsed': datetime(2015, 1, 1)
    },
    'EnableDate': datetime(2015, 1, 1)
  }
}
```

Response Structure

- (*dict*) –

Contains the response to a successful CreateVirtualMFADevice request.

- **VirtualMFADevice** (*dict*) –

A newly created virtual MFA device.

- * **SerialNumber** (*string*) –

The serial number associated with VirtualMFADevice.

- * **Base32StringSeed** (*bytes*) –

The Base32 seed defined as specified in [RFC3548](#). The Base32StringSeed is Base64-encoded.

- * **QRCodePNG** (*bytes*) –

A QR code PNG image that encodes `otpauth://totp/${virtualMFADeviceName}@${AccountName}?secret=${Base32StringSeed}` where `virtualMFADeviceName` is one of the create call arguments, `AccountName` is the user name if set (otherwise, the account ID otherwise), and `Base32String` is the seed in Base32 format. The `Base32String` value is Base64-encoded.

- * **User** (*dict*) –

Contains information about an IAM user entity.

This data type is used as a response element in the following actions:

- CreateUser
- GetUser
- ListUsers

- **Path** (*string*) –

The path to the user. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **UserName** (*string*) –

The friendly name identifying the user.

- **UserId** (*string*) –

The stable and unique string identifying the user. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) that identifies the user. For more information about ARNs and how to use ARNs in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the user was created.

- **PasswordLastUsed** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the user's password was last used to sign in to an AWS website. For a list of AWS websites that capture a user's last sign-in time, see the [Credential Reports](#) topic in the *Using IAM* guide. If a password is used more than once in a five-minute span, only the first use is returned in this field. This field is null (not present) when:

- The user does not have a password

- The password exists but has never been used (at least not since IAM started tracking this information on October 20th, 2014)
- there is no sign-in data associated with the user

This value is returned only in the GetUser and ListUsers actions.

* **EnableDate** (*datetime*) –

The date and time on which the virtual MFA device was enabled.

deactivate_mfa_device (***kwargs*)

Deactivates the specified MFA device and removes it from association with the user name for which it was originally enabled.

For more information about creating and working with virtual MFA devices, go to [Using a Virtual MFA Device](#) in the *Using IAM* guide.

Request Syntax

```
response = client.deactivate_mfa_device(
    UserName='string',
    SerialNumber='string'
)
```

Parameters

- **UserName** (*string*) – [REQUIRED]

The name of the user whose MFA device you want to deactivate.

- **SerialNumber** (*string*) – [REQUIRED]

The serial number that uniquely identifies the MFA device. For virtual MFA devices, the serial number is the device ARN.

Returns None

delete_access_key (***kwargs*)

Deletes the access key associated with the specified user.

If you do not specify a user name, IAM determines the user name implicitly based on the AWS access key ID signing the request. Because this action works for access keys under the AWS account, you can use this action to manage root credentials even if the AWS account has no associated users.

Request Syntax

```
response = client.delete_access_key(
    UserName='string',
    AccessKeyId='string'
)
```

Parameters

- **UserName** (*string*) – The name of the user whose key you want to delete.
- **AccessKeyId** (*string*) – [REQUIRED]

The access key ID for the access key ID and secret access key you want to delete.

Returns None

delete_account_alias (***kwargs*)

Deletes the specified AWS account alias. For information about using an AWS account alias, see [Using an Alias for Your AWS Account ID](#) in the *IAM User Guide*.

Request Syntax

```
response = client.delete_account_alias(
    AccountAlias='string'
)
```

Parameters **AccountAlias** (*string*) – [REQUIRED]

The name of the account alias to delete.

Returns None

delete_account_password_policy()

Deletes the password policy for the AWS account.

Request Syntax

```
response = client.delete_account_password_policy()
```

Returns None

delete_group(kwargs)**

Deletes the specified group. The group must not contain any users or have any attached policies.

Request Syntax

```
response = client.delete_group(
    GroupName='string'
)
```

Parameters **GroupName** (*string*) – [REQUIRED]

The name of the group to delete.

Returns None

delete_group_policy(kwargs)**

Deletes the specified inline policy that is embedded in the specified group.

A group can also have managed policies attached to it. To detach a managed policy from a group, use `DetachGroupPolicy`. For more information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = client.delete_group_policy(
    GroupName='string',
    PolicyName='string'
)
```

Parameters

- **GroupName** (*string*) – [REQUIRED]

The name (friendly name, not ARN) identifying the group that the policy is embedded in.

- **PolicyName** (*string*) – [REQUIRED]

The name identifying the policy document to delete.

Returns None

delete_instance_profile(kwargs)**

Deletes the specified instance profile. The instance profile must not have an associated role.

Warning: Make sure you do not have any Amazon EC2 instances running with the instance profile you are about to delete. Deleting a role or instance profile that is associated with a running instance will break any applications running on the instance.

For more information about instance profiles, go to [About Instance Profiles](#) .

Request Syntax

```
response = client.delete_instance_profile(
    InstanceProfileName='string'
)
```

Parameters `InstanceProfileName` (*string*) – [REQUIRED]

The name of the instance profile to delete.

Returns None

delete_login_profile (***kwargs*)

Deletes the password for the specified user, which terminates the user’s ability to access AWS services through the AWS Management Console.

Warning: Deleting a user’s password does not prevent a user from accessing IAM through the command line interface or the API. To prevent all user access you must also either make the access key inactive or delete it. For more information about making keys inactive or deleting them, see [UpdateAccessKey](#) and [DeleteAccessKey](#) .

Request Syntax

```
response = client.delete_login_profile(
    UserName='string'
)
```

Parameters `UserName` (*string*) – [REQUIRED]

The name of the user whose password you want to delete.

Returns None

delete_open_id_connect_provider (***kwargs*)

Deletes an IAM OpenID Connect identity provider.

Deleting an OIDC provider does not update any roles that reference the provider as a principal in their trust policies. Any attempt to assume a role that references a provider that has been deleted will fail.

This action is idempotent; it does not fail or return an error if you call the action for a provider that was already deleted.

Request Syntax

```
response = client.delete_open_id_connect_provider(
    OpenIDConnectProviderArn='string'
)
```

Parameters `OpenIDConnectProviderArn` (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the IAM OpenID Connect provider to delete. You can get a list of OpenID Connect provider ARNs by using the [ListOpenIDConnectProviders](#) action.

Returns None

delete_policy (**kwargs)

Deletes the specified managed policy.

Before you can delete a managed policy, you must detach the policy from all users, groups, and roles that it is attached to, and you must delete all of the policy's versions. The following steps describe the process for deleting a managed policy:

- Detach the policy from all users, groups, and roles that the policy is attached to, using the `DetachUserPolicy`, `DetachGroupPolicy`, or `DetachRolePolicy` APIs. To list all the users, groups, and roles that a policy is attached to, use `ListEntitiesForPolicy`.
- Delete all versions of the policy using `DeletePolicyVersion`. To list the policy's versions, use `ListPolicyVersions`. You cannot use `DeletePolicyVersion` to delete the version that is marked as the default version. You delete the policy's default version in the next step of the process.
- Delete the policy (this automatically deletes the policy's default version) using this API.

For information about managed policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = client.delete_policy(
    PolicyArn='string'
)
```

Parameters `PolicyArn` (*string*) – [REQUIRED]

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

Returns None

delete_policy_version (**kwargs)

Deletes the specified version of the specified managed policy.

You cannot delete the default version of a policy using this API. To delete the default version of a policy, use `DeletePolicy`. To find out which version of a policy is marked as the default version, use `ListPolicyVersions`.

For information about versions for managed policies, refer to [Versioning for Managed Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = client.delete_policy_version(
    PolicyArn='string',
    VersionId='string'
)
```

Parameters

- **PolicyArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

- **VersionId** (*string*) – [REQUIRED]

The policy version to delete.

For more information about managed policy versions, see [Versioning for Managed Policies](#) in the *IAM User Guide*.

Returns None

delete_role (**kwargs)

Deletes the specified role. The role must not have any policies attached. For more information about roles, go to [Working with Roles](#).

Warning: Make sure you do not have any Amazon EC2 instances running with the role you are about to delete. Deleting a role or instance profile that is associated with a running instance will break any applications running on the instance.

Request Syntax

```
response = client.delete_role(  
    RoleName='string'  
)
```

Parameters **RoleName** (*string*) – [REQUIRED]

The name of the role to delete.

Returns None

delete_role_policy (**kwargs)

Deletes the specified inline policy that is embedded in the specified role.

A role can also have managed policies attached to it. To detach a managed policy from a role, use `DetachRolePolicy`. For more information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = client.delete_role_policy(  
    RoleName='string',  
    PolicyName='string'  
)
```

Parameters

- **RoleName** (*string*) – [REQUIRED]

The name (friendly name, not ARN) identifying the role that the policy is embedded in.

- **PolicyName** (*string*) – [REQUIRED]

The name identifying the policy document to delete.

Returns None

delete_saml_provider (**kwargs)

Deletes a SAML provider.

Deleting the provider does not update any roles that reference the SAML provider as a principal in their trust policies. Any attempt to assume a role that references a SAML provider that has been deleted will fail.

Note: This operation requires [Signature Version 4](#).

Request Syntax

```
response = client.delete_saml_provider(  
    SAMLProviderArn='string'  
)
```

Parameters `SAMLProviderArn` (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the SAML provider to delete.

Returns None

delete_server_certificate (***kwargs*)

Deletes the specified server certificate.

Warning: If you are using a server certificate with Elastic Load Balancing, deleting the certificate could have implications for your application. If Elastic Load Balancing doesn't detect the deletion of bound certificates, it may continue to use the certificates. This could cause Elastic Load Balancing to stop accepting traffic. We recommend that you remove the reference to the certificate from Elastic Load Balancing before using this command to delete the certificate. For more information, go to [DeleteLoadBalancerListeners](#) in the *Elastic Load Balancing API Reference* .

Request Syntax

```
response = client.delete_server_certificate(
    ServerCertificateName='string'
)
```

Parameters `ServerCertificateName` (*string*) – [REQUIRED]

The name of the server certificate you want to delete.

Returns None

delete_signing_certificate (***kwargs*)

Deletes the specified signing certificate associated with the specified user.

If you do not specify a user name, IAM determines the user name implicitly based on the AWS access key ID signing the request. Because this action works for access keys under the AWS account, you can use this action to manage root credentials even if the AWS account has no associated users.

Request Syntax

```
response = client.delete_signing_certificate(
    UserName='string',
    CertificateId='string'
)
```

Parameters

- **UserName** (*string*) – The name of the user the signing certificate belongs to.
- **CertificateId** (*string*) – [REQUIRED]

The ID of the signing certificate to delete.

Returns None

delete_ssh_public_key (***kwargs*)

Deletes the specified SSH public key.

The SSH public key deleted by this action is used only for authenticating the associated IAM user to an AWS CodeCommit repository. For more information about using SSH keys to authenticate to an AWS CodeCommit repository, see [Set up AWS CodeCommit for SSH Connections](#) in the *AWS CodeCommit User Guide* .

Request Syntax

```
response = client.delete_ssh_public_key(
    UserName='string',
    SSHPublicKeyId='string'
)
```

Parameters

- **UserName** (*string*) – [REQUIRED]
The name of the IAM user associated with the SSH public key.
- **SSHPublicKeyId** (*string*) – [REQUIRED]
The unique identifier for the SSH public key.

Returns None**delete_user** (***kwargs*)

Deletes the specified user. The user must not belong to any groups, have any keys or signing certificates, or have any attached policies.

Request Syntax

```
response = client.delete_user(
    UserName='string'
)
```

Parameters **UserName** (*string*) – [REQUIRED]

The name of the user to delete.

Returns None**delete_user_policy** (***kwargs*)

Deletes the specified inline policy that is embedded in the specified user.

A user can also have managed policies attached to it. To detach a managed policy from a user, use `DetachUserPolicy`. For more information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = client.delete_user_policy(
    UserName='string',
    PolicyName='string'
)
```

Parameters

- **UserName** (*string*) – [REQUIRED]
The name (friendly name, not ARN) identifying the user that the policy is embedded in.
- **PolicyName** (*string*) – [REQUIRED]
The name identifying the policy document to delete.

Returns None**delete_virtual_mfa_device** (***kwargs*)

Deletes a virtual MFA device.

Note: You must deactivate a user’s virtual MFA device before you can delete it. For information about deactivating MFA devices, see `DeactivateMFADevice`.

Request Syntax

```
response = client.delete_virtual_mfa_device(
    SerialNumber='string'
)
```

Parameters `SerialNumber` (*string*) – [REQUIRED]

The serial number that uniquely identifies the MFA device. For virtual MFA devices, the serial number is the same as the ARN.

Returns None

detach_group_policy (***kwargs*)

Removes the specified managed policy from the specified group.

A group can also have inline policies embedded with it. To delete an inline policy, use the `DeleteGroupPolicy` API. For information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = client.detach_group_policy(
    GroupName='string',
    PolicyArn='string'
)
```

Parameters

- **GroupName** (*string*) – [REQUIRED]

The name (friendly name, not ARN) of the group to detach the policy from.

- **PolicyArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

Returns None

detach_role_policy (***kwargs*)

Removes the specified managed policy from the specified role.

A role can also have inline policies embedded with it. To delete an inline policy, use the `DeleteRolePolicy` API. For information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = client.detach_role_policy(
    RoleName='string',
    PolicyArn='string'
)
```

Parameters

- **RoleName** (*string*) – [REQUIRED]

The name (friendly name, not ARN) of the role to detach the policy from.

- **PolicyArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\)](#) and [AWS Service Namespaces](#) in the *AWS General Reference*.

Returns None

detach_user_policy (**kwargs)

Removes the specified managed policy from the specified user.

A user can also have inline policies embedded with it. To delete an inline policy, use the `DeleteUserPolicy` API. For information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = client.detach_user_policy(
    UserName='string',
    PolicyArn='string'
)
```

Parameters

- **UserName** (*string*) – [REQUIRED]
The name (friendly name, not ARN) of the user to detach the policy from.
- **PolicyArn** (*string*) – [REQUIRED]
The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\)](#) and [AWS Service Namespaces](#) in the *AWS General Reference*.

Returns None

enable_mfa_device (**kwargs)

Enables the specified MFA device and associates it with the specified user name. When enabled, the MFA device is required for every subsequent login by the user name associated with the device.

Request Syntax

```
response = client.enable_mfa_device(
    UserName='string',
    SerialNumber='string',
    AuthenticationCode1='string',
    AuthenticationCode2='string'
)
```

Parameters

- **UserName** (*string*) – [REQUIRED]
The name of the user for whom you want to enable the MFA device.
- **SerialNumber** (*string*) – [REQUIRED]
The serial number that uniquely identifies the MFA device. For virtual MFA devices, the serial number is the device ARN.
- **AuthenticationCode1** (*string*) – [REQUIRED]
An authentication code emitted by the device.
- **AuthenticationCode2** (*string*) – [REQUIRED]
A subsequent authentication code emitted by the device.

Returns None

generate_credential_report()

Generates a credential report for the AWS account. For more information about the credential report, see [Getting Credential Reports](#) in the *IAM User Guide*.

Request Syntax

```
response = client.generate_credential_report()
```

Return type dict

Returns**Response Syntax**

```
{
    'State': 'STARTED'|'INPROGRESS'|'COMPLETE',
    'Description': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful GenerateCredentialReport request.

- **State** (*string*) –

Information about the state of the credential report.

- **Description** (*string*) –

Information about the credential report.

generate_presigned_url (ClientMethod, Params=None, ExpiresIn=3600, HttpMethod=None)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to *ClientMethod*.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_access_key_last_used (kwargs)**

Retrieves information about when the specified access key was last used. The information includes the date and time of last use, along with the AWS service and region that were specified in the last request made with that key.

Request Syntax

```
response = client.get_access_key_last_used(
    AccessKeyId='string'
)
```

Parameters **AccessKeyId** (*string*) – **[REQUIRED]**

The identifier of an access key.

Return type dict

Returns**Response Syntax**

```
{
    'UserName': 'string',
    'AccessKeyLastUsed': {
        'LastUsedDate': datetime(2015, 1, 1),
        'ServiceName': 'string',
        'Region': 'string'
    }
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `GetAccessKeyLastUsed` request. It is also returned as a member of the `AccessKeyMetaData` structure returned by the `ListAccessKeys` action.

- **UserName** (*string*) –

The name of the AWS IAM user that owns this access key.

- **AccessKeyLastUsed** (*dict*) –

Contains information about the last time the access key was used.

- * **LastUsedDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the access key was most recently used. This field is null when:

- The user does not have an access key.
- An access key exists but has never been used, at least not since IAM started tracking this information on April 22nd, 2015.
- There is no sign-in data associated with the user

- * **ServiceName** (*string*) –

The name of the AWS service with which this access key was most recently used. This field is null when:

- The user does not have an access key.
- An access key exists but has never been used, at least not since IAM started tracking this information on April 22nd, 2015.
- There is no sign-in data associated with the user

- * **Region** (*string*) –

The AWS region where this access key was most recently used. This field is null when:

- The user does not have an access key.
- An access key exists but has never been used, at least not since IAM started tracking this information on April 22nd, 2015.
- There is no sign-in data associated with the user

For more information about AWS regions, see [Regions and Endpoints](#) in the Amazon Web Services General Reference.

`get_account_authorization_details` (**kwargs)

Retrieves information about all IAM users, groups, roles, and policies in your account, including their relationships to one another. Use this API to obtain a snapshot of the configuration of IAM permissions (users, groups, roles, and policies) in your account.

You can optionally filter the results using the `Filter` parameter. You can paginate the results using the `MaxItems` and `Marker` parameters.

Request Syntax


```

response = client.get_account_authorization_details(
    Filter=[
        'User' | 'Role' | 'Group' | 'LocalManagedPolicy' | 'AWSManagedPolicy',
    ],
    MaxItems=123,
    Marker='string'
)

```

Parameters

- **Filter** (*list*) – A list of entity types (user, group, role, local managed policy, or AWS managed policy) for filtering the results.
– (*string*) –
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.

Return type dict

Returns

Response Syntax

```

{
  'UserDetailList': [
    {
      'Path': 'string',
      'UserName': 'string',
      'UserId': 'string',
      'Arn': 'string',
      'CreateDate': datetime(2015, 1, 1),
      'UserPolicyList': [
        {
          'PolicyName': 'string',
          'PolicyDocument': 'string'
        },
      ],
      'GroupList': [
        'string',
      ],
      'AttachedManagedPolicies': [
        {
          'PolicyName': 'string',
          'PolicyArn': 'string'
        },
      ],
    },
  ],
  'GroupDetailList': [

```

```
{
    'Path': 'string',
    'GroupName': 'string',
    'GroupId': 'string',
    'Arn': 'string',
    'CreateDate': datetime(2015, 1, 1),
    'GroupPolicyList': [
        {
            'PolicyName': 'string',
            'PolicyDocument': 'string'
        },
    ],
    'AttachedManagedPolicies': [
        {
            'PolicyName': 'string',
            'PolicyArn': 'string'
        },
    ],
},
],
'RoleDetailList': [
    {
        'Path': 'string',
        'RoleName': 'string',
        'RoleId': 'string',
        'Arn': 'string',
        'CreateDate': datetime(2015, 1, 1),
        'AssumeRolePolicyDocument': 'string',
        'InstanceProfileList': [
            {
                'Path': 'string',
                'InstanceProfileName': 'string',
                'InstanceProfileId': 'string',
                'Arn': 'string',
                'CreateDate': datetime(2015, 1, 1),
                'Roles': [
                    {
                        'Path': 'string',
                        'RoleName': 'string',
                        'RoleId': 'string',
                        'Arn': 'string',
                        'CreateDate': datetime(2015, 1, 1),
                        'AssumeRolePolicyDocument': 'string'
                    },
                ],
            },
        ],
    },
],
'RolePolicyList': [
    {
        'PolicyName': 'string',
        'PolicyDocument': 'string'
    },
],
'AttachedManagedPolicies': [
    {
        'PolicyName': 'string',
        'PolicyArn': 'string'
    },
],
```

```

    ]
  },
],
'Policies': [
  {
    'PolicyName': 'string',
    'PolicyId': 'string',
    'Arn': 'string',
    'Path': 'string',
    'DefaultVersionId': 'string',
    'AttachmentCount': 123,
    'IsAttachable': True|False,
    'Description': 'string',
    'CreateDate': datetime(2015, 1, 1),
    'UpdateDate': datetime(2015, 1, 1),
    'PolicyVersionList': [
      {
        'Document': 'string',
        'VersionId': 'string',
        'IsDefaultVersion': True|False,
        'CreateDate': datetime(2015, 1, 1)
      },
    ]
  },
],
'IsTruncated': True|False,
'Marker': 'string'
}

```

Response Structure

- *(dict)* –

Contains the response to a successful GetAccountAuthorizationDetails request.

- **UserDetailList** (*list*) –

A list containing information about IAM users.

- * *(dict)* –

Contains information about an IAM user, including all the user's policies and all the IAM groups the user is in.

This data type is used as a response element in the GetAccountAuthorizationDetails action.

- **Path** (*string*) –

The path to the user. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **UserName** (*string*) –

The friendly name identifying the user.

- **UserId** (*string*) –

The stable and unique string identifying the user. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- **Arn** (*string*) –

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

- **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the user was created.

- **UserPolicyList** (*list*) –

A list of the inline policies embedded in the user.

- (*dict*) –

Contains information about an IAM policy, including the policy document.

This data type is used as a response element in the `GetAccountAuthorizationDetails` action.

- **PolicyName** (*string*) –

The name of the policy.

- **PolicyDocument** (*string*) –

The policy document.

- **GroupList** (*list*) –

A list of IAM groups that the user is in.

- (*string*) –

- **AttachedManagedPolicies** (*list*) –

A list of the managed policies attached to the user.

- (*dict*) –

Contains information about an attached policy.

An attached policy is a managed policy that has been attached to a user, group, or role. This data type is used as a response element in the `ListAttachedGroupPolicies`, `ListAttachedRolePolicies`, `ListAttachedUserPolicies`, and `GetAccountAuthorizationDetails` actions.

For more information about managed policies, refer to [Managed Policies and Inline Policies](#) in the *Using IAM* guide.

- **PolicyName** (*string*) –

The friendly name of the attached policy.

- **PolicyArn** (*string*) –

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

- **GroupDetailList** (*list*) –

A list containing information about IAM groups.

- * (*dict*) –

Contains information about an IAM group, including all of the group's policies.

This data type is used as a response element in the `GetAccountAuthorizationDetails` action.

- **Path** (*string*) –

The path to the group. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **GroupName** (*string*) –

The friendly name that identifies the group.

- **GroupId** (*string*) –

The stable and unique string identifying the group. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- **Arn** (*string*) –

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

- **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the group was created.

- **GroupPolicyList** (*list*) –

A list of the inline policies embedded in the group.

- (*dict*) –

Contains information about an IAM policy, including the policy document.

This data type is used as a response element in the `GetAccountAuthorizationDetails` action.

- **PolicyName** (*string*) –

The name of the policy.

- **PolicyDocument** (*string*) –

The policy document.

- **AttachedManagedPolicies** (*list*) –

A list of the managed policies attached to the group.

- (*dict*) –

Contains information about an attached policy.

An attached policy is a managed policy that has been attached to a user, group, or role. This data type is used as a response element in the `ListAttachedGroupPolicies`, `ListAttachedRolePolicies`, `ListAttachedUserPolicies`, and `GetAccountAuthorizationDetails` actions.

For more information about managed policies, refer to [Managed Policies and Inline Policies](#) in the *Using IAM* guide.

- **PolicyName** (*string*) –

The friendly name of the attached policy.

- **PolicyArn** (*string*) –

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

– **RoleDetailList** (*list*) –

A list containing information about IAM roles.

* (*dict*) –

Contains information about an IAM role, including all of the role's policies.

This data type is used as a response element in the `GetAccountAuthorizationDetails` action.

· **Path** (*string*) –

The path to the role. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

· **RoleName** (*string*) –

The friendly name that identifies the role.

· **RoleId** (*string*) –

The stable and unique string identifying the role. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

· **Arn** (*string*) –

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

· **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the role was created.

· **AssumeRolePolicyDocument** (*string*) –

The trust policy that grants permission to assume the role.

· **InstanceProfileList** (*list*) –

Contains a list of instance profiles.

· (*dict*) –

Contains information about an instance profile.

This data type is used as a response element in the following actions:

· `CreateInstanceProfile`

· `GetInstanceProfile`

· `ListInstanceProfiles`

· `ListInstanceProfilesForRole`

· **Path** (*string*) –

The path to the instance profile. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **InstanceProfileName** (*string*) –
The name identifying the instance profile.
- **InstanceProfileId** (*string*) –
The stable and unique string identifying the instance profile. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.
- **Arn** (*string*) –
The Amazon Resource Name (ARN) specifying the instance profile. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.
- **CreateDate** (*datetime*) –
The date when the instance profile was created.
- **Roles** (*list*) –
The role associated with the instance profile.
- (*dict*) –
Contains information about an IAM role.

This data type is used as a response element in the following actions:
 - CreateRole
 - GetRole
 - ListRoles
- **Path** (*string*) –
The path to the role. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.
- **RoleName** (*string*) –
The friendly name that identifies the role.
- **RoleId** (*string*) –
The stable and unique string identifying the role. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.
- **Arn** (*string*) –
The Amazon Resource Name (ARN) specifying the role. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.
- **CreateDate** (*datetime*) –
The date and time, in [ISO 8601 date-time format](#) , when the role was created.
- **AssumeRolePolicyDocument** (*string*) –
The policy that grants an entity permission to assume the role.
- **RolePolicyList** (*list*) –
A list of inline policies embedded in the role. These policies are the role's access (permissions) policies.
- (*dict*) –
Contains information about an IAM policy, including the policy document.

This data type is used as a response element in the `GetAccountAuthorizationDetails` action.

- **PolicyName** (*string*) –

The name of the policy.

- **PolicyDocument** (*string*) –

The policy document.

- **AttachedManagedPolicies** (*list*) –

A list of managed policies attached to the role. These policies are the role's access (permissions) policies.

- (*dict*) –

Contains information about an attached policy.

An attached policy is a managed policy that has been attached to a user, group, or role. This data type is used as a response element in the `ListAttachedGroupPolicies`, `ListAttachedRolePolicies`, `ListAttachedUserPolicies`, and `GetAccountAuthorizationDetails` actions.

For more information about managed policies, refer to [Managed Policies and Inline Policies](#) in the *Using IAM* guide.

- **PolicyName** (*string*) –

The friendly name of the attached policy.

- **PolicyArn** (*string*) –

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

– **Policies** (*list*) –

A list containing information about managed policies.

- * (*dict*) –

Contains information about a managed policy, including the policy's ARN, versions, and the number of principal entities (users, groups, and roles) that the policy is attached to.

This data type is used as a response element in the `GetAccountAuthorizationDetails` action.

For more information about managed policies, see [Managed Policies and Inline Policies](#) in the *Using IAM* guide.

- **PolicyName** (*string*) –

The friendly name (not ARN) identifying the policy.

- **PolicyId** (*string*) –

The stable and unique string identifying the policy.

For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- **Arn** (*string*) –

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

- **Path** (*string*) –

The path to the policy.

For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **DefaultVersionId** (*string*) –

The identifier for the version of the policy that is set as the default (operative) version.

For more information about policy versions, see [Versioning for Managed Policies](#) in the *Using IAM* guide.

- **AttachmentCount** (*integer*) –

The number of principal entities (users, groups, and roles) that the policy is attached to.

- **IsAttachable** (*boolean*) –

Specifies whether the policy can be attached to an IAM user, group, or role.

- **Description** (*string*) –

A friendly description of the policy.

- **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the policy was created.

- **UpdateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the policy was last updated.

When a policy has only one version, this field contains the date and time when the policy was created. When a policy has more than one version, this field contains the date and time when the most recent policy version was created.

- **PolicyVersionList** (*list*) –

A list containing information about the versions of the policy.

- (*dict*) –

Contains information about a version of a managed policy.

This data type is used as a response element in the `CreatePolicyVersion`, `GetPolicyVersion`, `ListPolicyVersions`, and `GetAccountAuthorizationDetails` actions.

For more information about managed policies, refer to [Managed Policies and Inline Policies](#) in the *Using IAM* guide.

- **Document** (*string*) –

The policy document.

The policy document is returned in the response to the `GetPolicyVersion` and `GetAccountAuthorizationDetails` operations. It is not returned in the response to the `CreatePolicyVersion` or `ListPolicyVersions` operations.

- **VersionId** (*string*) –

The identifier for the policy version.

Policy version identifiers always begin with `v` (always lower-case). When a policy is created, the first policy version is `v1`.

- **IsDefaultVersion** (*boolean*) –

Specifies whether the policy version is set as the policy's default version.

- **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the policy version was created.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

- **Marker** (*string*) –

When `IsTruncated` is `true`, this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

get_account_password_policy()

Retrieves the password policy for the AWS account. For more information about using a password policy, go to [Managing an IAM Password Policy](#).

Request Syntax

```
response = client.get_account_password_policy()
```

Return type dict

Returns

Response Syntax

```
{
  'PasswordPolicy': {
    'MinimumPasswordLength': 123,
    'RequireSymbols': True|False,
    'RequireNumbers': True|False,
    'RequireUppercaseCharacters': True|False,
    'RequireLowercaseCharacters': True|False,
    'AllowUsersToChangePassword': True|False,
    'ExpirePasswords': True|False,
    'MaxPasswordAge': 123,
    'PasswordReusePrevention': 123,
    'HardExpiry': True|False
  }
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `GetAccountPasswordPolicy` request.

– **PasswordPolicy** (*dict*) –

Contains information about the account password policy.

This data type is used as a response element in the GetAccountPasswordPolicy action.

* **MinimumPasswordLength** (*integer*) –

Minimum length to require for IAM user passwords.

* **RequireSymbols** (*boolean*) –

Specifies whether to require symbols for IAM user passwords.

* **RequireNumbers** (*boolean*) –

Specifies whether to require numbers for IAM user passwords.

* **RequireUppercaseCharacters** (*boolean*) –

Specifies whether to require uppercase characters for IAM user passwords.

* **RequireLowercaseCharacters** (*boolean*) –

Specifies whether to require lowercase characters for IAM user passwords.

* **AllowUsersToChangePassword** (*boolean*) –

Specifies whether IAM users are allowed to change their own password.

* **ExpirePasswords** (*boolean*) –

Specifies whether IAM users are required to change their password after a specified number of days.

* **MaxPasswordAge** (*integer*) –

The number of days that an IAM user password is valid.

* **PasswordReusePrevention** (*integer*) –

Specifies the number of previous passwords that IAM users are prevented from reusing.

* **HardExpiry** (*boolean*) –

Specifies whether IAM users are prevented from setting a new password after their password has expired.

get_account_summary()

Retrieves information about IAM entity usage and IAM quotas in the AWS account.

For information about limitations on IAM entities, see [Limitations on IAM Entities](#) in the *IAM User Guide*.

Request Syntax

```
response = client.get_account_summary()
```

Return type dict

Returns

Response Syntax

```
{
    'SummaryMap': {
        'string': 123
```

```
}  
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `GetAccountSummary` request.

- **SummaryMap** (*dict*) –

A set of key value pairs containing information about IAM entity usage and IAM quotas.

SummaryMap contains the following keys:

- * **AccessKeysPerUserQuota** The maximum number of active access keys allowed for each IAM user.
- * **AccountAccessKeysPresent** This value is 1 if the AWS account (root) has an access key, otherwise it is 0.
- * **AccountMFAEnabled** This value is 1 if the AWS account (root) has an MFA device assigned, otherwise it is 0.
- * **AccountSigningCertificatesPresent** This value is 1 if the AWS account (root) has a signing certificate, otherwise it is 0.
- * **AssumeRolePolicySizeQuota** The maximum allowed size for assume role policy documents (trust policies), in non-whitespace characters.
- * **AttachedPoliciesPerGroupQuota** The maximum number of managed policies that can be attached to an IAM group.
- * **AttachedPoliciesPerRoleQuota** The maximum number of managed policies that can be attached to an IAM role.
- * **AttachedPoliciesPerUserQuota** The maximum number of managed policies that can be attached to an IAM user.
- * **GroupPolicySizeQuota** The maximum allowed size for the aggregate of all inline policies embedded in an IAM group, in non-whitespace characters.
- * **Groups** The number of IAM groups in the AWS account.
- * **GroupsPerUserQuota** The maximum number of IAM groups each IAM user can belong to.
- * **GroupsQuota** The maximum number of IAM groups allowed in the AWS account.
- * **InstanceProfiles** The number of instance profiles in the AWS account.
- * **InstanceProfilesQuota** The maximum number of instance profiles allowed in the AWS account.
- * **MFADevices** The number of MFA devices in the AWS account, including those assigned and unassigned.
- * **MFADevicesInUse** The number of MFA devices that have been assigned to an IAM user or to the AWS account (root).
- * **Policies** The number of customer managed policies in the AWS account.
- * **PoliciesQuota** The maximum number of customer managed policies allowed in the AWS account.
- * **PolicySizeQuota** The maximum allowed size of a customer managed policy, in non-whitespace characters.
- * **PolicyVersionsInUse** The number of managed policies that are attached to IAM users, groups, or roles in the AWS account.
- * **PolicyVersionsInUseQuota** The maximum number of managed

policies that can be attached to IAM users, groups, or roles in the AWS account.

- * **Providers** The number of identity providers in the AWS account.
- * **RolePolicySizeQuota** The maximum allowed size for the aggregate of all inline policies (access policies, not the trust policy) embedded in an IAM role, in non-whitespace characters.
- * **Roles** The number of IAM roles in the AWS account.
- * **RolesQuota** The maximum number of IAM roles allowed in the AWS account.
- * **ServerCertificates** The number of server certificates in the AWS account.
- * **ServerCertificatesQuota** The maximum number of server certificates allowed in the AWS account.
- * **SigningCertificatesPerUserQuota** The maximum number of X.509 signing certificates allowed for each IAM user.
- * **UserPolicySizeQuota** The maximum allowed size for the aggregate of all inline policies embedded in an IAM user, in non-whitespace characters.
- * **Users** The number of IAM users in the AWS account.
- * **UsersQuota** The maximum number of IAM users allowed in the AWS account.
- * **VersionsPerPolicyQuota** The maximum number of policy versions allowed for each managed policy.
- * *(string)* –
- *(integer)* –

get_context_keys_for_custom_policy (***kwargs*)

Gets a list of all of the context keys referenced in `Condition` elements in the input policies. The policies are supplied as a list of one or more strings. To get the context keys from policies associated with an IAM user, group, or role, use `GetContextKeysForPrincipalPolicy`.

Context keys are variables maintained by AWS and its services that provide details about the context of an API query request, and can be evaluated by using the `Condition` element of an IAM policy. Use `GetContextKeysForCustomPolicy` to understand what key names and values you must supply when you call `SimulateCustomPolicy`. Note that all parameters are shown in unencoded form here for clarity, but must be URL encoded to be included as a part of a real HTML request.

Request Syntax

```
response = client.get_context_keys_for_custom_policy(
    PolicyInputList=[
        'string',
    ]
)
```

Parameters `PolicyInputList` (*list*) – [REQUIRED]

A list of policies for which you want list of context keys used in `Condition` elements.

- *(string)* –

Return type dict

Returns

Response Syntax

```
{
    'ContextKeyNames': [
```

```
        'string',  
    ]  
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `GetContextKeysForPrincipalPolicy` or `GetContextKeysForCustomPolicy` request.

- **ContextKeyNames** (*list*) –

The list of context keys that are used in the `Condition` elements of the input policies.

- * (*string*) –

`get_context_keys_for_principal_policy` (**kwargs)

Gets a list of all of the context keys referenced in `Condition` elements in all of the IAM policies attached to the specified IAM entity. The entity can be an IAM user, group, or role. If you specify a user, then the request also includes all of the policies attached to groups that the user is a member of.

You can optionally include a list of one or more additional policies, specified as strings. If you want to include only a list of policies by string, use `GetContextKeysForCustomPolicy` instead.

Note: This API discloses information about the permissions granted to other users. If you do not want users to see other user’s permissions, then consider allowing them to use `GetContextKeysForCustomPolicy` instead.

Context keys are variables maintained by AWS and its services that provide details about the context of an API query request, and can be evaluated by using the `Condition` element of an IAM policy. Use `GetContextKeysForPrincipalPolicy` to understand what key names and values you must supply when you call `SimulatePrincipalPolicy`.

Request Syntax

```
response = client.get_context_keys_for_principal_policy(  
    PolicySourceArn='string',  
    PolicyInputList=[  
        'string',  
    ]  
)
```

Parameters

- **PolicySourceArn** (*string*) – **[REQUIRED]**

The ARN of a user, group, or role whose policies contain the context keys that you want listed. If you specify a user, the list includes context keys that are found in all policies attached to the user as well as to all groups that the user is a member of. If you pick a group or a role, then it includes only those context keys that are found in policies attached to that entity. Note that all parameters are shown in unencoded form here for clarity, but must be URL encoded to be included as a part of a real HTML request.

- **PolicyInputList** (*list*) – A optional list of additional policies for which you want list of context keys used in `Condition` elements.
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'ContextKeyNames': [
        'string',
    ]
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `GetContextKeysForPrincipalPolicy` or `GetContextKeysForCustomPolicy` request.

- **ContextKeyNames** (*list*) –

The list of context keys that are used in the `Condition` elements of the input policies.

* (*string*) –

`get_credential_report()`

Retrieves a credential report for the AWS account. For more information about the credential report, see [Getting Credential Reports](#) in the *IAM User Guide*.

Request Syntax

```
response = client.get_credential_report()
```

Return type `dict`

Returns

Response Syntax

```
{
    'Content': b'bytes',
    'ReportFormat': 'text/csv',
    'GeneratedTime': datetime(2015, 1, 1)
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `GetCredentialReport` request.

- **Content** (*bytes*) –

Contains the credential report. The report is Base64-encoded.

- **ReportFormat** (*string*) –

The format (MIME type) of the credential report.

- **GeneratedTime** (*datetime*) –

The date and time when the credential report was created, in [ISO 8601 date-time format](#).

`get_group(**kwargs)`

Returns a list of users that are in the specified group. You can paginate the results using the `MaxItems` and `Marker` parameters.

Request Syntax

```
response = client.get_group(
    GroupName='string',
    Marker='string',
```

```
MaxItems=123
)
```

Parameters

- **GroupName** (*string*) – [REQUIRED]

The name of the group.

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns

Response Syntax

```
{
  'Group': {
    'Path': 'string',
    'GroupName': 'string',
    'GroupId': 'string',
    'Arn': 'string',
    'CreateDate': datetime(2015, 1, 1)
  },
  'Users': [
    {
      'Path': 'string',
      'UserName': 'string',
      'UserId': 'string',
      'Arn': 'string',
      'CreateDate': datetime(2015, 1, 1),
      'PasswordLastUsed': datetime(2015, 1, 1)
    },
  ],
  'IsTruncated': True|False,
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `GetGroup` request.

- **Group** (*dict*) –

Information about the group.

- * **Path** (*string*) –

The path to the group. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **GroupName** (*string*) –

The friendly name that identifies the group.

- * **GroupId** (*string*) –

The stable and unique string identifying the group. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **Arn** (*string*) –

The Amazon Resource Name (ARN) specifying the group. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the group was created.

- **Users** (*list*) –

A list of users in the group.

- * (*dict*) –

Contains information about an IAM user entity.

This data type is used as a response element in the following actions:

- CreateUser
- GetUser
- ListUsers

- **Path** (*string*) –

The path to the user. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **UserName** (*string*) –

The friendly name identifying the user.

- **UserId** (*string*) –

The stable and unique string identifying the user. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) that identifies the user. For more information about ARNs and how to use ARNs in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the user was created.

- **PasswordLastUsed** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the user's password was last used to sign in to an AWS website. For a list of AWS websites that capture a user's last sign-in time, see the [Credential Reports](#) topic in the *Using IAM* guide. If a password is used more than once in a five-minute span, only the first use is returned in this field. This field is null (not present) when:

- The user does not have a password
- The password exists but has never been used (at least not since IAM started tracking this information on October 20th, 2014)

- there is no sign-in data associated with the user

This value is returned only in the `GetUser` and `ListUsers` actions.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

- **Marker** (*string*) –

When `IsTruncated` is `true`, this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

get_group_policy (***kwargs*)

Retrieves the specified inline policy document that is embedded in the specified group.

A group can also have managed policies attached to it. To retrieve a managed policy document that is attached to a group, use `GetPolicy` to determine the policy's default version, then use `GetPolicyVersion` to retrieve the policy document.

For more information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = client.get_group_policy(
    GroupName='string',
    PolicyName='string'
)
```

Parameters

- **GroupName** (*string*) – [REQUIRED]

The name of the group the policy is associated with.

- **PolicyName** (*string*) – [REQUIRED]

The name of the policy document to get.

Return type dict

Returns

Response Syntax

```
{
    'GroupName': 'string',
    'PolicyName': 'string',
    'PolicyDocument': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `GetGroupPolicy` request.

- **GroupName** (*string*) –

The group the policy is associated with.

- **PolicyName** (*string*) –
The name of the policy.
- **PolicyDocument** (*string*) –
The policy document.

get_instance_profile (***kwargs*)

Retrieves information about the specified instance profile, including the instance profile's path, GUID, ARN, and role. For more information about instance profiles, go to [About Instance Profiles](#) . For more information about ARNs, go to [ARNs](#) .

Request Syntax

```
response = client.get_instance_profile(
    InstanceProfileName='string'
)
```

Parameters **InstanceProfileName** (*string*) – [REQUIRED]

The name of the instance profile to get information about.

Return type dict

Returns

Response Syntax

```
{
  'InstanceProfile': {
    'Path': 'string',
    'InstanceProfileName': 'string',
    'InstanceProfileId': 'string',
    'Arn': 'string',
    'CreateDate': datetime(2015, 1, 1),
    'Roles': [
      {
        'Path': 'string',
        'RoleName': 'string',
        'RoleId': 'string',
        'Arn': 'string',
        'CreateDate': datetime(2015, 1, 1),
        'AssumeRolePolicyDocument': 'string'
      },
    ]
  }
}
```

Response Structure

- (*dict*) –

Contains the response to a successful GetInstanceProfile request.

- **InstanceProfile** (*dict*) –

Information about the instance profile.

- * **Path** (*string*) –

The path to the instance profile. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **InstanceProfileName** (*string*) –

The name identifying the instance profile.

* **InstanceProfileId** (*string*) –

The stable and unique string identifying the instance profile. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

* **Arn** (*string*) –

The Amazon Resource Name (ARN) specifying the instance profile. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

* **CreateDate** (*datetime*) –

The date when the instance profile was created.

* **Roles** (*list*) –

The role associated with the instance profile.

· (*dict*) –

Contains information about an IAM role.

This data type is used as a response element in the following actions:

- CreateRole
- GetRole
- ListRoles
- **Path** (*string*) –

The path to the role. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **RoleName** (*string*) –

The friendly name that identifies the role.

- **RoleId** (*string*) –

The stable and unique string identifying the role. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) specifying the role. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the role was created.

- **AssumeRolePolicyDocument** (*string*) –

The policy that grants an entity permission to assume the role.

get_login_profile (***kwargs*)

Retrieves the user name and password-creation date for the specified user. If the user has not been assigned a password, the action returns a 404 (`NoSuchEntity`) error.

Request Syntax

```
response = client.get_login_profile(  
    UserName='string'  
)
```

Parameters `UserName` (*string*) – [REQUIRED]

The name of the user whose login profile you want to retrieve.

Return type dict

Returns

Response Syntax

```
{
  'LoginProfile': {
    'UserName': 'string',
    'CreateDate': datetime(2015, 1, 1),
    'PasswordResetRequired': True|False
  }
}
```

Response Structure

- (*dict*) –

Contains the response to a successful GetLoginProfile request.

- **LoginProfile** (*dict*) –

The user name and password create date for the user.

- * **UserName** (*string*) –

The name of the user, which can be used for signing in to the AWS Management Console.

- * **CreateDate** (*datetime*) –

The date when the password for the user was created.

- * **PasswordResetRequired** (*boolean*) –

Specifies whether the user is required to set a new password on next sign-in.

get_open_id_connect_provider (***kwargs*)

Returns information about the specified OpenID Connect provider.

Request Syntax

```
response = client.get_open_id_connect_provider(
    OpenIDConnectProviderArn='string'
)
```

Parameters `OpenIDConnectProviderArn` (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the IAM OpenID Connect (OIDC) provider to get information for. You can get a list of OIDC provider ARNs by using the ListOpenIDConnectProviders action.

Return type dict

Returns

Response Syntax

```
{
  'Url': 'string',
  'ClientIDList': [
    'string',
  ],
  'ThumbprintList': [
    'string',
  ]
}
```

```
],  
  'CreateDate': datetime(2015, 1, 1)  
}
```

Response Structure

- (*dict*) –

Contains the response to a successful GetOpenIDConnectProvider request.

- **Url** (*string*) –

The URL that the IAM OpenID Connect provider is associated with. For more information, see CreateOpenIDConnectProvider .

- **ClientIDList** (*list*) –

A list of client IDs (also known as audiences) that are associated with the specified IAM OpenID Connect provider. For more information, see CreateOpenIDConnectProvider .

* (*string*) –

- **ThumbprintList** (*list*) –

A list of certificate thumbprints that are associated with the specified IAM OpenID Connect provider. For more information, see CreateOpenIDConnectProvider .

* (*string*) –

Contains a thumbprint for an identity provider's server certificate.

The identity provider's server certificate thumbprint is the hex-encoded SHA-1 hash value of the self-signed X.509 certificate used by the domain where the OpenID Connect provider makes its keys available. It is always a 40-character string.

- **CreateDate** (*datetime*) –

The date and time when the IAM OpenID Connect provider entity was created in the AWS account.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_policy (***kwargs*)

Retrieves information about the specified managed policy, including the policy's default version and the total number of users, groups, and roles that the policy is attached to. For a list of the specific users, groups, and roles that the policy is attached to, use the ListEntitiesForPolicy API. This API returns metadata about the policy. To retrieve the policy document for a specific version of the policy, use GetPolicyVersion .

This API retrieves information about managed policies. To retrieve information about an inline policy that is embedded with a user, group, or role, use the GetUserPolicy , GetGroupPolicy , or GetRolePolicy API.

For more information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = client.get_policy(
    PolicyArn='string'
)
```

Parameters **PolicyArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

Return type dict

Returns

Response Syntax

```
{
  'Policy': {
    'PolicyName': 'string',
    'PolicyId': 'string',
    'Arn': 'string',
    'Path': 'string',
    'DefaultVersionId': 'string',
    'AttachmentCount': 123,
    'IsAttachable': True|False,
    'Description': 'string',
    'CreateDate': datetime(2015, 1, 1),
    'UpdateDate': datetime(2015, 1, 1)
  }
}
```

Response Structure

- (*dict*) –

Contains the response to a successful GetPolicy request.

- **Policy** (*dict*) –

Information about the policy.

- * **PolicyName** (*string*) –

The friendly name (not ARN) identifying the policy.

- * **PolicyId** (*string*) –

The stable and unique string identifying the policy.

For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **Arn** (*string*) –

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

- * **Path** (*string*) –

The path to the policy.

For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **DefaultVersionId** (*string*) –

The identifier for the version of the policy that is set as the default version.

- * **AttachmentCount** (*integer*) –

The number of entities (users, groups, and roles) that the policy is attached to.

- * **IsAttachable** (*boolean*) –

Specifies whether the policy can be attached to an IAM user, group, or role.

- * **Description** (*string*) –

A friendly description of the policy.

This element is included in the response to the `GetPolicy` operation. It is not included in the response to the `ListPolicies` operation.

- * **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#) , when the policy was created.

- * **UpdateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#) , when the policy was last updated.

When a policy has only one version, this field contains the date and time when the policy was created. When a policy has more than one version, this field contains the date and time when the most recent policy version was created.

get_policy_version (***kwargs*)

Retrieves information about the specified version of the specified managed policy, including the policy document.

To list the available versions for a policy, use `ListPolicyVersions` .

This API retrieves information about managed policies. To retrieve information about an inline policy that is embedded in a user, group, or role, use the `GetUserPolicy` , `GetGroupPolicy` , or `GetRolePolicy` API.

For more information about the types of policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide* .

Request Syntax

```
response = client.get_policy_version(
    PolicyArn='string',
    VersionId='string'
)
```

Parameters

- **PolicyArn** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\)](#) and [AWS Service Namespaces](#) in the *AWS General Reference* .

- **VersionId** (*string*) – **[REQUIRED]**

Identifies the policy version to retrieve.

Return type dict

Returns

Response Syntax

```
{
  'PolicyVersion': {
    'Document': 'string',
    'VersionId': 'string',
    'IsDefaultVersion': True|False,
    'CreateDate': datetime(2015, 1, 1)
  }
}
```

Response Structure

- (*dict*) –

Contains the response to a successful GetPolicyVersion request.

- **PolicyVersion** (*dict*) –

Information about the policy version.

For more information about managed policy versions, see [Versioning for Managed Policies](#) in the *IAM User Guide* .

- * **Document** (*string*) –

The policy document.

The policy document is returned in the response to the GetPolicyVersion and GetAccountAuthorizationDetails operations. It is not returned in the response to the CreatePolicyVersion or ListPolicyVersions operations.

- * **VersionId** (*string*) –

The identifier for the policy version.

Policy version identifiers always begin with v (always lowercase). When a policy is created, the first policy version is v1 .

- * **IsDefaultVersion** (*boolean*) –

Specifies whether the policy version is set as the policy's default version.

- * **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#) , when the policy version was created.

get_role (***kwargs*)

Retrieves information about the specified role, including the role's path, GUID, ARN, and the policy granting permission to assume the role. For more information about ARNs, go to [ARNs](#) . For more information about roles, go to [Working with Roles](#) .

Request Syntax

```
response = client.get_role(  
    RoleName='string'  
)
```

Parameters **RoleName** (*string*) – [REQUIRED]

The name of the role to get information about.

Return type dict**Returns****Response Syntax**

```
{  
    'Role': {  
        'Path': 'string',  
        'RoleName': 'string',  
        'RoleId': 'string',  
        'Arn': 'string',  
        'CreateDate': datetime(2015, 1, 1),  
        'AssumeRolePolicyDocument': 'string'  
    }  
}
```

Response Structure

- (*dict*) –

Contains the response to a successful GetRole request.

- **Role** (*dict*) –

Information about the role.

- * **Path** (*string*) –

The path to the role. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **RoleName** (*string*) –

The friendly name that identifies the role.

- * **RoleId** (*string*) –

The stable and unique string identifying the role. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **Arn** (*string*) –

The Amazon Resource Name (ARN) specifying the role. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#) , when the role was created.

- * **AssumeRolePolicyDocument** (*string*) –

The policy that grants an entity permission to assume the role.

get_role_policy (***kwargs*)

Retrieves the specified inline policy document that is embedded with the specified role.

A role can also have managed policies attached to it. To retrieve a managed policy document that is attached to a role, use `GetPolicy` to determine the policy’s default version, then use `GetPolicyVersion` to retrieve the policy document.

For more information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

For more information about roles, go to [Using Roles to Delegate Permissions and Federate Identities](#).

Request Syntax

```
response = client.get_role_policy(
    RoleName='string',
    PolicyName='string'
)
```

Parameters

- **RoleName** (*string*) – [REQUIRED]
The name of the role associated with the policy.
- **PolicyName** (*string*) – [REQUIRED]
The name of the policy document to get.

Return type dict

Returns

Response Syntax

```
{
    'RoleName': 'string',
    'PolicyName': 'string',
    'PolicyDocument': 'string'
}
```

Response Structure

- (*dict*) –
Contains the response to a successful GetRolePolicy request.
 - **RoleName** (*string*) –
The role the policy is associated with.
 - **PolicyName** (*string*) –
The name of the policy.
 - **PolicyDocument** (*string*) –
The policy document.

get_saml_provider (***kwargs*)

Returns the SAML provider metadocument that was uploaded when the provider was created or updated.

Note: This operation requires [Signature Version 4](#).

Request Syntax

```
response = client.get_saml_provider(
    SAMLProviderArn='string'
)
```

Parameters **SAMLProviderArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the SAML provider to get information about.

Return type dict

Returns

Response Syntax

```
{
    'SAMLMetadataDocument': 'string',
    'CreateDate': datetime(2015, 1, 1),
    'ValidUntil': datetime(2015, 1, 1)
}
```

Response Structure

- (*dict*) –

Contains the response to a successful GetSAMLProvider request.

- **SAMLMetadataDocument** (*string*) –

The XML metadata document that includes information about an identity provider.

- **CreateDate** (*datetime*) –

The date and time when the SAML provider was created.

- **ValidUntil** (*datetime*) –

The expiration date and time for the SAML provider.

get_server_certificate (***kwargs*)

Retrieves information about the specified server certificate.

Request Syntax

```
response = client.get_server_certificate(
    ServerCertificateName='string'
)
```

Parameters **ServerCertificateName** (*string*) – [REQUIRED]

The name of the server certificate you want to retrieve information about.

Return type dict

Returns

Response Syntax

```
{
    'ServerCertificate': {
        'ServerCertificateMetadata': {
            'Path': 'string',
            'ServerCertificateName': 'string',
            'ServerCertificateId': 'string',
            'Arn': 'string',
            'UploadDate': datetime(2015, 1, 1),
            'Expiration': datetime(2015, 1, 1)
        },
        'CertificateBody': 'string',
        'CertificateChain': 'string'
    }
}
```

Response Structure

- (*dict*) –

Contains the response to a successful GetServerCertificate request.

– **ServerCertificate** (*dict*) –

Information about the server certificate.

* **ServerCertificateMetadata** (*dict*) –

The meta information of the server certificate, such as its name, path, ID, and ARN.

· **Path** (*string*) –

The path to the server certificate. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

· **ServerCertificateName** (*string*) –

The name that identifies the server certificate.

· **ServerCertificateId** (*string*) –

The stable and unique string identifying the server certificate. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

· **Arn** (*string*) –

The Amazon Resource Name (ARN) specifying the server certificate. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

· **UploadDate** (*datetime*) –

The date when the server certificate was uploaded.

· **Expiration** (*datetime*) –

The date on which the certificate is set to expire.

* **CertificateBody** (*string*) –

The contents of the public key certificate.

* **CertificateChain** (*string*) –

The contents of the public key certificate chain.

get_ssh_public_key (***kwargs*)

Retrieves the specified SSH public key, including metadata about the key.

The SSH public key retrieved by this action is used only for authenticating the associated IAM user to an AWS CodeCommit repository. For more information about using SSH keys to authenticate to an AWS CodeCommit repository, see [Set up AWS CodeCommit for SSH Connections](#) in the *AWS CodeCommit User Guide*.

Request Syntax

```
response = client.get_ssh_public_key(
    UserName='string',
    SSHPublicKeyId='string',
    Encoding='SSH' | 'PEM'
)
```

Parameters

• **UserName** (*string*) – [REQUIRED]

The name of the IAM user associated with the SSH public key.

• **SSHPublicKeyId** (*string*) – [REQUIRED]

The unique identifier for the SSH public key.

- **Encoding** (*string*) – [REQUIRED]

Specifies the public key encoding format to use in the response. To retrieve the public key in ssh-rsa format, use `SSH`. To retrieve the public key in PEM format, use `PEM`.

Return type dict

Returns

Response Syntax

```
{
  'SSHPublicKey': {
    'UserName': 'string',
    'SSHPublicKeyId': 'string',
    'Fingerprint': 'string',
    'SSHPublicKeyBody': 'string',
    'Status': 'Active'|'Inactive',
    'UploadDate': datetime(2015, 1, 1)
  }
}
```

Response Structure

- (*dict*) –

Contains the response to a successful GetSSHPublicKey request.

- **SSHPublicKey** (*dict*) –

Information about the SSH public key.

- * **UserName** (*string*) –

The name of the IAM user associated with the SSH public key.

- * **SSHPublicKeyId** (*string*) –

The unique identifier for the SSH public key.

- * **Fingerprint** (*string*) –

The MD5 message digest of the SSH public key.

- * **SSHPublicKeyBody** (*string*) –

The SSH public key.

- * **Status** (*string*) –

The status of the SSH public key. `Active` means the key can be used for authentication with an AWS CodeCommit repository. `Inactive` means the key cannot be used.

- * **UploadDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the SSH public key was uploaded.

get_user (***kwargs*)

Retrieves information about the specified user, including the user's creation date, path, unique ID, and ARN.

If you do not specify a user name, IAM determines the user name implicitly based on the AWS access key ID used to sign the request.

Request Syntax

```
response = client.get_user(
    UserName='string'
)
```

Parameters **UserName** (*string*) – The name of the user to get information about.

This parameter is optional. If it is not included, it defaults to the user making the request.

Return type dict

Returns

Response Syntax

```
{
  'User': {
    'Path': 'string',
    'UserName': 'string',
    'UserId': 'string',
    'Arn': 'string',
    'CreateDate': datetime(2015, 1, 1),
    'PasswordLastUsed': datetime(2015, 1, 1)
  }
}
```

Response Structure

- (*dict*) –

Contains the response to a successful GetUser request.

- **User** (*dict*) –

Information about the user.

- * **Path** (*string*) –

The path to the user. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **UserName** (*string*) –

The friendly name identifying the user.

- * **UserId** (*string*) –

The stable and unique string identifying the user. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **Arn** (*string*) –

The Amazon Resource Name (ARN) that identifies the user. For more information about ARNs and how to use ARNs in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#) , when the user was created.

- * **PasswordLastUsed** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#) , when the user's password was last used to sign in to an AWS website. For a list of AWS websites that capture a user's last sign-in time, see the [Credential Reports](#) topic in the *Using IAM* guide. If a password is used more than once in a five-minute span, only the first use is returned in this field. This field is null (not present) when:

- The user does not have a password

- The password exists but has never been used (at least not since IAM started tracking this information on October 20th, 2014)
- there is no sign-in data associated with the user

This value is returned only in the GetUser and ListUsers actions.

get_user_policy (**kwargs)

Retrieves the specified inline policy document that is embedded in the specified user.

A user can also have managed policies attached to it. To retrieve a managed policy document that is attached to a user, use GetPolicy to determine the policy's default version, then use GetPolicyVersion to retrieve the policy document.

For more information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = client.get_user_policy(
    UserName='string',
    PolicyName='string'
)
```

Parameters

- **UserName** (*string*) – [REQUIRED]
The name of the user who the policy is associated with.
- **PolicyName** (*string*) – [REQUIRED]
The name of the policy document to get.

Return type dict

Returns

Response Syntax

```
{
    'UserName': 'string',
    'PolicyName': 'string',
    'PolicyDocument': 'string'
}
```

Response Structure

- (*dict*) –
Contains the response to a successful GetUserPolicy request.
 - **UserName** (*string*) –
The user the policy is associated with.
 - **PolicyName** (*string*) –
The name of the policy.
 - **PolicyDocument** (*string*) –
The policy document.

get_waiter (*waiter_name*)

list_access_keys (**kwargs)

Returns information about the access key IDs associated with the specified user. If there are none, the action returns an empty list.

Although each user is limited to a small number of keys, you can still paginate the results using the `MaxItems` and `Marker` parameters.

If the `UserName` field is not specified, the `UserName` is determined implicitly based on the AWS access key ID used to sign the request. Because this action works for access keys under the AWS account, you can use this action to manage root credentials even if the AWS account has no associated users.

Note: To ensure the security of your AWS account, the secret access key is accessible only during key and user creation.

Request Syntax

```
response = client.list_access_keys(
    UserName='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **UserName** (*string*) – The name of the user.
- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns

Response Syntax

```
{
  'AccessKeyMetadata': [
    {
      'UserName': 'string',
      'AccessKeyId': 'string',
      'Status': 'Active'|'Inactive',
      'CreateDate': datetime(2015, 1, 1)
    },
  ],
  'IsTruncated': True|False,
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –
Contains the response to a successful `ListAccessKeys` request.
 - **AccessKeyMetadata** (*list*) –
A list of access key metadata.

* (*dict*) –

Contains information about an AWS access key, without its secret key.

This data type is used as a response element in the `ListAccessKeys` action.

- **UserName** (*string*) –

The name of the IAM user that the key is associated with.

- **AccessKeyId** (*string*) –

The ID for this access key.

- **Status** (*string*) –

The status of the access key. `Active` means the key is valid for API calls; `Inactive` means it is not.

- **CreateDate** (*datetime*) –

The date when the access key was created.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

- **Marker** (*string*) –

When `IsTruncated` is `true`, this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

list_account_aliases (***kwargs*)

Lists the account aliases associated with the account. For information about using an AWS account alias, see [Using an Alias for Your AWS Account ID](#) in the *IAM User Guide*.

You can paginate the results using the `MaxItems` and `Marker` parameters.

Request Syntax

```
response = client.list_account_aliases(
    Marker='string',
    MaxItems=123
)
```

Parameters

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and

`Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns

Response Syntax

```
{
    'AccountAliases': [
        'string',
    ],
    'IsTruncated': True|False,
    'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `ListAccountAliases` request.

- **AccountAliases** (*list*) –

A list of aliases associated with the account.

* (*string*) –

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

- **Marker** (*string*) –

When `IsTruncated` is `true`, this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

`list_attached_group_policies` (***kwargs*)

Lists all managed policies that are attached to the specified group.

A group can also have inline policies embedded with it. To list the inline policies for a group, use the `ListGroupPolicies` API. For information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

You can paginate the results using the `MaxItems` and `Marker` parameters. You can use the `PathPrefix` parameter to limit the list of policies to only those matching the specified path prefix. If there are no policies attached to the specified group (or none that match the specified path prefix), the action returns an empty list.

Request Syntax

```
response = client.list_attached_group_policies(
    GroupName='string',
    PathPrefix='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **GroupName** (*string*) – [REQUIRED]

The name (friendly name, not ARN) of the group to list attached policies for.

- **PathPrefix** (*string*) – The path prefix for filtering the results. This parameter is optional. If it is not included, it defaults to a slash (/), listing all policies.
- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns

Response Syntax

```
{
  'AttachedPolicies': [
    {
      'PolicyName': 'string',
      'PolicyArn': 'string'
    },
  ],
  'IsTruncated': True|False,
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `ListAttachedGroupPolicies` request.

- **AttachedPolicies** (*list*) –

A list of the attached policies.

- * (*dict*) –

Contains information about an attached policy.

An attached policy is a managed policy that has been attached to a user, group, or role. This data type is used as a response element in the `ListAttachedGroupPolicies`, `ListAttachedRolePolicies`, `ListAttachedUserPolicies`, and `GetAccountAuthorizationDetails` actions.

For more information about managed policies, refer to [Managed Policies and Inline Policies](#) in the *Using IAM* guide.

- **PolicyName** (*string*) –

The friendly name of the attached policy.

- **PolicyArn** (*string*) –

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

– **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

– **Marker** (*string*) –

When `IsTruncated` is `true`, this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

`list_attached_role_policies` (***kwargs*)

Lists all managed policies that are attached to the specified role.

A role can also have inline policies embedded with it. To list the inline policies for a role, use the `ListRolePolicies` API. For information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

You can paginate the results using the `MaxItems` and `Marker` parameters. You can use the `PathPrefix` parameter to limit the list of policies to only those matching the specified path prefix. If there are no policies attached to the specified role (or none that match the specified path prefix), the action returns an empty list.

Request Syntax

```
response = client.list_attached_role_policies(
    RoleName='string',
    PathPrefix='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **RoleName** (*string*) – [REQUIRED]

The name (friendly name, not ARN) of the role to list attached policies for.

- **PathPrefix** (*string*) – The path prefix for filtering the results. This parameter is optional. If it is not included, it defaults to a slash (/), listing all policies.
- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns

Response Syntax

```
{
  'AttachedPolicies': [
    {
      'PolicyName': 'string',
      'PolicyArn': 'string'
    },
  ],
  'IsTruncated': True|False,
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful ListAttachedRolePolicies request.

- **AttachedPolicies** (*list*) –

A list of the attached policies.

- * (*dict*) –

Contains information about an attached policy.

An attached policy is a managed policy that has been attached to a user, group, or role. This data type is used as a response element in the ListAttachedGroupPolicies , ListAttachedRolePolicies , ListAttachedUserPolicies , and GetAccountAuthorizationDetails actions.

For more information about managed policies, refer to [Managed Policies and Inline Policies](#) in the *Using IAM* guide.

- **PolicyName** (*string*) –

The friendly name of the attached policy.

- **PolicyArn** (*string*) –

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference* .

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the Marker request parameter to retrieve more items. Note that IAM might return fewer than the MaxItems number of results even when there are more results available. We recommend that you check IsTruncated after every call to ensure that you receive all of your results.

- **Marker** (*string*) –

When IsTruncated is true , this element is present and contains the value to use for the Marker parameter in a subsequent pagination request.

list_attached_user_policies (**kwargs)

Lists all managed policies that are attached to the specified user.

A user can also have inline policies embedded with it. To list the inline policies for a user, use the `ListUserPolicies` API. For information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

You can paginate the results using the `MaxItems` and `Marker` parameters. You can use the `PathPrefix` parameter to limit the list of policies to only those matching the specified path prefix. If there are no policies attached to the specified group (or none that match the specified path prefix), the action returns an empty list.

Request Syntax

```
response = client.list_attached_user_policies(
    UserName='string',
    PathPrefix='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **UserName** (*string*) – [REQUIRED]

The name (friendly name, not ARN) of the user to list attached policies for.

- **PathPrefix** (*string*) – The path prefix for filtering the results. This parameter is optional. If it is not included, it defaults to a slash (/), listing all policies.
- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns

Response Syntax

```
{
  'AttachedPolicies': [
    {
      'PolicyName': 'string',
      'PolicyArn': 'string'
    },
  ],
  'IsTruncated': True|False,
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `ListAttachedUserPolicies` request.

- **AttachedPolicies** (*list*) –

A list of the attached policies.

- * (*dict*) –

Contains information about an attached policy.

An attached policy is a managed policy that has been attached to a user, group, or role. This data type is used as a response element in the `ListAttachedGroupPolicies`, `ListAttachedRolePolicies`, `ListAttachedUserPolicies`, and `GetAccountAuthorizationDetails` actions.

For more information about managed policies, refer to [Managed Policies and Inline Policies](#) in the *Using IAM* guide.

- **PolicyName** (*string*) –

The friendly name of the attached policy.

- **PolicyArn** (*string*) –

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

- **Marker** (*string*) –

When `IsTruncated` is `true`, this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

list_entities_for_policy (***kwargs*)

Lists all users, groups, and roles that the specified managed policy is attached to.

You can use the optional `EntityFilter` parameter to limit the results to a particular type of entity (users, groups, or roles). For example, to list only the roles that are attached to the specified policy, set `EntityFilter` to `Role`.

You can paginate the results using the `MaxItems` and `Marker` parameters.

Request Syntax

```
response = client.list_entities_for_policy(
    PolicyArn='string',
    EntityFilter='User'|'Role'|'Group'|'LocalManagedPolicy'|'AWSManagedPolicy',
    PathPrefix='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **PolicyArn** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\)](#) and [AWS Service Namespaces](#) in the *AWS General Reference*.

- **EntityFilter** (*string*) – The entity type to use for filtering the results.

For example, when `EntityFilter` is `Role`, only the roles that are attached to the specified policy are returned. This parameter is optional. If it is not included, all attached entities (users, groups, and roles) are returned.

- **PathPrefix** (*string*) – The path prefix for filtering the results. This parameter is optional. If it is not included, it defaults to a slash (/), listing all entities.
- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns

Response Syntax

```
{
  'PolicyGroups': [
    {
      'GroupName': 'string'
    },
  ],
  'PolicyUsers': [
    {
      'UserName': 'string'
    },
  ],
  'PolicyRoles': [
    {
      'RoleName': 'string'
    },
  ],
  'IsTruncated': True|False,
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `ListEntitiesForPolicy` request.

- **PolicyGroups** (*list*) –

A list of groups that the policy is attached to.

- * (*dict*) –

Contains information about a group that a managed policy is attached to.

This data type is used as a response element in the `ListEntitiesForPolicy` action.

For more information about managed policies, refer to [Managed Policies and Inline Policies](#) in the *Using IAM* guide.

- **GroupName** (*string*) –

The name (friendly name, not ARN) identifying the group.

- **PolicyUsers** (*list*) –

A list of users that the policy is attached to.

- * (*dict*) –

Contains information about a user that a managed policy is attached to.

This data type is used as a response element in the `ListEntitiesForPolicy` action.

For more information about managed policies, refer to [Managed Policies and Inline Policies](#) in the *Using IAM* guide.

- **UserName** (*string*) –

The name (friendly name, not ARN) identifying the user.

- **PolicyRoles** (*list*) –

A list of roles that the policy is attached to.

- * (*dict*) –

Contains information about a role that a managed policy is attached to.

This data type is used as a response element in the `ListEntitiesForPolicy` action.

For more information about managed policies, refer to [Managed Policies and Inline Policies](#) in the *Using IAM* guide.

- **RoleName** (*string*) –

The name (friendly name, not ARN) identifying the role.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

- **Marker** (*string*) –

When `IsTruncated` is `true`, this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

list_group_policies (***kwargs*)

Lists the names of the inline policies that are embedded in the specified group.

A group can also have managed policies attached to it. To list the managed policies that are attached to a group, use `ListAttachedGroupPolicies`. For more information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

You can paginate the results using the `MaxItems` and `Marker` parameters. If there are no inline policies embedded with the specified group, the action returns an empty list.

Request Syntax

```
response = client.list_group_policies(
    GroupName='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **GroupName** (*string*) – [REQUIRED]

The name of the group to list policies for.

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns

Response Syntax

```
{
  'PolicyNames': [
    'string',
  ],
  'IsTruncated': True|False,
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `ListGroupPolicies` request.

- **PolicyNames** (*list*) –

A list of policy names.

* (*string*) –

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

- **Marker** (*string*) –

When `IsTruncated` is `true`, this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

list_groups (**kwargs)

Lists the groups that have the specified path prefix.

You can paginate the results using the `MaxItems` and `Marker` parameters.

Request Syntax

```
response = client.list_groups(  
    PathPrefix='string',  
    Marker='string',  
    MaxItems=123  
)
```

Parameters

- **PathPrefix** (*string*) – The path prefix for filtering the results. For example, the prefix `/division_abc/subdivision_xyz/` gets all groups whose path starts with `/division_abc/subdivision_xyz/`.

This parameter is optional. If it is not included, it defaults to a slash (`/`), listing all groups.

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns**Response Syntax**

```
{  
    'Groups': [  
        {  
            'Path': 'string',  
            'GroupName': 'string',  
            'GroupId': 'string',  
            'Arn': 'string',  
            'CreateDate': datetime(2015, 1, 1)  
        },  
    ],  
    'IsTruncated': True|False,  
    'Marker': 'string'  
}
```

Response Structure

- (*dict*) –
Contains the response to a successful `ListGroups` request.

– **Groups** (*list*) –

A list of groups.

* (*dict*) –

Contains information about an IAM group entity.

This data type is used as a response element in the following actions:

- `CreateGroup`
- `GetGroup`
- `ListGroups`

· **Path** (*string*) –

The path to the group. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

· **GroupName** (*string*) –

The friendly name that identifies the group.

· **GroupId** (*string*) –

The stable and unique string identifying the group. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

· **Arn** (*string*) –

The Amazon Resource Name (ARN) specifying the group. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

· **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the group was created.

– **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

– **Marker** (*string*) –

When `IsTruncated` is `true`, this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

list_groups_for_user (***kwargs*)

Lists the groups the specified user belongs to.

You can paginate the results using the `MaxItems` and `Marker` parameters.

Request Syntax

```
response = client.list_groups_for_user(
    UserName='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **UserName** (*string*) – [REQUIRED]

The name of the user to list groups for.

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns

Response Syntax

```
{
  'Groups': [
    {
      'Path': 'string',
      'GroupName': 'string',
      'GroupId': 'string',
      'Arn': 'string',
      'CreateDate': datetime(2015, 1, 1)
    },
  ],
  'IsTruncated': True|False,
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `ListGroupsForUser` request.

- **Groups** (*list*) –

A list of groups.

- * (*dict*) –

Contains information about an IAM group entity.

This data type is used as a response element in the following actions:

- `CreateGroup`
 - `GetGroup`
 - `ListGroups`

- **Path** (*string*) –

The path to the group. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **GroupName** (*string*) –

The friendly name that identifies the group.

- **GroupId** (*string*) –

The stable and unique string identifying the group. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) specifying the group. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the group was created.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

- **Marker** (*string*) –

When `IsTruncated` is `true`, this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

list_instance_profiles (***kwargs*)

Lists the instance profiles that have the specified path prefix. If there are none, the action returns an empty list. For more information about instance profiles, go to [About Instance Profiles](#).

You can paginate the results using the `MaxItems` and `Marker` parameters.

Request Syntax

```
response = client.list_instance_profiles(
    PathPrefix='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **PathPrefix** (*string*) – The path prefix for filtering the results. For example, the prefix `/application_abc/component_xyz/` gets all instance profiles whose path starts with `/application_abc/component_xyz/`.

This parameter is optional. If it is not included, it defaults to a slash (/), listing all instance profiles.

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results avail-

able. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns

Response Syntax

```
{
  'InstanceProfiles': [
    {
      'Path': 'string',
      'InstanceProfileName': 'string',
      'InstanceProfileId': 'string',
      'Arn': 'string',
      'CreateDate': datetime(2015, 1, 1),
      'Roles': [
        {
          'Path': 'string',
          'RoleName': 'string',
          'RoleId': 'string',
          'Arn': 'string',
          'CreateDate': datetime(2015, 1, 1),
          'AssumeRolePolicyDocument': 'string'
        },
      ],
    },
  ],
  'IsTruncated': True|False,
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `ListInstanceProfiles` request.

- **InstanceProfiles** (*list*) –

A list of instance profiles.

- * (*dict*) –

Contains information about an instance profile.

This data type is used as a response element in the following actions:

- `CreateInstanceProfile`
- `GetInstanceProfile`
- `ListInstanceProfiles`
- `ListInstanceProfilesForRole`

- **Path** (*string*) –

The path to the instance profile. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **InstanceProfileName** (*string*) –

The name identifying the instance profile.

- **InstanceProfileId** (*string*) –

The stable and unique string identifying the instance profile. For more information about IDs, see [IAM Identifiers](#) in the

Using IAM guide.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) specifying the instance profile. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- **CreateDate** (*datetime*) –

The date when the instance profile was created.

- **Roles** (*list*) –

The role associated with the instance profile.

- (*dict*) –

Contains information about an IAM role.

This data type is used as a response element in the following actions:

- `CreateRole`
- `GetRole`
- `ListRoles`
- **Path** (*string*) –

The path to the role. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **RoleName** (*string*) –

The friendly name that identifies the role.

- **RoleId** (*string*) –

The stable and unique string identifying the role. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) specifying the role. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the role was created.

- **AssumeRolePolicyDocument** (*string*) –

The policy that grants an entity permission to assume the role.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

- **Marker** (*string*) –

When `IsTruncated` is `true`, this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

list_instance_profiles_for_role (***kwargs*)

Lists the instance profiles that have the specified associated role. If there are none, the action returns an empty list. For more information about instance profiles, go to [About Instance Profiles](#).

You can paginate the results using the `MaxItems` and `Marker` parameters.

Request Syntax

```
response = client.list_instance_profiles_for_role(
    RoleName='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **RoleName** (*string*) – [REQUIRED]

The name of the role to list instance profiles for.

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns

Response Syntax

```
{
  'InstanceProfiles': [
    {
      'Path': 'string',
      'InstanceProfileName': 'string',
      'InstanceProfileId': 'string',
      'Arn': 'string',
      'CreateDate': datetime(2015, 1, 1),
      'Roles': [
        {
          'Path': 'string',
          'RoleName': 'string',
          'RoleId': 'string',
          'Arn': 'string',
          'CreateDate': datetime(2015, 1, 1),
          'AssumeRolePolicyDocument': 'string'
        },
      ],
    },
  ],
  'IsTruncated': True|False,
  'Marker': 'string'
}
```

Response Structure

- *(dict)* –

Contains the response to a successful `ListInstanceProfilesForRole` request.

- **InstanceProfiles** (*list*) –

A list of instance profiles.

- * *(dict)* –

Contains information about an instance profile.

This data type is used as a response element in the following actions:

- `CreateInstanceProfile`
- `GetInstanceProfile`
- `ListInstanceProfiles`
- `ListInstanceProfilesForRole`

- **Path** (*string*) –

The path to the instance profile. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **InstanceProfileName** (*string*) –

The name identifying the instance profile.

- **InstanceProfileId** (*string*) –

The stable and unique string identifying the instance profile. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) specifying the instance profile. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- **CreateDate** (*datetime*) –

The date when the instance profile was created.

- **Roles** (*list*) –

The role associated with the instance profile.

- *(dict)* –

Contains information about an IAM role.

This data type is used as a response element in the following actions:

- `CreateRole`
- `GetRole`
- `ListRoles`
- **Path** (*string*) –

The path to the role. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **RoleName** (*string*) –

The friendly name that identifies the role.

- **RoleId** (*string*) –

The stable and unique string identifying the role. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) specifying the role. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the role was created.

- **AssumeRolePolicyDocument** (*string*) –

The policy that grants an entity permission to assume the role.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

- **Marker** (*string*) –

When `IsTruncated` is `true`, this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

list_mfa_devices (***kwargs*)

Lists the MFA devices. If the request includes the user name, then this action lists all the MFA devices associated with the specified user name. If you do not specify a user name, IAM determines the user name implicitly based on the AWS access key ID signing the request.

You can paginate the results using the `MaxItems` and `Marker` parameters.

Request Syntax

```
response = client.list_mfa_devices(
    UserName='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **UserName** (*string*) – The name of the user whose MFA devices you want to list.
- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns

Response Syntax

```
{
  'MFADevices': [
    {
      'UserName': 'string',
      'SerialNumber': 'string',
      'EnableDate': datetime(2015, 1, 1)
    },
  ],
  'IsTruncated': True|False,
  'Marker': 'string'
}
```

Response Structure

- *(dict)* –

Contains the response to a successful ListMFADevices request.

- **MFADevices** (*list*) –

A list of MFA devices.

- * *(dict)* –

Contains information about an MFA device.

This data type is used as a response element in the ListMFADevices action.

- **UserName** (*string*) –

The user with whom the MFA device is associated.

- **SerialNumber** (*string*) –

The serial number that uniquely identifies the MFA device. For virtual MFA devices, the serial number is the device ARN.

- **EnableDate** (*datetime*) –

The date when the MFA device was enabled for the user.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the Marker request parameter to retrieve more items. Note that IAM might return fewer than the MaxItems number of results even when there are more results available. We recommend that you check IsTruncated after every call to ensure that you receive all of your results.

- **Marker** (*string*) –

When IsTruncated is true, this element is present and contains the value to use for the Marker parameter in a subsequent pagination request.

`list_open_id_connect_providers()`

Lists information about the OpenID Connect providers in the AWS account.

Request Syntax

```
response = client.list_open_id_connect_providers()
```

Return type dict

Returns

Response Syntax

```
{
    'OpenIDConnectProviderList': [
        {
            'Arn': 'string'
        },
    ]
}
```

Response Structure

- *(dict)* –

Contains the response to a successful ListOpenIDConnectProviders request.

- **OpenIDConnectProviderList** (*list*) –

The list of IAM OpenID Connect providers in the AWS account.

- * *(dict)* –

Contains the Amazon Resource Name (ARN) for an IAM OpenID Connect provider.

- **Arn** (*string*) –

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference* .

list_policies (***kwargs*)

Lists all the managed policies that are available to your account, including your own customer managed policies and all AWS managed policies.

You can filter the list of policies that is returned using the optional `OnlyAttached` , `Scope` , and `PathPrefix` parameters. For example, to list only the customer managed policies in your AWS account, set `Scope` to `Local` . To list only AWS managed policies, set `Scope` to `AWS` .

You can paginate the results using the `MaxItems` and `Marker` parameters.

For more information about managed policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide* .

Request Syntax

```
response = client.list_policies(
    Scope='All'|'AWS'|'Local',
    OnlyAttached=True|False,
    PathPrefix='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **Scope** (*string*) – The scope to use for filtering the results.

To list only AWS managed policies, set `Scope` to `AWS` . To list only the customer managed policies in your AWS account, set `Scope` to `Local` .

This parameter is optional. If it is not included, or if it is set to `All` , all policies are returned.

- **OnlyAttached** (*boolean*) – A flag to filter the results to only the attached policies.

When `OnlyAttached` is `true`, the returned list contains only the policies that are attached to a user, group, or role. When `OnlyAttached` is `false`, or when the parameter is not included, all policies are returned.

- **PathPrefix** (*string*) – The path prefix for filtering the results. This parameter is optional. If it is not included, it defaults to a slash (/), listing all policies.
- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns

Response Syntax

```
{
  'Policies': [
    {
      'PolicyName': 'string',
      'PolicyId': 'string',
      'Arn': 'string',
      'Path': 'string',
      'DefaultVersionId': 'string',
      'AttachmentCount': 123,
      'IsAttachable': True|False,
      'Description': 'string',
      'CreateDate': datetime(2015, 1, 1),
      'UpdateDate': datetime(2015, 1, 1)
    },
  ],
  'IsTruncated': True|False,
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `ListPolicies` request.

– **Policies** (*list*) –

A list of policies.

* (*dict*) –

Contains information about a managed policy.

This data type is used as a response element in the `CreatePolicy`, `GetPolicy`, and `ListPolicies` actions.

For more information about managed policies, refer to [Managed Policies and Inline Policies](#) in the *Using IAM* guide.

- **PolicyName** (*string*) –

The friendly name (not ARN) identifying the policy.

- **PolicyId** (*string*) –

The stable and unique string identifying the policy.

For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- **Arn** (*string*) –

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

- **Path** (*string*) –

The path to the policy.

For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **DefaultVersionId** (*string*) –

The identifier for the version of the policy that is set as the default version.

- **AttachmentCount** (*integer*) –

The number of entities (users, groups, and roles) that the policy is attached to.

- **IsAttachable** (*boolean*) –

Specifies whether the policy can be attached to an IAM user, group, or role.

- **Description** (*string*) –

A friendly description of the policy.

This element is included in the response to the `GetPolicy` operation. It is not included in the response to the `ListPolicies` operation.

- **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the policy was created.

- **UpdateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the policy was last updated.

When a policy has only one version, this field contains the date and time when the policy was created. When a policy has more than one version, this field contains the date and time when the most recent policy version was created.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might

return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

– **Marker** (*string*) –

When `IsTruncated` is `true`, this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

list_policy_versions (***kwargs*)

Lists information about the versions of the specified managed policy, including the version that is set as the policy's default version.

For more information about managed policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = client.list_policy_versions(
    PolicyArn='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

• **PolicyArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns

Response Syntax

```
{
  'Versions': [
    {
      'Document': 'string',
      'VersionId': 'string',
      'IsDefaultVersion': True|False,
      'CreateDate': datetime(2015, 1, 1)
    },
  ],
}
```

```
'IsTruncated': True|False,  
'Marker': 'string'  
}
```

Response Structure

- *(dict)* –

Contains the response to a successful ListPolicyVersions request.

- **Versions** (*list*) –

A list of policy versions.

For more information about managed policy versions, see [Versioning for Managed Policies](#) in the *IAM User Guide* .

- * *(dict)* –

Contains information about a version of a managed policy.

This data type is used as a response element in the CreatePolicyVersion , GetPolicyVersion , ListPolicyVersions , and GetAccountAuthorizationDetails actions.

For more information about managed policies, refer to [Managed Policies and Inline Policies](#) in the *Using IAM* guide.

- **Document** (*string*) –

The policy document.

The policy document is returned in the response to the GetPolicyVersion and GetAccountAuthorizationDetails operations. It is not returned in the response to the CreatePolicyVersion or ListPolicyVersions operations.

- **VersionId** (*string*) –

The identifier for the policy version.

Policy version identifiers always begin with v (always lower-case). When a policy is created, the first policy version is v1 .

- **IsDefaultVersion** (*boolean*) –

Specifies whether the policy version is set as the policy’s default version.

- **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#) , when the policy version was created.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the Marker request parameter to retrieve more items. Note that IAM might return fewer than the MaxItems number of results even when there are more results available. We recommend that you check IsTruncated after every call to ensure that you receive all of your results.

- **Marker** (*string*) –

When IsTruncated is true , this element is present and contains the value to use for the Marker parameter in a subsequent pagination request.

list_role_policies (**kwargs)

Lists the names of the inline policies that are embedded in the specified role.

A role can also have managed policies attached to it. To list the managed policies that are attached to a role, use `ListAttachedRolePolicies`. For more information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

You can paginate the results using the `MaxItems` and `Marker` parameters. If there are no inline policies embedded with the specified role, the action returns an empty list.

Request Syntax

```
response = client.list_role_policies(
    RoleName='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **RoleName** (*string*) – [REQUIRED]

The name of the role to list policies for.

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns**Response Syntax**

```
{
    'PolicyNames': [
        'string',
    ],
    'IsTruncated': True|False,
    'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `ListRolePolicies` request.

- **PolicyNames** (*list*) –

A list of policy names.

- (*string*) –

– **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

– **Marker** (*string*) –

When `IsTruncated` is `true`, this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

list_roles (***kwargs*)

Lists the roles that have the specified path prefix. If there are none, the action returns an empty list. For more information about roles, go to [Working with Roles](#).

You can paginate the results using the `MaxItems` and `Marker` parameters.

Request Syntax

```
response = client.list_roles(
    PathPrefix='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **PathPrefix** (*string*) – The path prefix for filtering the results. For example, the prefix `/application_abc/component_xyz/` gets all roles whose path starts with `/application_abc/component_xyz/`.

This parameter is optional. If it is not included, it defaults to a slash (`/`), listing all roles.

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns

Response Syntax

```
{
  'Roles': [
    {
      'Path': 'string',
      'RoleName': 'string',
```

```

        'RoleId': 'string',
        'Arn': 'string',
        'CreateDate': datetime(2015, 1, 1),
        'AssumeRolePolicyDocument': 'string'
    },
],
'IsTruncated': True|False,
'Marker': 'string'
}

```

Response Structure

- (*dict*) –

Contains the response to a successful ListRoles request.

- **Roles** (*list*) –

A list of roles.

- * (*dict*) –

Contains information about an IAM role.

This data type is used as a response element in the following actions:

- CreateRole
- GetRole
- ListRoles

- **Path** (*string*) –

The path to the role. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **RoleName** (*string*) –

The friendly name that identifies the role.

- **RoleId** (*string*) –

The stable and unique string identifying the role. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) specifying the role. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the role was created.

- **AssumeRolePolicyDocument** (*string*) –

The policy that grants an entity permission to assume the role.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

- **Marker** (*string*) –

When `IsTruncated` is `true`, this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

`list_saml_providers()`

Lists the SAML providers in the account.

Note: This operation requires [Signature Version 4](#).

Request Syntax

```
response = client.list_saml_providers()
```

Return type dict

Returns

Response Syntax

```
{
  'SAMLProviderList': [
    {
      'Arn': 'string',
      'ValidUntil': datetime(2015, 1, 1),
      'CreateDate': datetime(2015, 1, 1)
    },
  ]
}
```

Response Structure

- (*dict*) –

Contains the response to a successful ListSAMLProviders request.

- **SAMLProviderList** (*list*) –

The list of SAML providers for this account.

- * (*dict*) –

Contains the list of SAML providers for this account.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) of the SAML provider.

- **ValidUntil** (*datetime*) –

The expiration date and time for the SAML provider.

- **CreateDate** (*datetime*) –

The date and time when the SAML provider was created.

`list_server_certificates(**kwargs)`

Lists the server certificates that have the specified path prefix. If none exist, the action returns an empty list.

You can paginate the results using the `MaxItems` and `Marker` parameters.

Request Syntax

```
response = client.list_server_certificates(
    PathPrefix='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **PathPrefix** (*string*) – The path prefix for filtering the results. For example: `/company/servercerts` would get all server certificates for which the path starts with `/company/servercerts`.

This parameter is optional. If it is not included, it defaults to a slash (`/`), listing all server certificates.

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns**Response Syntax**

```
{
  'ServerCertificateMetadataList': [
    {
      'Path': 'string',
      'ServerCertificateName': 'string',
      'ServerCertificateId': 'string',
      'Arn': 'string',
      'UploadDate': datetime(2015, 1, 1),
      'Expiration': datetime(2015, 1, 1)
    },
  ],
  'IsTruncated': True|False,
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `ListServerCertificates` request.

- **ServerCertificateMetadataList** (*list*) –

A list of server certificates.

- * (*dict*) –

Contains information about a server certificate without its certificate body, certificate chain, and private key.

This data type is used as a response element in the `UploadServerCertificate` and `ListServerCertificates` actions.

- **Path** (*string*) –

The path to the server certificate. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **ServerCertificateName** (*string*) –
The name that identifies the server certificate.
- **ServerCertificateId** (*string*) –
The stable and unique string identifying the server certificate.
For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.
- **Arn** (*string*) –
The Amazon Resource Name (ARN) specifying the server certificate. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.
- **UploadDate** (*datetime*) –
The date when the server certificate was uploaded.
- **Expiration** (*datetime*) –
The date on which the certificate is set to expire.
- **IsTruncated** (*boolean*) –
A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.
- **Marker** (*string*) –
When `IsTruncated` is `true`, this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

list_signing_certificates (***kwargs*)

Returns information about the signing certificates associated with the specified user. If there are none, the action returns an empty list.

Although each user is limited to a small number of signing certificates, you can still paginate the results using the `MaxItems` and `Marker` parameters.

If the `UserName` field is not specified, the user name is determined implicitly based on the AWS access key ID used to sign the request. Because this action works for access keys under the AWS account, you can use this action to manage root credentials even if the AWS account has no associated users.

Request Syntax

```
response = client.list_signing_certificates(
    UserName='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **UserName** (*string*) – The name of the user.
- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items be-

yond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns

Response Syntax

```
{
  'Certificates': [
    {
      'UserName': 'string',
      'CertificateId': 'string',
      'CertificateBody': 'string',
      'Status': 'Active'|'Inactive',
      'UploadDate': datetime(2015, 1, 1)
    },
  ],
  'IsTruncated': True|False,
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `ListSigningCertificates` request.

- **Certificates** (*list*) –

A list of the user's signing certificate information.

- * (*dict*) –

Contains information about an X.509 signing certificate.

This data type is used as a response element in the `UploadSigningCertificate` and `ListSigningCertificates` actions.

- **UserName** (*string*) –

The name of the user the signing certificate is associated with.

- **CertificateId** (*string*) –

The ID for the signing certificate.

- **CertificateBody** (*string*) –

The contents of the signing certificate.

- **Status** (*string*) –

The status of the signing certificate. `Active` means the key is valid for API calls, while `Inactive` means it is not.

- **UploadDate** (*datetime*) –

The date when the signing certificate was uploaded.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might

return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

– **Marker** (*string*) –

When `IsTruncated` is `true`, this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

list_ssh_public_keys (***kwargs*)

Returns information about the SSH public keys associated with the specified IAM user. If there are none, the action returns an empty list.

The SSH public keys returned by this action are used only for authenticating the IAM user to an AWS CodeCommit repository. For more information about using SSH keys to authenticate to an AWS CodeCommit repository, see [Set up AWS CodeCommit for SSH Connections](#) in the *AWS CodeCommit User Guide*.

Although each user is limited to a small number of keys, you can still paginate the results using the `MaxItems` and `Marker` parameters.

Request Syntax

```
response = client.list_ssh_public_keys(
    UserName='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **UserName** (*string*) – The name of the IAM user to list SSH public keys for. If none is specified, the `UserName` field is determined implicitly based on the AWS access key used to sign the request.
- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns

Response Syntax

```
{
  'SSHPublicKeys': [
    {
      'UserName': 'string',
      'SSHPublicKeyId': 'string',
      'Status': 'Active'|'Inactive',
      'UploadDate': datetime(2015, 1, 1)
```

```

    },
    ],
    'IsTruncated': True|False,
    'Marker': 'string'
}

```

Response Structure

- *(dict)* –

Contains the response to a successful ListSSHPublicKeys request.

- **SSHPublicKeys** (*list*) –

A list of SSH public keys.

- * *(dict)* –

Contains information about an SSH public key, without the key's body or fingerprint.

This data type is used as a response element in the ListSSHPublicKeys action.

- **UserName** (*string*) –

The name of the IAM user associated with the SSH public key.

- **SSHPublicKeyId** (*string*) –

The unique identifier for the SSH public key.

- **Status** (*string*) –

The status of the SSH public key. `Active` means the key can be used for authentication with an AWS CodeCommit repository. `Inactive` means the key cannot be used.

- **UploadDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the SSH public key was uploaded.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

- **Marker** (*string*) –

When `IsTruncated` is `true`, this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

list_user_policies (***kwargs*)

Lists the names of the inline policies embedded in the specified user.

A user can also have managed policies attached to it. To list the managed policies that are attached to a user, use `ListAttachedUserPolicies`. For more information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

You can paginate the results using the `MaxItems` and `Marker` parameters. If there are no inline policies embedded with the specified user, the action returns an empty list.

Request Syntax

```
response = client.list_user_policies(
    UserName='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **UserName** (*string*) – [REQUIRED]

The name of the user to list policies for.

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns

Response Syntax

```
{
  'PolicyNames': [
    'string',
  ],
  'IsTruncated': True|False,
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `ListUserPolicies` request.

- **PolicyNames** (*list*) –

A list of policy names.

* (*string*) –

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

- **Marker** (*string*) –

When `IsTruncated` is `true`, this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

list_users (**kwargs)

Lists the IAM users that have the specified path prefix. If no path prefix is specified, the action returns all users in the AWS account. If there are none, the action returns an empty list.

You can paginate the results using the `MaxItems` and `Marker` parameters.

Request Syntax

```
response = client.list_users(
    PathPrefix='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **PathPrefix** (*string*) – The path prefix for filtering the results. For example: `/division_abc/subdivision_xyz/`, which would get all user names whose path starts with `/division_abc/subdivision_xyz/`.

This parameter is optional. If it is not included, it defaults to a slash (/), listing all user names.

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns**Response Syntax**

```
{
  'Users': [
    {
      'Path': 'string',
      'UserName': 'string',
      'UserId': 'string',
      'Arn': 'string',
      'CreateDate': datetime(2015, 1, 1),
      'PasswordLastUsed': datetime(2015, 1, 1)
    },
  ],
  'IsTruncated': True|False,
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `ListUsers` request.

– **Users** (*list*) –

A list of users.

* (*dict*) –

Contains information about an IAM user entity.

This data type is used as a response element in the following actions:

- CreateUser
- GetUser
- ListUsers

· **Path** (*string*) –

The path to the user. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

· **UserName** (*string*) –

The friendly name identifying the user.

· **UserId** (*string*) –

The stable and unique string identifying the user. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

· **Arn** (*string*) –

The Amazon Resource Name (ARN) that identifies the user. For more information about ARNs and how to use ARNs in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

· **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the user was created.

· **PasswordLastUsed** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the user's password was last used to sign in to an AWS website. For a list of AWS websites that capture a user's last sign-in time, see the [Credential Reports](#) topic in the *Using IAM* guide. If a password is used more than once in a five-minute span, only the first use is returned in this field. This field is null (not present) when:

- The user does not have a password
- The password exists but has never been used (at least not since IAM started tracking this information on October 20th, 2014)
- there is no sign-in data associated with the user

This value is returned only in the GetUser and ListUsers actions.

– **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

– **Marker** (*string*) –

When `IsTruncated` is `true`, this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

`list_virtual_mfa_devices` (**kwargs)

Lists the virtual MFA devices under the AWS account by assignment status. If you do not specify an assignment status, the action returns a list of all virtual MFA devices. Assignment status can be `Assigned`, `Unassigned`, or `Any`.

You can paginate the results using the `MaxItems` and `Marker` parameters.

Request Syntax

```
response = client.list_virtual_mfa_devices(
    AssignmentStatus='Assigned'|'Unassigned'|'Any',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **AssignmentStatus** (*string*) – The status (unassigned or assigned) of the devices to list. If you do not specify an `AssignmentStatus`, the action defaults to `Any` which lists both assigned and unassigned virtual MFA devices.
- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type dict

Returns

Response Syntax

```
{
  'VirtualMFADevices': [
    {
      'SerialNumber': 'string',
      'Base32StringSeed': b'bytes',
      'QRCodePNG': b'bytes',
      'User': {
        'Path': 'string',
        'UserName': 'string',
        'UserId': 'string',
        'Arn': 'string',
        'CreateDate': datetime(2015, 1, 1),
        'PasswordLastUsed': datetime(2015, 1, 1)
      },
      'EnableDate': datetime(2015, 1, 1)
    },
  ],
}
```

```
'IsTruncated': True|False,  
'Marker': 'string'  
}
```

Response Structure

- (*dict*) –

Contains the response to a successful ListVirtualMFADevices request.

- **VirtualMFADevices** (*list*) –

The list of virtual MFA devices in the current account that match the AssignmentStatus value that was passed in the request.

- * (*dict*) –

Contains information about a virtual MFA device.

- **SerialNumber** (*string*) –

The serial number associated with VirtualMFADevice .

- **Base32StringSeed** (*bytes*) –

The Base32 seed defined as specified in [RFC3548](#) . The Base32StringSeed is Base64-encoded.

- **QRCodePNG** (*bytes*) –

A QR code PNG image that encodes otpauth://totp/\$virtualMFADeviceName@\$AccountName?secret= where \$virtualMFADeviceName is one of the create call arguments, AccountName is the user name if set (otherwise, the account ID otherwise), and Base32String is the seed in Base32 format. The Base32String value is Base64-encoded.

- **User** (*dict*) –

Contains information about an IAM user entity.

This data type is used as a response element in the following actions:

- CreateUser
- GetUser
- ListUsers
- **Path** (*string*) –

The path to the user. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **UserName** (*string*) –

The friendly name identifying the user.

- **UserId** (*string*) –

The stable and unique string identifying the user. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) that identifies the user. For more information about ARNs and how to use ARNs in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#) , when the user was created.

- **PasswordLastUsed** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#) , when the user’s password was last used to sign in to an AWS website. For a list of AWS websites that capture a user’s last sign-in time, see the [Credential Reports](#) topic in the *Using IAM* guide. If a password is used more than once in a five-minute span, only the first use is returned in this field. This field is null (not present) when:

- The user does not have a password
- The password exists but has never been used (at least not since IAM started tracking this information on October 20th, 2014)
- there is no sign-in data associated with the user

This value is returned only in the `GetUser` and `ListUsers` actions.

- **EnableDate** (*datetime*) –

The date and time on which the virtual MFA device was enabled.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

- **Marker** (*string*) –

When `IsTruncated` is `true` , this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

put_group_policy (***kwargs*)

Adds (or updates) an inline policy document that is embedded in the specified group.

A user can also have managed policies attached to it. To attach a managed policy to a group, use `AttachGroupPolicy` . To create a new managed policy, use `CreatePolicy` . For information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide* .

For information about limits on the number of inline policies that you can embed in a group, see [Limitations on IAM Entities](#) in the *IAM User Guide* .

Note: Because policy documents can be large, you should use POST rather than GET when calling `PutGroupPolicy` . For general information about using the Query API with IAM, go to [Making Query Requests](#) in the *Using IAM* guide.

Request Syntax

```
response = client.put_group_policy(
    GroupName='string',
    PolicyName='string',
    PolicyDocument='string'
)
```

Parameters

- **GroupName** (*string*) – [REQUIRED]
The name of the group to associate the policy with.
- **PolicyName** (*string*) – [REQUIRED]
The name of the policy document.
- **PolicyDocument** (*string*) – [REQUIRED]
The policy document.

Returns None

put_role_policy (***kwargs*)

Adds (or updates) an inline policy document that is embedded in the specified role.

When you embed an inline policy in a role, the inline policy is used as the role's access (permissions) policy. The role's trust policy is created at the same time as the role, using `CreateRole` . You can update a role's trust policy using `UpdateAssumeRolePolicy` . For more information about roles, go to [Using Roles to Delegate Permissions and Federate Identities](#) .

A role can also have a managed policy attached to it. To attach a managed policy to a role, use `AttachRolePolicy` . To create a new managed policy, use `CreatePolicy` . For information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide* .

For information about limits on the number of inline policies that you can embed with a role, see [Limitations on IAM Entities](#) in the *IAM User Guide* .

Note: Because policy documents can be large, you should use POST rather than GET when calling `PutRolePolicy` . For general information about using the Query API with IAM, go to [Making Query Requests](#) in the *Using IAM* guide.

Request Syntax

```
response = client.put_role_policy(  
    RoleName='string',  
    PolicyName='string',  
    PolicyDocument='string'  
)
```

Parameters

- **RoleName** (*string*) – [REQUIRED]
The name of the role to associate the policy with.
- **PolicyName** (*string*) – [REQUIRED]
The name of the policy document.
- **PolicyDocument** (*string*) – [REQUIRED]
The policy document.

Returns None

put_user_policy (***kwargs*)

Adds (or updates) an inline policy document that is embedded in the specified user.

A user can also have a managed policy attached to it. To attach a managed policy to a user, use `AttachUserPolicy` . To create a new managed policy, use `CreatePolicy` . For information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide* .

For information about limits on the number of inline policies that you can embed in a user, see [Limitations on IAM Entities](#) in the *IAM User Guide* .

Note: Because policy documents can be large, you should use POST rather than GET when calling

`PutUserPolicy` . For general information about using the Query API with IAM, go to [Making Query Requests](#) in the *Using IAM* guide.

Request Syntax

```
response = client.put_user_policy(
    UserName='string',
    PolicyName='string',
    PolicyDocument='string'
)
```

Parameters

- **UserName** (*string*) – [REQUIRED]
The name of the user to associate the policy with.
- **PolicyName** (*string*) – [REQUIRED]
The name of the policy document.
- **PolicyDocument** (*string*) – [REQUIRED]
The policy document.

Returns None

remove_client_id_from_open_id_connect_provider (***kwargs*)

Removes the specified client ID (also known as audience) from the list of client IDs registered for the specified IAM OpenID Connect provider.

This action is idempotent; it does not fail or return an error if you try to remove a client ID that was removed previously.

Request Syntax

```
response = client.remove_client_id_from_open_id_connect_provider(
    OpenIDConnectProviderArn='string',
    ClientID='string'
)
```

Parameters

- **OpenIDConnectProviderArn** (*string*) – [REQUIRED]
The Amazon Resource Name (ARN) of the IAM OpenID Connect (OIDC) provider to remove the client ID from. You can get a list of OIDC provider ARNs by using the `ListOpenIDConnectProviders` action.
- **ClientID** (*string*) – [REQUIRED]
The client ID (also known as audience) to remove from the IAM OpenID Connect provider. For more information about client IDs, see `CreateOpenIDConnectProvider` .

Returns None

remove_role_from_instance_profile (***kwargs*)

Removes the specified role from the specified instance profile.

Warning: Make sure you do not have any Amazon EC2 instances running with the role you are about to remove from the instance profile. Removing a role from an instance profile that is associated with a running instance will break any applications running on the instance.

For more information about roles, go to [Working with Roles](#) . For more information about instance profiles, go to [About Instance Profiles](#) .

Request Syntax

```
response = client.remove_role_from_instance_profile(  
    InstanceProfileName='string',  
    RoleName='string'  
)
```

Parameters

- **InstanceProfileName** (*string*) – [REQUIRED]

The name of the instance profile to update.

- **RoleName** (*string*) – [REQUIRED]

The name of the role to remove.

Returns None

remove_user_from_group (***kwargs*)

Removes the specified user from the specified group.

Request Syntax

```
response = client.remove_user_from_group(  
    GroupName='string',  
    UserName='string'  
)
```

Parameters

- **GroupName** (*string*) – [REQUIRED]

The name of the group to update.

- **UserName** (*string*) – [REQUIRED]

The name of the user to remove.

Returns None

resync_mfa_device (***kwargs*)

Synchronizes the specified MFA device with AWS servers.

For more information about creating and working with virtual MFA devices, go to [Using a Virtual MFA Device](#) in the *Using IAM* guide.

Request Syntax

```
response = client.resync_mfa_device(  
    UserName='string',  
    SerialNumber='string',  
    AuthenticationCode1='string',  
    AuthenticationCode2='string'  
)
```

Parameters

- **UserName** (*string*) – [REQUIRED]

The name of the user whose MFA device you want to resynchronize.

- **SerialNumber** (*string*) – [REQUIRED]

Serial number that uniquely identifies the MFA device.

- **AuthenticationCode1** (*string*) – [REQUIRED]

An authentication code emitted by the device.

- **AuthenticationCode2** (*string*) – [REQUIRED]

A subsequent authentication code emitted by the device.

Returns None

set_default_policy_version (***kwargs*)

Sets the specified version of the specified policy as the policy’s default (operative) version.

This action affects all users, groups, and roles that the policy is attached to. To list the users, groups, and roles that the policy is attached to, use the `ListEntitiesForPolicy` API.

For information about managed policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = client.set_default_policy_version(
    PolicyArn='string',
    VersionId='string'
)
```

Parameters

- **PolicyArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

- **VersionId** (*string*) – [REQUIRED]

The version of the policy to set as the default (operative) version.

For more information about managed policy versions, see [Versioning for Managed Policies](#) in the *IAM User Guide*.

Returns None

simulate_custom_policy (***kwargs*)

Simulate a set of IAM policies against a list of API actions and AWS resources to determine the policies’ effective permissions. The policies are provided as a list of strings.

The simulation does not perform the API actions, it only checks the authorization to determine if the simulated policies allow or deny the actions.

If you want to simulate existing policies attached to an IAM user, group, or role, use `SimulatePrincipalPolicy` instead.

Context keys are variables maintained by AWS and its services that provide details about the context of an API query request, and can be evaluated by using the `Condition` element of an IAM policy. To get the list of context keys required by the policies to simulate them correctly, use `GetContextKeysForCustomPolicy`.

If the output is long, you can paginate the results using the `MaxItems` and `Marker` parameters.

Request Syntax

```
response = client.simulate_custom_policy(
    PolicyInputList=[
        'string',
    ],
    ActionNames=[
        'string',
    ]
)
```

```
],
ResourceArns=[
    'string',
],
ContextEntries=[
    {
        'ContextKeyName': 'string',
        'ContextKeyValues': [
            'string',
        ],
        'ContextKeyType': 'string'|'stringList'|'numeric'|'numericList'|'boolean'|'booleanList'
    },
],
MaxItems=123,
Marker='string'
)
```

Parameters

- **PolicyInputList** (*list*) – [REQUIRED]

A list of policy documents to include in the simulation. Each document is specified as a string containing the complete, valid JSON text of an IAM policy.

– (*string*) –

- **ActionNames** (*list*) – [REQUIRED]

A list of names of API actions to evaluate in the simulation. Each action is evaluated for each resource. Each action must include the service identifier, such as `iam:CreateUser`.

– (*string*) –

- **ResourceArns** (*list*) – A list of ARNs of AWS resources to include in the simulation. If this parameter is not provided then the value defaults to `*` (all resources). Each API in the **ActionNames** parameter is evaluated for each resource in this list. The simulation determines the access result (allowed or denied) of each combination and reports it in the response.

– (*string*) –

- **ContextEntries** (*list*) – A list of context keys and corresponding values that are used by the simulation. Whenever a context key is evaluated by a **Condition** element in one of the simulated IAM permission policies, the corresponding value is supplied.

– (*dict*) –

Contains information about a condition context key. It includes the name of the key and specifies the value (or values, if the context key supports multiple values) to use in the simulation. This information is used when evaluating the **Condition** elements of the input policies.

This data type is used as an input parameter to “`SimulatePolicy`”.

- * **ContextKeyName** (*string*) –

The full name of a condition context key, including the service prefix. For example, `aws:SourceIp` or `s3:VersionId`.

- * **ContextKeyValues** (*list*) –

The value (or values, if the condition context key supports multiple values) to provide to the simulation for use when the key is referenced by a **Condition** element in an input policy.

· (*string*) –

- * **ContextKeyType** (*string*) –

The data type of the value (or values) specified in the `ContextKeyValues` parameter.

- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.

Return type dict

Returns

Response Syntax

```
{
  'EvaluationResults': [
    {
      'EvalActionName': 'string',
      'EvalResourceName': 'string',
      'EvalDecision': 'allowed'|'explicitDeny'|'implicitDeny',
      'MatchedStatements': [
        {
          'SourcePolicyId': 'string',
          'SourcePolicyType': 'user'|'group'|'role'|'aws-managed'|'user-',
          'StartPosition': {
            'Line': 123,
            'Column': 123
          },
          'EndPosition': {
            'Line': 123,
            'Column': 123
          }
        },
      ],
      'MissingContextValues': [
        'string',
      ]
    },
  ],
  'IsTruncated': True|False,
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `SimulatePrincipalPolicy` or `SimulateCustomPolicy` request.

- **EvaluationResults** (*list*) –

The results of the simulation.

* (*dict*) –

Contains the results of a simulation.

This data type is used by the return parameter of “`SimulatePolicy`”

- **EvalActionName** (*string*) –

The name of the API action tested on the indicated resource.

- **EvalResourceName** (*string*) –

The ARN of the resource that the indicated API action was tested on.

- **EvalDecision** (*string*) –

The result of the simulation.

- **MatchedStatements** (*list*) –

A list of the statements in the input policies that determine the result for this scenario. Remember that even if multiple statements allow the action on the resource, if only one statement denies that action, then the explicit deny overrides any allow, and the deny statement is the only entry included in the result.

- (*dict*) –

Contains a reference to a `Statement` element in a policy document that determines the result of the simulation.

This data type is used by the `MatchedStatements` member of the “`EvaluationResult`” type.

- **SourcePolicyId** (*string*) –

The identifier of the policy that was provided as an input.

- **SourcePolicyType** (*string*) –

The type of the policy.

- **StartPosition** (*dict*) –

The row and column of the beginning of the `Statement` in an IAM policy.

- **Line** (*integer*) –

The line containing the specified position in the document.

- **Column** (*integer*) –

The column in the line containing the specified position in the document.

- **EndPosition** (*dict*) –

The row and column of the end of a `Statement` in an IAM policy.

- **Line** (*integer*) –

The line containing the specified position in the document.

- **Column** (*integer*) –

The column in the line containing the specified position in the document.

- **MissingContextValues** (*list*) –

A list of context keys that are required by the included input policies but that were not provided by one of the input param-

eters. To discover the context keys used by a set of policies, you can call `GetContextKeysForCustomPolicy` or `GetContextKeysForPrincipalPolicy`.

If the response includes any keys in this list, then the reported results might be untrustworthy because the simulation could not completely evaluate all of the conditions specified in the policies that would occur in a real world request.

· *(string)* –

– **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

– **Marker** (*string*) –

When `IsTruncated` is `true`, this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

simulate_principal_policy (***kwargs*)

Simulate the set of IAM policies attached to an IAM entity against a list of API actions and AWS resources to determine the policies' effective permissions. The entity can be an IAM user, group, or role. If you specify a user, then the simulation also includes all of the policies attached to groups that the user is a member of.

You can optionally include a list of one or more additional policies specified as strings to include in the simulation. If you want to simulate only policies specified as strings, use `SimulateCustomPolicy` instead.

The simulation does not perform the API actions, it only checks the authorization to determine if the simulated policies allow or deny the actions.

Note: This API discloses information about the permissions granted to other users. If you do not want users to see other user's permissions, then consider allowing them to use `SimulateCustomPolicy` instead.

Context keys are variables maintained by AWS and its services that provide details about the context of an API query request, and can be evaluated by using the `Condition` element of an IAM policy. To get the list of context keys required by the policies to simulate them correctly, use `GetContextKeysForPrincipalPolicy`.

If the output is long, you can paginate the results using the `MaxItems` and `Marker` parameters.

Request Syntax

```
response = client.simulate_principal_policy(
    PolicySourceArn='string',
    PolicyInputList=[
        'string',
    ],
    ActionNames=[
        'string',
    ],
    ResourceArns=[
        'string',
    ],
    ContextEntries=[
        {
            'ContextKeyName': 'string',
```

```
        'ContextKeyValues': [
            'string',
        ],
        'ContextKeyType': 'string'|'stringList'|'numeric'|'numericList'|'boolean'|'booleanList',
    },
    ],
    MaxItems=123,
    Marker='string'
)
```

Parameters

- **PolicySourceArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of a user, group, or role whose policies you want to include in the simulation. If you specify a user, group, or role, the simulation includes all policies associated with that entity. If you specify a user, the simulation also includes all policies attached to any groups the user is a member of.

- **PolicyInputList** (*list*) – An optional list of additional policy documents to include in the simulation. Each document is specified as a string containing the complete, valid JSON text of an IAM policy.

- (*string*) –

- **ActionNames** (*list*) – [REQUIRED]

A list of names of API actions to evaluate in the simulation. Each action is evaluated for each resource. Each action must include the service identifier, such as `iam:CreateUser`.

- (*string*) –

- **ResourceArns** (*list*) – A list of ARNs of AWS resources to include in the simulation. If this parameter is not provided then the value defaults to `*` (all resources). Each API in the `ActionNames` parameter is evaluated for each resource in this list. The simulation determines the access result (allowed or denied) of each combination and reports it in the response.

- (*string*) –

- **ContextEntries** (*list*) – A list of context keys and corresponding values that are used by the simulation. Whenever a context key is evaluated by a `Condition` element in one of the simulated IAM permission policies, the corresponding value is supplied.

- (*dict*) –

Contains information about a condition context key. It includes the name of the key and specifies the value (or values, if the context key supports multiple values) to use in the simulation. This information is used when evaluating the `Condition` elements of the input policies.

This data type is used as an input parameter to “`SimulatePolicy`”.

- * **ContextKeyName** (*string*) –

The full name of a condition context key, including the service prefix. For example, `aws:SourceIp` or `s3:VersionId`.

- * **ContextKeyValues** (*list*) –

The value (or values, if the condition context key supports multiple values) to provide to the simulation for use when the key is referenced by a `Condition` element in an input policy.

- (*string*) –

- * **ContextKeyType** (*string*) –

The data type of the value (or values) specified in the `ContextKeyValues` parameter.

- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.

Return type dict

Returns

Response Syntax

```
{
  'EvaluationResults': [
    {
      'EvalActionName': 'string',
      'EvalResourceName': 'string',
      'EvalDecision': 'allowed'|'explicitDeny'|'implicitDeny',
      'MatchedStatements': [
        {
          'SourcePolicyId': 'string',
          'SourcePolicyType': 'user'|'group'|'role'|'aws-managed'|'user-',
          'StartPosition': {
            'Line': 123,
            'Column': 123
          },
          'EndPosition': {
            'Line': 123,
            'Column': 123
          }
        },
      ],
      'MissingContextValues': [
        'string',
      ]
    },
  ],
  'IsTruncated': True|False,
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `SimulatePrincipalPolicy` or `SimulateCustomPolicy` request.

- **EvaluationResults** (*list*) –

The results of the simulation.

* (*dict*) –

Contains the results of a simulation.

This data type is used by the return parameter of “`SimulatePolicy`”

- **EvalActionName** (*string*) –

The name of the API action tested on the indicated resource.

- **EvalResourceName** (*string*) –

The ARN of the resource that the indicated API action was tested on.

- **EvalDecision** (*string*) –

The result of the simulation.

- **MatchedStatements** (*list*) –

A list of the statements in the input policies that determine the result for this scenario. Remember that even if multiple statements allow the action on the resource, if only one statement denies that action, then the explicit deny overrides any allow, and the deny statement is the only entry included in the result.

- (*dict*) –

Contains a reference to a `Statement` element in a policy document that determines the result of the simulation.

This data type is used by the `MatchedStatements` member of the “`EvaluationResult`” type.

- **SourcePolicyId** (*string*) –

The identifier of the policy that was provided as an input.

- **SourcePolicyType** (*string*) –

The type of the policy.

- **StartPosition** (*dict*) –

The row and column of the beginning of the `Statement` in an IAM policy.

- **Line** (*integer*) –

The line containing the specified position in the document.

- **Column** (*integer*) –

The column in the line containing the specified position in the document.

- **EndPosition** (*dict*) –

The row and column of the end of a `Statement` in an IAM policy.

- **Line** (*integer*) –

The line containing the specified position in the document.

- **Column** (*integer*) –

The column in the line containing the specified position in the document.

- **MissingContextValues** (*list*) –

A list of context keys that are required by the included input policies but that were not provided by one of the input param-

eters. To discover the context keys used by a set of policies, you can call `GetContextKeysForCustomPolicy` or `GetContextKeysForPrincipalPolicy`.

If the response includes any keys in this list, then the reported results might be untrustworthy because the simulation could not completely evaluate all of the conditions specified in the policies that would occur in a real world request.

· *(string)* –

– **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

– **Marker** (*string*) –

When `IsTruncated` is `true`, this element is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

update_access_key (***kwargs*)

Changes the status of the specified access key from Active to Inactive, or vice versa. This action can be used to disable a user's key as part of a key rotation work flow.

If the `UserName` field is not specified, the `UserName` is determined implicitly based on the AWS access key ID used to sign the request. Because this action works for access keys under the AWS account, you can use this action to manage root credentials even if the AWS account has no associated users.

For information about rotating keys, see [Managing Keys and Certificates](#) in the *IAM User Guide*.

Request Syntax

```
response = client.update_access_key(
    UserName='string',
    AccessKeyId='string',
    Status='Active'|'Inactive'
)
```

Parameters

- **UserName** (*string*) – The name of the user whose key you want to update.
- **AccessKeyId** (*string*) – **[REQUIRED]**

The access key ID of the secret access key you want to update.

- **Status** (*string*) – **[REQUIRED]**

The status you want to assign to the secret access key. `Active` means the key can be used for API calls to AWS, while `Inactive` means the key cannot be used.

Returns None

update_account_password_policy (***kwargs*)

Updates the password policy settings for the AWS account.

Note: This action does not support partial updates. No parameters are required, but if you do not specify a parameter, that parameter's value reverts to its default value. See the **Request Parameters** section for each parameter's default value.

For more information about using a password policy, see [Managing an IAM Password Policy](#) in the *IAM User Guide* .

Request Syntax

```
response = client.update_account_password_policy(
    MinimumPasswordLength=123,
    RequireSymbols=True|False,
    RequireNumbers=True|False,
    RequireUppercaseCharacters=True|False,
    RequireLowercaseCharacters=True|False,
    AllowUsersToChangePassword=True|False,
    MaxPasswordAge=123,
    PasswordReusePrevention=123,
    HardExpiry=True|False
)
```

Parameters

- **MinimumPasswordLength** (*integer*) – The minimum number of characters allowed in an IAM user password.

Default value: 6

- **RequireSymbols** (*boolean*) – Specifies whether IAM user passwords must contain at least one of the following non-alphanumeric characters:

! @ # \$ % ^ & * () _ + - = [] { } | ' ,

Default value: false

- **RequireNumbers** (*boolean*) – Specifies whether IAM user passwords must contain at least one numeric character (0 to 9).

Default value: false

- **RequireUppercaseCharacters** (*boolean*) – Specifies whether IAM user passwords must contain at least one uppercase character from the ISO basic Latin alphabet (A to Z).

Default value: false

- **RequireLowercaseCharacters** (*boolean*) – Specifies whether IAM user passwords must contain at least one lowercase character from the ISO basic Latin alphabet (a to z).

Default value: false

- **AllowUsersToChangePassword** (*boolean*) – Allows all IAM users in your account to use the AWS Management Console to change their own passwords. For more information, see [Letting IAM Users Change Their Own Passwords](#) in the *IAM User Guide* .

Default value: false

- **MaxPasswordAge** (*integer*) – The number of days that an IAM user password is valid. The default value of 0 means IAM user passwords never expire.

Default value: 0

- **PasswordReusePrevention** (*integer*) – Specifies the number of previous passwords that IAM users are prevented from reusing. The default value of 0 means IAM users are not prevented from reusing previous passwords.

Default value: 0

- **HardExpiry** (*boolean*) – Prevents IAM users from setting a new password after their password has expired.

Default value: false

Returns None

update_assume_role_policy (**kwargs)

Updates the policy that grants an entity permission to assume a role. For more information about roles, go to [Using Roles to Delegate Permissions and Federate Identities](#) .

Request Syntax

```
response = client.update_assume_role_policy(
    RoleName='string',
    PolicyDocument='string'
)
```

Parameters

- **RoleName** (*string*) – [REQUIRED]

The name of the role to update.

- **PolicyDocument** (*string*) – [REQUIRED]

The policy that grants an entity permission to assume the role.

Returns None

update_group (**kwargs)

Updates the name and/or the path of the specified group.

Warning: You should understand the implications of changing a group's path or name. For more information, see [Renaming Users and Groups](#) in the *IAM User Guide* .

Note: To change a group name the requester must have appropriate permissions on both the source object and the target object. For example, to change Managers to MGRs, the entity making the request must have permission on Managers and MGRs, or must have permission on all (*). For more information about permissions, see [Permissions and Policies](#) .

Request Syntax

```
response = client.update_group(
    GroupName='string',
    NewPath='string',
    NewGroupName='string'
)
```

Parameters

- **GroupName** (*string*) – [REQUIRED]

Name of the group to update. If you're changing the name of the group, this is the original name.

- **NewPath** (*string*) – New path for the group. Only include this if changing the group's path.
- **NewGroupName** (*string*) – New name for the group. Only include this if changing the group's name.

Returns None

update_login_profile (**kwargs)

Changes the password for the specified user.

Users can change their own passwords by calling `ChangePassword` . For more information about modifying passwords, see [Managing Passwords](#) in the *IAM User Guide* .

Request Syntax

```
response = client.update_login_profile(
    UserName='string',
    Password='string',
    PasswordResetRequired=True|False
)
```

Parameters

- **UserName** (*string*) – [REQUIRED]
The name of the user whose password you want to update.
- **Password** (*string*) – The new password for the specified user.
- **PasswordResetRequired** (*boolean*) – Require the specified user to set a new password on next sign-in.

Returns None

update_open_id_connect_provider_thumbprint (***kwargs*)

Replaces the existing list of server certificate thumbprints with a new list.

The list that you pass with this action completely replaces the existing list of thumbprints. (The lists are not merged.)

Typically, you need to update a thumbprint only when the identity provider’s certificate changes, which occurs rarely. However, if the provider’s certificate *does* change, any attempt to assume an IAM role that specifies the OIDC provider as a principal will fail until the certificate thumbprint is updated.

Note: Because trust for the OpenID Connect provider is ultimately derived from the provider’s certificate and is validated by the thumbprint, it is a best practice to limit access to the `UpdateOpenIDConnectProviderThumbprint` action to highly-privileged users.

Request Syntax

```
response = client.update_open_id_connect_provider_thumbprint(
    OpenIDConnectProviderArn='string',
    ThumbprintList=[
        'string',
    ]
)
```

Parameters

- **OpenIDConnectProviderArn** (*string*) – [REQUIRED]
The Amazon Resource Name (ARN) of the IAM OpenID Connect (OIDC) provider to update the thumbprint for. You can get a list of OIDC provider ARNs by using the `ListOpenIDConnectProviders` action.
- **ThumbprintList** (*list*) – [REQUIRED]
A list of certificate thumbprints that are associated with the specified IAM OpenID Connect provider. For more information, see `CreateOpenIDConnectProvider`.
 - (*string*) –
Contains a thumbprint for an identity provider’s server certificate.
The identity provider’s server certificate thumbprint is the hex-encoded SHA-1 hash value of the self-signed X.509 certificate used by the domain where the OpenID Connect provider makes its keys available. It is always a 40-character string.

Returns None

update_saml_provider (***kwargs*)

Updates the metadata document for an existing SAML provider.

Note: This operation requires [Signature Version 4](#).

Request Syntax

```
response = client.update_saml_provider(
    SAMLMetadataDocument='string',
    SAMLProviderArn='string'
)
```

Parameters

- **SAMLMetadataDocument** (*string*) – **[REQUIRED]**

An XML document generated by an identity provider (IdP) that supports SAML 2.0. The document includes the issuer's name, expiration information, and keys that can be used to validate the SAML authentication response (assertions) that are received from the IdP. You must generate the metadata document using the identity management software that is used as your organization's IdP.

- **SAMLProviderArn** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the SAML provider to update.

Return type dict

Returns

Response Syntax

```
{
    'SAMLProviderArn': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful UpdateSAMLProvider request.

- **SAMLProviderArn** (*string*) –

The Amazon Resource Name (ARN) of the SAML provider that was updated.

update_server_certificate (***kwargs*)

Updates the name and/or the path of the specified server certificate.

Warning: You should understand the implications of changing a server certificate's path or name. For more information, see [Managing Server Certificates](#) in the *IAM User Guide*.

Note: To change a server certificate name the requester must have appropriate permissions on both the source object and the target object. For example, to change the name from ProductionCert to ProdCert, the entity making the request must have permission on ProductionCert and ProdCert, or must have permission on all (*). For more information about permissions, see [Permissions and Policies](#).

Request Syntax

```
response = client.update_server_certificate(
    ServerCertificateName='string',
    NewPath='string',
    NewServerCertificateName='string'
)
```

Parameters

- **ServerCertificateName** (*string*) – [REQUIRED]
The name of the server certificate that you want to update.
- **NewPath** (*string*) – The new path for the server certificate. Include this only if you are updating the server certificate's path.
- **NewServerCertificateName** (*string*) – The new name for the server certificate. Include this only if you are updating the server certificate's name. The name of the certificate cannot contain any spaces.

Returns None

update_signing_certificate (***kwargs*)

Changes the status of the specified signing certificate from active to disabled, or vice versa. This action can be used to disable a user's signing certificate as part of a certificate rotation work flow.

If the `UserName` field is not specified, the `UserName` is determined implicitly based on the AWS access key ID used to sign the request. Because this action works for access keys under the AWS account, you can use this action to manage root credentials even if the AWS account has no associated users.

Request Syntax

```
response = client.update_signing_certificate(
    UserName='string',
    CertificateId='string',
    Status='Active' | 'Inactive'
)
```

Parameters

- **UserName** (*string*) – The name of the user the signing certificate belongs to.
- **CertificateId** (*string*) – [REQUIRED]
The ID of the signing certificate you want to update.
- **Status** (*string*) – [REQUIRED]
The status you want to assign to the certificate. `Active` means the certificate can be used for API calls to AWS, while `Inactive` means the certificate cannot be used.

Returns None

update_ssh_public_key (***kwargs*)

Sets the status of the specified SSH public key to active or inactive. SSH public keys that are inactive cannot be used for authentication. This action can be used to disable a user's SSH public key as part of a key rotation work flow.

The SSH public key affected by this action is used only for authenticating the associated IAM user to an AWS CodeCommit repository. For more information about using SSH keys to authenticate to an AWS CodeCommit repository, see [Set up AWS CodeCommit for SSH Connections](#) in the *AWS CodeCommit User Guide*.

Request Syntax

```
response = client.update_ssh_public_key(
    UserName='string',
    SSHPublicKeyId='string',
    Status='Active'|'Inactive'
)
```

Parameters

- **UserName** (*string*) – [REQUIRED]
The name of the IAM user associated with the SSH public key.
- **SSHPublicKeyId** (*string*) – [REQUIRED]
The unique identifier for the SSH public key.
- **Status** (*string*) – [REQUIRED]
The status to assign to the SSH public key. *Active* means the key can be used for authentication with an AWS CodeCommit repository. *Inactive* means the key cannot be used.

Returns None

update_user (***kwargs*)

Updates the name and/or the path of the specified user.

Warning: You should understand the implications of changing a user’s path or name. For more information, see [Renaming Users and Groups](#) in the *IAM User Guide* .

Note: To change a user name the requester must have appropriate permissions on both the source object and the target object. For example, to change Bob to Robert, the entity making the request must have permission on Bob and Robert, or must have permission on all (*). For more information about permissions, see [Permissions and Policies](#) .

Request Syntax

```
response = client.update_user(
    UserName='string',
    NewPath='string',
    NewUserName='string'
)
```

Parameters

- **UserName** (*string*) – [REQUIRED]
Name of the user to update. If you’re changing the name of the user, this is the original user name.
- **NewPath** (*string*) – New path for the user. Include this parameter only if you’re changing the user’s path.
- **NewUserName** (*string*) – New name for the user. Include this parameter only if you’re changing the user’s name.

Returns None

upload_server_certificate (***kwargs*)

Uploads a server certificate entity for the AWS account. The server certificate entity includes a public key certificate, a private key, and an optional certificate chain, which should all be PEM-encoded.

For information about the number of server certificates you can upload, see [Limitations on IAM Entities](#) in the *IAM User Guide* .

Note: Because the body of the public key certificate, private key, and the certificate chain can be large, you should use POST rather than GET when calling `UploadServerCertificate`. For information about setting up signatures and authorization through the API, go to [Signing AWS API Requests](#) in the *AWS General Reference*. For general information about using the Query API with IAM, go to [Making Query Requests](#) in the *IAM User Guide*.

Request Syntax

```
response = client.upload_server_certificate(
    Path='string',
    ServerCertificateName='string',
    CertificateBody='string',
    PrivateKey='string',
    CertificateChain='string'
)
```

Parameters

- **Path** (*string*) – The path for the server certificate. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

This parameter is optional. If it is not included, it defaults to a slash (/).

Note: If you are uploading a server certificate specifically for use with Amazon CloudFront distributions, you must specify a path using the `--path` option. The path must begin with `/cloudfront` and must include a trailing slash (for example, `/cloudfront/test/`).

- **ServerCertificateName** (*string*) – **[REQUIRED]**

The name for the server certificate. Do not include the path in this value. The name of the certificate cannot contain any spaces.

- **CertificateBody** (*string*) – **[REQUIRED]**

The contents of the public key certificate in PEM-encoded format.

- **PrivateKey** (*string*) – **[REQUIRED]**

The contents of the private key in PEM-encoded format.

- **CertificateChain** (*string*) – The contents of the certificate chain. This is typically a concatenation of the PEM-encoded public key certificates of the chain.

Return type dict

Returns

Response Syntax

```
{
  'ServerCertificateMetadata': {
    'Path': 'string',
    'ServerCertificateName': 'string',
    'ServerCertificateId': 'string',
    'Arn': 'string',
    'UploadDate': datetime(2015, 1, 1),
    'Expiration': datetime(2015, 1, 1)
  }
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `UploadServerCertificate` request.

– **ServerCertificateMetadata** (*dict*) –

The meta information of the uploaded server certificate without its certificate body, certificate chain, and private key.

* **Path** (*string*) –

The path to the server certificate. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

* **ServerCertificateName** (*string*) –

The name that identifies the server certificate.

* **ServerCertificateId** (*string*) –

The stable and unique string identifying the server certificate. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

* **Arn** (*string*) –

The Amazon Resource Name (ARN) specifying the server certificate. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

* **UploadDate** (*datetime*) –

The date when the server certificate was uploaded.

* **Expiration** (*datetime*) –

The date on which the certificate is set to expire.

upload_signing_certificate (***kwargs*)

Uploads an X.509 signing certificate and associates it with the specified user. Some AWS services use X.509 signing certificates to validate requests that are signed with a corresponding private key. When you upload the certificate, its default status is `Active`.

If the `UserName` field is not specified, the user name is determined implicitly based on the AWS access key ID used to sign the request. Because this action works for access keys under the AWS account, you can use this action to manage root credentials even if the AWS account has no associated users.

Note: Because the body of a X.509 certificate can be large, you should use POST rather than GET when calling `UploadSigningCertificate`. For information about setting up signatures and authorization through the API, go to [Signing AWS API Requests](#) in the *AWS General Reference*. For general information about using the Query API with IAM, go to [Making Query Requests](#) in the *Using IAM* guide.

Request Syntax

```
response = client.upload_signing_certificate(
    UserName='string',
    CertificateBody='string'
)
```

Parameters

- **UserName** (*string*) – The name of the user the signing certificate is for.
- **CertificateBody** (*string*) – **[REQUIRED]**

The contents of the signing certificate.

Return type dict

Returns

Response Syntax

```
{
    'Certificate': {
        'UserName': 'string',
        'CertificateId': 'string',
        'CertificateBody': 'string',
        'Status': 'Active'|'Inactive',
        'UploadDate': datetime(2015, 1, 1)
    }
}
```

Response Structure

- (*dict*) –

Contains the response to a successful UploadSigningCertificate request.

- **Certificate** (*dict*) –

Information about the certificate.

- * **UserName** (*string*) –

The name of the user the signing certificate is associated with.

- * **CertificateId** (*string*) –

The ID for the signing certificate.

- * **CertificateBody** (*string*) –

The contents of the signing certificate.

- * **Status** (*string*) –

The status of the signing certificate. `Active` means the key is valid for API calls, while `Inactive` means it is not.

- * **UploadDate** (*datetime*) –

The date when the signing certificate was uploaded.

upload_ssh_public_key (**kwargs)

Uploads an SSH public key and associates it with the specified IAM user.

The SSH public key uploaded by this action can be used only for authenticating the associated IAM user to an AWS CodeCommit repository. For more information about using SSH keys to authenticate to an AWS CodeCommit repository, see [Set up AWS CodeCommit for SSH Connections](#) in the *AWS CodeCommit User Guide*.

Request Syntax

```
response = client.upload_ssh_public_key(
    UserName='string',
    SSHPublicKeyBody='string'
)
```

Parameters

- **UserName** (*string*) – **[REQUIRED]**

The name of the IAM user to associate the SSH public key with.

- **SSHPublicKeyBody** (*string*) – **[REQUIRED]**

The SSH public key. The public key must be encoded in ssh-rsa format or PEM format.

Return type dict

Returns

Response Syntax

```
{
  'SSHPublicKey': {
    'UserName': 'string',
    'SSHPublicKeyId': 'string',
    'Fingerprint': 'string',
    'SSHPublicKeyBody': 'string',
    'Status': 'Active'|'Inactive',
    'UploadDate': datetime(2015, 1, 1)
  }
}
```

Response Structure

- (*dict*) –

Contains the response to a successful UploadSSHPublicKey request.

- **SSHPublicKey** (*dict*) –

Contains information about the SSH public key.

- * **UserName** (*string*) –

The name of the IAM user associated with the SSH public key.

- * **SSHPublicKeyId** (*string*) –

The unique identifier for the SSH public key.

- * **Fingerprint** (*string*) –

The MD5 message digest of the SSH public key.

- * **SSHPublicKeyBody** (*string*) –

The SSH public key.

- * **Status** (*string*) –

The status of the SSH public key. `Active` means the key can be used for authentication with an AWS CodeCommit repository. `Inactive` means the key cannot be used.

- * **UploadDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the SSH public key was uploaded.

Paginators

The available paginators are:

- `IAM.Paginator.GetGroup`
- `IAM.Paginator.ListAccessKeys`
- `IAM.Paginator.ListAccountAliases`
- `IAM.Paginator.ListGroupPolicies`
- `IAM.Paginator.ListGroups`
- `IAM.Paginator.ListGroupsForUser`
- `IAM.Paginator.ListInstanceProfiles`
- `IAM.Paginator.ListInstanceProfilesForRole`
- `IAM.Paginator.ListMFADevices`

- `IAM.Paginator.ListPolicies`
- `IAM.Paginator.ListRolePolicies`
- `IAM.Paginator.ListRoles`
- `IAM.Paginator.ListServerCertificates`
- `IAM.Paginator.ListSigningCertificates`
- `IAM.Paginator.ListUserPolicies`
- `IAM.Paginator.ListUsers`
- `IAM.Paginator.ListVirtualMFADevices`

class `IAM.Paginator.GetGroup`

```
paginator = client.get_paginator('get_group')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `IAM.Client.get_group()`.

Request Syntax

```
response_iterator = paginator.paginate(
    GroupName='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **GroupName** (*string*) – **[REQUIRED]**

The name of the group.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
    'Group': {
        'Path': 'string',
        'GroupName': 'string',
```



```

        'GroupId': 'string',
        'Arn': 'string',
        'CreateDate': datetime(2015, 1, 1)
    },
    'Users': [
        {
            'Path': 'string',
            'UserName': 'string',
            'UserId': 'string',
            'Arn': 'string',
            'CreateDate': datetime(2015, 1, 1),
            'PasswordLastUsed': datetime(2015, 1, 1)
        },
    ],
    'IsTruncated': True|False,
    'NextToken': 'string'
}

```

Response Structure

- *(dict)* –

Contains the response to a successful GetGroup request.

- **Group** *(dict)* –

Information about the group.

- * **Path** *(string)* –

The path to the group. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **GroupName** *(string)* –

The friendly name that identifies the group.

- * **GroupId** *(string)* –

The stable and unique string identifying the group. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **Arn** *(string)* –

The Amazon Resource Name (ARN) specifying the group. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- * **CreateDate** *(datetime)* –

The date and time, in [ISO 8601 date-time format](#), when the group was created.

- **Users** *(list)* –

A list of users in the group.

- * *(dict)* –

Contains information about an IAM user entity.

This data type is used as a response element in the following actions:

- CreateUser
- GetUser
- ListUsers

- **Path** *(string)* –

The path to the user. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **UserName** (*string*) –
The friendly name identifying the user.
- **UserId** (*string*) –
The stable and unique string identifying the user. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.
- **Arn** (*string*) –
The Amazon Resource Name (ARN) that identifies the user. For more information about ARNs and how to use ARNs in policies, see [IAM Identifiers](#) in the *Using IAM* guide.
- **CreateDate** (*datetime*) –
The date and time, in [ISO 8601 date-time format](#), when the user was created.
- **PasswordLastUsed** (*datetime*) –
The date and time, in [ISO 8601 date-time format](#), when the user's password was last used to sign in to an AWS website. For a list of AWS websites that capture a user's last sign-in time, see the [Credential Reports](#) topic in the *Using IAM* guide. If a password is used more than once in a five-minute span, only the first use is returned in this field. This field is null (not present) when:
 - The user does not have a password
 - The password exists but has never been used (at least not since IAM started tracking this information on October 20th, 2014)
 - there is no sign-in data associated with the userThis value is returned only in the `GetUser` and `ListUsers` actions.
- **IsTruncated** (*boolean*) –
A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.
- **NextToken** (*string*) –
A token to resume pagination.

class IAM.Paginator.**ListAccessKeys**

```
paginator = client.get_paginator('list_access_keys')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `IAM.Client.list_access_keys()`.

Request Syntax

```
response_iterator = paginator.paginate(  
    UserName='string',  
    PaginationConfig={  
        'MaxItems': 123,
```

```

        'PageSize': 123,
        'StartingToken': 'string'
    }
)

```

Parameters

- **UserName** (*string*) – The name of the user.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```

{
  'AccessKeyMetadata': [
    {
      'UserName': 'string',
      'AccessKeyId': 'string',
      'Status': 'Active'|'Inactive',
      'CreateDate': datetime(2015, 1, 1)
    },
  ],
  'IsTruncated': True|False,
  'NextToken': 'string'
}

```

Response Structure

- (*dict*) –
Contains the response to a successful `ListAccessKeys` request.
 - **AccessKeyMetadata** (*list*) –
A list of access key metadata.
 - * (*dict*) –
Contains information about an AWS access key, without its secret key.
This data type is used as a response element in the `ListAccessKeys` action.
 - **UserName** (*string*) –
The name of the IAM user that the key is associated with.
 - **AccessKeyId** (*string*) –
The ID for this access key.

- **Status** (*string*) –

The status of the access key. `Active` means the key is valid for API calls; `Inactive` means it is not.

- **CreateDate** (*datetime*) –

The date when the access key was created.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

- **NextToken** (*string*) –

A token to resume pagination.

class `IAM.Paginator.ListAccountAliases`

```
paginator = client.get_paginator('list_account_aliases')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `IAM.Client.list_account_aliases()`.

Request Syntax

```
response_iterator = paginator.paginate(
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
    'AccountAliases': [
```

```

        'string',
    ],
    'IsTruncated': True|False,
    'NextToken': 'string'
}

```

Response Structure

- *(dict)* –

Contains the response to a successful ListAccountAliases request.

- **AccountAliases** (*list*) –

A list of aliases associated with the account.

* (*string*) –

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the Marker request parameter to retrieve more items. Note that IAM might return fewer than the MaxItems number of results even when there are more results available. We recommend that you check IsTruncated after every call to ensure that you receive all of your results.

- **NextToken** (*string*) –

A token to resume pagination.

class IAM.Paginator.ListGroupPolicies

```
paginator = client.get_paginator('list_group_policies')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `IAM.Client.list_group_policies()`.

Request Syntax

```

response_iterator = paginator.paginate(
    GroupName='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)

```

Parameters

- **GroupName** (*string*) – **[REQUIRED]**

The name of the group to list policies for.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a NextToken will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
    'PolicyNames': [
        'string',
    ],
    'IsTruncated': True|False,
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `ListGroupPolicies` request.

- **PolicyNames** (*list*) –

A list of policy names.

* (*string*) –

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

- **NextToken** (*string*) –

A token to resume pagination.

class `IAM.Paginator.ListGroups`

```
paginator = client.get_paginator('list_groups')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `IAM.Client.list_groups()`.

Request Syntax

```
response_iterator = paginator.paginate(
    PathPrefix='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **PathPrefix** (*string*) – The path prefix for filtering the results. For example, the prefix `/division_abc/subdivision_xyz/` gets all groups whose path starts with `/division_abc/subdivision_xyz/`.

This parameter is optional. If it is not included, it defaults to a slash (`/`), listing all groups.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Groups': [
    {
      'Path': 'string',
      'GroupName': 'string',
      'GroupId': 'string',
      'Arn': 'string',
      'CreateDate': datetime(2015, 1, 1)
    },
  ],
  'IsTruncated': True|False,
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `ListGroups` request.

- **Groups** (*list*) –

A list of groups.

- * (*dict*) –

Contains information about an IAM group entity.

This data type is used as a response element in the following actions:

- `CreateGroup`
- `GetGroup`
- `ListGroups`
- **Path** (*string*) –

The path to the group. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **GroupName** (*string*) –
The friendly name that identifies the group.
- **GroupId** (*string*) –
The stable and unique string identifying the group. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.
- **Arn** (*string*) –
The Amazon Resource Name (ARN) specifying the group. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.
- **CreateDate** (*datetime*) –
The date and time, in [ISO 8601 date-time format](#), when the group was created.
- **IsTruncated** (*boolean*) –
A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.
- **NextToken** (*string*) –
A token to resume pagination.

class `IAM.Paginator.ListGroupsForUser`

```
paginator = client.get_paginator('list_groups_for_user')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `IAM.Client.list_groups_for_user()`.

Request Syntax

```
response_iterator = paginator.paginate(
    UserName='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **UserName** (*string*) – **[REQUIRED]**
The name of the user to list groups for.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Groups': [
    {
      'Path': 'string',
      'GroupName': 'string',
      'GroupId': 'string',
      'Arn': 'string',
      'CreateDate': datetime(2015, 1, 1)
    },
  ],
  'IsTruncated': True|False,
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `ListGroupsForUser` request.

- **Groups** (*list*) –

A list of groups.

- * (*dict*) –

Contains information about an IAM group entity.

This data type is used as a response element in the following actions:

- `CreateGroup`
- `GetGroup`
- `ListGroups`

- **Path** (*string*) –

The path to the group. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **GroupName** (*string*) –

The friendly name that identifies the group.

- **GroupId** (*string*) –

The stable and unique string identifying the group. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) specifying the group. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- **CreateDate** (*datetime*) –

The date and time, in ISO 8601 date-time format , when the group was created.

– **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

– **NextToken** (*string*) –

A token to resume pagination.

class IAM.Paginator.**ListInstanceProfiles**

```
paginator = client.get_paginator('list_instance_profiles')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `IAM.Client.list_instance_profiles()`.

Request Syntax

```
response_iterator = paginator.paginate(
    PathPrefix='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **PathPrefix** (*string*) – The path prefix for filtering the results. For example, the prefix `/application_abc/component_xyz/` gets all instance profiles whose path starts with `/application_abc/component_xyz/`.

This parameter is optional. If it is not included, it defaults to a slash (`/`), listing all instance profiles.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

– **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

– **PageSize** (*integer*) –

The size of each page.

– **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```

{
  'InstanceProfiles': [
    {
      'Path': 'string',
      'InstanceProfileName': 'string',
      'InstanceProfileId': 'string',
      'Arn': 'string',
      'CreateDate': datetime(2015, 1, 1),
      'Roles': [
        {
          'Path': 'string',
          'RoleName': 'string',
          'RoleId': 'string',
          'Arn': 'string',
          'CreateDate': datetime(2015, 1, 1),
          'AssumeRolePolicyDocument': 'string'
        },
      ],
    },
  ],
  'IsTruncated': True|False,
  'NextToken': 'string'
}

```

Response Structure

- *(dict)* –

Contains the response to a successful ListInstanceProfiles request.

- **InstanceProfiles** (*list*) –

A list of instance profiles.

- * *(dict)* –

Contains information about an instance profile.

This data type is used as a response element in the following actions:

- CreateInstanceProfile
- GetInstanceProfile
- ListInstanceProfiles
- ListInstanceProfilesForRole

- **Path** (*string*) –

The path to the instance profile. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **InstanceProfileName** (*string*) –

The name identifying the instance profile.

- **InstanceProfileId** (*string*) –

The stable and unique string identifying the instance profile. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) specifying the instance profile. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- **CreateDate** (*datetime*) –

The date when the instance profile was created.

- **Roles** (*list*) –

The role associated with the instance profile.

- (*dict*) –

Contains information about an IAM role.

This data type is used as a response element in the following actions:

- CreateRole
- GetRole
- ListRoles
- **Path** (*string*) –

The path to the role. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **RoleName** (*string*) –

The friendly name that identifies the role.

- **RoleId** (*string*) –

The stable and unique string identifying the role. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) specifying the role. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the role was created.

- **AssumeRolePolicyDocument** (*string*) –

The policy that grants an entity permission to assume the role.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

- **NextToken** (*string*) –

A token to resume pagination.

class IAM.Paginator.**ListInstanceProfilesForRole**

```
paginator = client.get_paginator('list_instance_profiles_for_role')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `IAM.Client.list_instance_profiles_for_role()`.

Request Syntax

```

response_iterator = paginator.paginate(
    RoleName='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)

```

Parameters

- **RoleName** (*string*) – [REQUIRED]

The name of the role to list instance profiles for.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```

{
    'InstanceProfiles': [
        {
            'Path': 'string',
            'InstanceProfileName': 'string',
            'InstanceProfileId': 'string',
            'Arn': 'string',
            'CreateDate': datetime(2015, 1, 1),
            'Roles': [
                {
                    'Path': 'string',
                    'RoleName': 'string',
                    'RoleId': 'string',
                    'Arn': 'string',
                    'CreateDate': datetime(2015, 1, 1),
                    'AssumeRolePolicyDocument': 'string'
                },
            ],
        },
    ],
    'IsTruncated': True|False,
    'NextToken': 'string'
}

```

Response Structure

- (*dict*) –

Contains the response to a successful `ListInstanceProfilesForRole` request.

– **InstanceProfiles** (*list*) –

A list of instance profiles.

* (*dict*) –

Contains information about an instance profile.

This data type is used as a response element in the following actions:

- `CreateInstanceProfile`
- `GetInstanceProfile`
- `ListInstanceProfiles`
- `ListInstanceProfilesForRole`

· **Path** (*string*) –

The path to the instance profile. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

· **InstanceProfileName** (*string*) –

The name identifying the instance profile.

· **InstanceProfileId** (*string*) –

The stable and unique string identifying the instance profile. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

· **Arn** (*string*) –

The Amazon Resource Name (ARN) specifying the instance profile. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

· **CreateDate** (*datetime*) –

The date when the instance profile was created.

· **Roles** (*list*) –

The role associated with the instance profile.

· (*dict*) –

Contains information about an IAM role.

This data type is used as a response element in the following actions:

- `CreateRole`
- `GetRole`
- `ListRoles`
- **Path** (*string*) –

The path to the role. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

· **RoleName** (*string*) –

The friendly name that identifies the role.

· **RoleId** (*string*) –

The stable and unique string identifying the role. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

· **Arn** (*string*) –

The Amazon Resource Name (ARN) specifying the role. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the role was created.

- **AssumeRolePolicyDocument** (*string*) –

The policy that grants an entity permission to assume the role.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

- **NextToken** (*string*) –

A token to resume pagination.

class IAM.Paginator.**ListMFADevices**

```
paginator = client.get_paginator('list_mfa_devices')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `IAM.Client.list_mfa_devices()`.

Request Syntax

```
response_iterator = paginator.paginate(
    UserName='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **UserName** (*string*) – The name of the user whose MFA devices you want to list.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'MFADevices': [
    {
      'UserName': 'string',
      'SerialNumber': 'string',
      'EnableDate': datetime(2015, 1, 1)
    },
  ],
  'IsTruncated': True|False,
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful ListMFADevices request.

- **MFADevices** (*list*) –

A list of MFA devices.

- * (*dict*) –

Contains information about an MFA device.

This data type is used as a response element in the ListMFADevices action.

- **UserName** (*string*) –

The user with whom the MFA device is associated.

- **SerialNumber** (*string*) –

The serial number that uniquely identifies the MFA device. For virtual MFA devices, the serial number is the device ARN.

- **EnableDate** (*datetime*) –

The date when the MFA device was enabled for the user.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the Marker request parameter to retrieve more items. Note that IAM might return fewer than the MaxItems number of results even when there are more results available. We recommend that you check IsTruncated after every call to ensure that you receive all of your results.

- **NextToken** (*string*) –

A token to resume pagination.

class IAM.Paginator.**ListPolicies**

```
paginator = client.get_paginator('list_policies')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `IAM.Client.list_policies()`.

Request Syntax


```

response_iterator = paginator.paginate(
    Scope='All'|'AWS'|'Local',
    OnlyAttached=True|False,
    PathPrefix='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)

```

Parameters

- **Scope** (*string*) – The scope to use for filtering the results.

To list only AWS managed policies, set `Scope` to `AWS`. To list only the customer managed policies in your AWS account, set `Scope` to `Local`.

This parameter is optional. If it is not included, or if it is set to `All`, all policies are returned.

- **OnlyAttached** (*boolean*) – A flag to filter the results to only the attached policies.

When `OnlyAttached` is `true`, the returned list contains only the policies that are attached to a user, group, or role. When `OnlyAttached` is `false`, or when the parameter is not included, all policies are returned.

- **PathPrefix** (*string*) – The path prefix for filtering the results. This parameter is optional. If it is not included, it defaults to a slash (/), listing all policies.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```

{
    'Policies': [
        {
            'PolicyName': 'string',
            'PolicyId': 'string',
            'Arn': 'string',
            'Path': 'string',
            'DefaultVersionId': 'string',
            'AttachmentCount': 123,
            'IsAttachable': True|False,
            'Description': 'string',
            'CreateDate': datetime(2015, 1, 1),

```

```
        'UpdateDate': datetime(2015, 1, 1)
    },
],
'IsTruncated': True|False,
'NextToken': 'string'
}
```

Response Structure

- *(dict)* –

Contains the response to a successful ListPolicies request.

- **Policies** (*list*) –

A list of policies.

- * *(dict)* –

Contains information about a managed policy.

This data type is used as a response element in the CreatePolicy , GetPolicy , and ListPolicies actions.

For more information about managed policies, refer to [Managed Policies and Inline Policies](#) in the *Using IAM* guide.

- **PolicyName** (*string*) –

The friendly name (not ARN) identifying the policy.

- **PolicyId** (*string*) –

The stable and unique string identifying the policy.

For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- **Arn** (*string*) –

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference* .

- **Path** (*string*) –

The path to the policy.

For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **DefaultVersionId** (*string*) –

The identifier for the version of the policy that is set as the default version.

- **AttachmentCount** (*integer*) –

The number of entities (users, groups, and roles) that the policy is attached to.

- **IsAttachable** (*boolean*) –

Specifies whether the policy can be attached to an IAM user, group, or role.

- **Description** (*string*) –

A friendly description of the policy.

This element is included in the response to the `GetPolicy` operation. It is not included in the response to the `ListPolicies` operation.

- **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the policy was created.

- **UpdateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the policy was last updated.

When a policy has only one version, this field contains the date and time when the policy was created. When a policy has more than one version, this field contains the date and time when the most recent policy version was created.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

- **NextToken** (*string*) –

A token to resume pagination.

class `IAM.Paginator.ListRolePolicies`

```
paginator = client.get_paginator('list_role_policies')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `IAM.Client.list_role_policies()`.

Request Syntax

```
response_iterator = paginator.paginate(
    RoleName='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **RoleName** (*string*) – [REQUIRED]

The name of the role to list policies for.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
    'PolicyNames': [
        'string',
    ],
    'IsTruncated': True|False,
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `ListRolePolicies` request.

- **PolicyNames** (*list*) –

A list of policy names.

* (*string*) –

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

- **NextToken** (*string*) –

A token to resume pagination.

class `IAM.Paginator.ListRoles`

```
paginator = client.get_paginator('list_roles')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `IAM.Client.list_roles()`.

Request Syntax

```
response_iterator = paginator.paginate(
    PathPrefix='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **PathPrefix** (*string*) – The path prefix for filtering the results. For example, the prefix `/application_abc/component_xyz/` gets all roles whose path starts with `/application_abc/component_xyz/`.

This parameter is optional. If it is not included, it defaults to a slash (`/`), listing all roles.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Roles': [
    {
      'Path': 'string',
      'RoleName': 'string',
      'RoleId': 'string',
      'Arn': 'string',
      'CreateDate': datetime(2015, 1, 1),
      'AssumeRolePolicyDocument': 'string'
    },
  ],
  'IsTruncated': True|False,
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `ListRoles` request.

- **Roles** (*list*) –

A list of roles.

- * (*dict*) –

Contains information about an IAM role.

This data type is used as a response element in the following actions:

- `CreateRole`
- `GetRole`
- `ListRoles`

- **Path** (*string*) –

The path to the role. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **RoleName** (*string*) –
The friendly name that identifies the role.
- **RoleId** (*string*) –
The stable and unique string identifying the role. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.
- **Arn** (*string*) –
The Amazon Resource Name (ARN) specifying the role. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.
- **CreateDate** (*datetime*) –
The date and time, in [ISO 8601 date-time format](#), when the role was created.
- **AssumeRolePolicyDocument** (*string*) –
The policy that grants an entity permission to assume the role.
- **IsTruncated** (*boolean*) –
A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.
- **NextToken** (*string*) –
A token to resume pagination.

`class IAM.Paginator.ListServerCertificates`

```
paginator = client.get_paginator('list_server_certificates')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `IAM.Client.list_server_certificates()`.

Request Syntax

```
response_iterator = paginator.paginate(  
    PathPrefix='string',  
    PaginationConfig={  
        'MaxItems': 123,  
        'PageSize': 123,  
        'StartingToken': 'string'  
    }  
)
```

Parameters

- **PathPrefix** (*string*) – The path prefix for filtering the results. For example: `/company/servercerts` would get all server certificates for which the path starts with `/company/servercerts`.

This parameter is optional. If it is not included, it defaults to a slash (`/`), listing all server certificates.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'ServerCertificateMetadataList': [
    {
      'Path': 'string',
      'ServerCertificateName': 'string',
      'ServerCertificateId': 'string',
      'Arn': 'string',
      'UploadDate': datetime(2015, 1, 1),
      'Expiration': datetime(2015, 1, 1)
    },
  ],
  'IsTruncated': True|False,
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
Contains the response to a successful `ListServerCertificates` request.
 - **ServerCertificateMetadataList** (*list*) –
A list of server certificates.
 - * (*dict*) –
Contains information about a server certificate without its certificate body, certificate chain, and private key.
This data type is used as a response element in the `UploadServerCertificate` and `ListServerCertificates` actions.
 - **Path** (*string*) –
The path to the server certificate. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.
 - **ServerCertificateName** (*string*) –
The name that identifies the server certificate.
 - **ServerCertificateId** (*string*) –
The stable and unique string identifying the server certificate. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- **Arn** (*string*) –
The Amazon Resource Name (ARN) specifying the server certificate. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.
- **UploadDate** (*datetime*) –
The date when the server certificate was uploaded.
- **Expiration** (*datetime*) –
The date on which the certificate is set to expire.
- **IsTruncated** (*boolean*) –
A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.
- **NextToken** (*string*) –
A token to resume pagination.

class IAM.Paginator.**ListSigningCertificates**

```
paginator = client.get_paginator('list_signing_certificates')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `IAM.Client.list_signing_certificates()`.

Request Syntax

```
response_iterator = paginator.paginate(  
    UserName='string',  
    PaginationConfig={  
        'MaxItems': 123,  
        'PageSize': 123,  
        'StartingToken': 'string'  
    }  
)
```

Parameters

- **UserName** (*string*) – The name of the user.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Certificates': [
    {
      'UserName': 'string',
      'CertificateId': 'string',
      'CertificateBody': 'string',
      'Status': 'Active'|'Inactive',
      'UploadDate': datetime(2015, 1, 1)
    },
  ],
  'IsTruncated': True|False,
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful ListSigningCertificates request.

- **Certificates** (*list*) –

A list of the user's signing certificate information.

- * (*dict*) –

Contains information about an X.509 signing certificate.

This data type is used as a response element in the UploadSigningCertificate and ListSigningCertificates actions.

- **UserName** (*string*) –

The name of the user the signing certificate is associated with.

- **CertificateId** (*string*) –

The ID for the signing certificate.

- **CertificateBody** (*string*) –

The contents of the signing certificate.

- **Status** (*string*) –

The status of the signing certificate. `Active` means the key is valid for API calls, while `Inactive` means it is not.

- **UploadDate** (*datetime*) –

The date when the signing certificate was uploaded.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

- **NextToken** (*string*) –

A token to resume pagination.

class IAM.Paginator.**ListUserPolicies**

```
paginator = client.get_paginator('list_user_policies')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `IAM.Client.list_user_policies()`.

Request Syntax

```
response_iterator = paginator.paginate(
    UserName='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **UserName** (*string*) – [REQUIRED]

The name of the user to list policies for.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns**Response Syntax**

```
{
    'PolicyNames': [
        'string',
    ],
    'IsTruncated': True|False,
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `ListUserPolicies` request.

- **PolicyNames** (*list*) –

A list of policy names.

* (*string*) –

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

– **NextToken** (*string*) –

A token to resume pagination.

class `IAM.Paginator.ListUsers`

```
paginator = client.get_paginator('list_users')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `IAM.Client.list_users()`.

Request Syntax

```
response_iterator = paginator.paginate(
    PathPrefix='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **PathPrefix** (*string*) – The path prefix for filtering the results. For example: `/division_abc/subdivision_xyz/`, which would get all user names whose path starts with `/division_abc/subdivision_xyz/`.

This parameter is optional. If it is not included, it defaults to a slash (`/`), listing all user names.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

– **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

– **PageSize** (*integer*) –

The size of each page.

– **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
    'Users': [
        {
            'Path': 'string',
```

```
        'UserName': 'string',
        'UserId': 'string',
        'Arn': 'string',
        'CreateDate': datetime(2015, 1, 1),
        'PasswordLastUsed': datetime(2015, 1, 1)
    },
],
'IsTruncated': True|False,
'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful ListUsers request.

- **Users** (*list*) –

A list of users.

- * (*dict*) –

Contains information about an IAM user entity.

This data type is used as a response element in the following actions:

- CreateUser
- GetUser
- ListUsers

- **Path** (*string*) –

The path to the user. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **UserName** (*string*) –

The friendly name identifying the user.

- **UserId** (*string*) –

The stable and unique string identifying the user. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) that identifies the user. For more information about ARNs and how to use ARNs in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the user was created.

- **PasswordLastUsed** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the user's password was last used to sign in to an AWS website. For a list of AWS websites that capture a user's last sign-in time, see the [Credential Reports](#) topic in the *Using IAM* guide. If a password is used more than once in a five-minute span, only the first use is returned in this field. This field is null (not present) when:

- The user does not have a password
- The password exists but has never been used (at least not since IAM started tracking this information on October 20th, 2014)

- there is no sign-in data associated with the user

This value is returned only in the `GetUser` and `ListUsers` actions.

– **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

– **NextToken** (*string*) –

A token to resume pagination.

class `IAM.Paginator.ListVirtualMFADevices`

```
paginator = client.get_paginator('list_virtual_mfa_devices')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `IAM.Client.list_virtual_mfa_devices()`.

Request Syntax

```
response_iterator = paginator.paginate(
    AssignmentStatus='Assigned'|'Unassigned'|'Any',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **AssignmentStatus** (*string*) – The status (unassigned or assigned) of the devices to list. If you do not specify an `AssignmentStatus`, the action defaults to `Any` which lists both assigned and unassigned virtual MFA devices.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

– **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

– **PageSize** (*integer*) –

The size of each page.

– **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
  'VirtualMFADevices': [
    {
      'SerialNumber': 'string',
      'Base32StringSeed': b'bytes',
      'QRCodePNG': b'bytes',
      'User': {
        'Path': 'string',
        'UserName': 'string',
        'UserId': 'string',
        'Arn': 'string',
        'CreateDate': datetime(2015, 1, 1),
        'PasswordLastUsed': datetime(2015, 1, 1)
      },
      'EnableDate': datetime(2015, 1, 1)
    },
  ],
  'IsTruncated': True|False,
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful ListVirtualMFADevices request.

- **VirtualMFADevices** (*list*) –

The list of virtual MFA devices in the current account that match the AssignmentStatus value that was passed in the request.

- * (*dict*) –

Contains information about a virtual MFA device.

- **SerialNumber** (*string*) –

The serial number associated with VirtualMFADevice .

- **Base32StringSeed** (*bytes*) –

The Base32 seed defined as specified in [RFC3548](#) . The Base32StringSeed is Base64-encoded.

- **QRCodePNG** (*bytes*) –

A QR code PNG image that encodes otpauth://totp/\$virtualMFADeviceName@\$AccountName?secret= where \$virtualMFADeviceName is one of the create call arguments, AccountName is the user name if set (otherwise, the account ID otherwise), and Base32String is the seed in Base32 format. The Base32String value is Base64-encoded.

- **User** (*dict*) –

Contains information about an IAM user entity.

This data type is used as a response element in the following actions:

- CreateUser
- GetUser
- ListUsers
- **Path** (*string*) –

The path to the user. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

- **UserName** (*string*) –

The friendly name identifying the user.

- **UserId** (*string*) –

The stable and unique string identifying the user. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

- **Arn** (*string*) –

The Amazon Resource Name (ARN) that identifies the user. For more information about ARNs and how to use ARNs in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

- **CreateDate** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the user was created.

- **PasswordLastUsed** (*datetime*) –

The date and time, in [ISO 8601 date-time format](#), when the user's password was last used to sign in to an AWS website. For a list of AWS websites that capture a user's last sign-in time, see the [Credential Reports](#) topic in the *Using IAM* guide. If a password is used more than once in a five-minute span, only the first use is returned in this field. This field is null (not present) when:

- The user does not have a password
 - The password exists but has never been used (at least not since IAM started tracking this information on October 20th, 2014)
 - there is no sign-in data associated with the user
- This value is returned only in the `GetUser` and `ListUsers` actions.

- **EnableDate** (*datetime*) –

The date and time on which the virtual MFA device was enabled.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more items to return. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more items. Note that IAM might return fewer than the `MaxItems` number of results even when there are more results available. We recommend that you check `IsTruncated` after every call to ensure that you receive all of your results.

- **NextToken** (*string*) –

A token to resume pagination.

Waiters

The available waiters are:

- `IAM.Waiter.InstanceProfileExists`
- `IAM.Waiter.UserExists`

class IAM.Waiter.InstanceProfileExists

```
waiter = client.get_waiter('instance_profile_exists')
```

wait (***kwargs*)

Polls `IAM.Client.get_instance_profile()` every 1 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(
    InstanceProfileName='string'
)
```

Parameters `InstanceProfileName` (*string*) – [REQUIRED]

The name of the instance profile to get information about.

Returns None

class IAM.Waiter.UserExists

```
waiter = client.get_waiter('user_exists')
```

wait (***kwargs*)

Polls `IAM.Client.get_user()` every 1 seconds until a successful state is reached. An error is returned after 20 failed checks.

Request Syntax

```
waiter.wait(
    UserName='string'
)
```

Parameters `UserName` (*string*) – The name of the user to get information about.

This parameter is optional. If it is not included, it defaults to the user making the request.

Returns None

Service Resource

class IAM.ServiceResource

A resource representing AWS Identity and Access Management (IAM):

```
import boto3
```

```
iam = boto3.resource('iam')
```

These are the resource's available actions:

- `change_password()`
- `create_account_alias()`
- `create_account_password_policy()`
- `create_group()`
- `create_instance_profile()`
- `create_policy()`

- `create_role()`
- `create_saml_provider()`
- `create_server_certificate()`
- `create_signing_certificate()`
- `create_user()`
- `create_virtual_mfa_device()`

These are the resource's available sub-resources:

- `AccessKey()`
- `AccessKeyPair()`
- `AccountPasswordPolicy()`
- `AccountSummary()`
- `AssumeRolePolicy()`
- `CurrentUser()`
- `Group()`
- `GroupPolicy()`
- `InstanceProfile()`
- `LoginProfile()`
- `MfaDevice()`
- `Policy()`
- `PolicyVersion()`
- `Role()`
- `RolePolicy()`
- `SamlProvider()`
- `ServerCertificate()`
- `SigningCertificate()`
- `User()`
- `UserPolicy()`
- `VirtualMfaDevice()`

These are the resource's available collections:

- `groups`
- `instance_profiles`
- `policies`
- `roles`
- `saml_providers`
- `server_certificates`
- `users`
- `virtual_mfa_devices`

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

`change_password` (***kwargs*)

Changes the password of the IAM user who is calling this action. The root account password is not affected by this action.

To change the password for a different user, see `UpdateLoginProfile`. For more information about modifying passwords, see [Managing Passwords](#) in the *IAM User Guide*.

Request Syntax

```
response = iam.change_password(
    OldPassword='string',
    NewPassword='string'
)
```

Parameters

- **OldPassword** (*string*) – [REQUIRED]
The IAM user’s current password.
- **NewPassword** (*string*) – [REQUIRED]
The new password. The new password must conform to the AWS account’s password policy, if one exists.

Returns None**create_account_alias** (***kwargs*)

Creates an alias for your AWS account. For information about using an AWS account alias, see [Using an Alias for Your AWS Account ID](#) in the *IAM User Guide* .

Request Syntax

```
response = iam.create_account_alias(  
    AccountAlias='string'  
)
```

Parameters **AccountAlias** (*string*) – [REQUIRED]

The account alias to create.

Returns None**create_account_password_policy** (***kwargs*)

Updates the password policy settings for the AWS account.

Note: This action does not support partial updates. No parameters are required, but if you do not specify a parameter, that parameter’s value reverts to its default value. See the **Request Parameters** section for each parameter’s default value.

For more information about using a password policy, see [Managing an IAM Password Policy](#) in the *IAM User Guide* .

Request Syntax

```
account_password_policy = iam.create_account_password_policy(  
    MinimumPasswordLength=123,  
    RequireSymbols=True|False,  
    RequireNumbers=True|False,  
    RequireUppercaseCharacters=True|False,  
    RequireLowercaseCharacters=True|False,  
    AllowUsersToChangePassword=True|False,  
    MaxPasswordAge=123,  
    PasswordReusePrevention=123,  
    HardExpiry=True|False  
)
```

Parameters

- **MinimumPasswordLength** (*integer*) – The minimum number of characters allowed in an IAM user password.
Default value: 6
- **RequireSymbols** (*boolean*) – Specifies whether IAM user passwords must contain at least one of the following non-alphanumeric characters:
`! @ # $ % ^ & * ( ) _ + - = [ ] { } | ‘`
Default value: false

- **RequireNumbers** (*boolean*) – Specifies whether IAM user passwords must contain at least one numeric character (0 to 9).

Default value: false

- **RequireUppercaseCharacters** (*boolean*) – Specifies whether IAM user passwords must contain at least one uppercase character from the ISO basic Latin alphabet (A to Z).

Default value: false

- **RequireLowercaseCharacters** (*boolean*) – Specifies whether IAM user passwords must contain at least one lowercase character from the ISO basic Latin alphabet (a to z).

Default value: false

- **AllowUsersToChangePassword** (*boolean*) – Allows all IAM users in your account to use the AWS Management Console to change their own passwords. For more information, see [Letting IAM Users Change Their Own Passwords](#) in the *IAM User Guide*.

Default value: false

- **MaxPasswordAge** (*integer*) – The number of days that an IAM user password is valid. The default value of 0 means IAM user passwords never expire.

Default value: 0

- **PasswordReusePrevention** (*integer*) – Specifies the number of previous passwords that IAM users are prevented from reusing. The default value of 0 means IAM users are not prevented from reusing previous passwords.

Default value: 0

- **HardExpiry** (*boolean*) – Prevents IAM users from setting a new password after their password has expired.

Default value: false

Return type `iam.AccountPasswordPolicy`

Returns AccountPasswordPolicy resource

create_group (***kwargs*)

Creates a new group.

For information about the number of groups you can create, see [Limitations on IAM Entities](#) in the *IAM User Guide*.

Request Syntax

```
group = iam.create_group(
    Path='string',
    GroupName='string'
)
```

Parameters

- **Path** (*string*) – The path to the group. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

This parameter is optional. If it is not included, it defaults to a slash (/).

- **GroupName** (*string*) – **[REQUIRED]**

The name of the group to create. Do not include the path in this value.

Return type `iam.Group`

Returns Group resource

create_instance_profile (**kwargs)

Creates a new instance profile. For information about instance profiles, go to [About Instance Profiles](#) .

For information about the number of instance profiles you can create, see [Limitations on IAM Entities](#) in the *IAM User Guide* .

Request Syntax

```
instance_profile = iam.create_instance_profile(
    InstanceProfileName='string',
    Path='string'
)
```

Parameters

- **InstanceProfileName** (*string*) – [REQUIRED]

The name of the instance profile to create.

- **Path** (*string*) – The path to the instance profile. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

This parameter is optional. If it is not included, it defaults to a slash (/).

Return type `iam.InstanceProfile`

Returns InstanceProfile resource

create_policy (**kwargs)

Creates a new managed policy for your AWS account.

This operation creates a policy version with a version identifier of v1 and sets v1 as the policy’s default version. For more information about policy versions, see [Versioning for Managed Policies](#) in the *IAM User Guide* .

For more information about managed policies in general, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide* .

Request Syntax

```
policy = iam.create_policy(
    PolicyName='string',
    Path='string',
    PolicyDocument='string',
    Description='string'
)
```

Parameters

- **PolicyName** (*string*) – [REQUIRED]

The name of the policy document.

- **Path** (*string*) – The path for the policy.

For more information about paths, see [IAM Identifiers](#) in the *IAM User Guide* .

This parameter is optional. If it is not included, it defaults to a slash (/).

- **PolicyDocument** (*string*) – [REQUIRED]

The policy document.

- **Description** (*string*) – A friendly description of the policy.

Typically used to store information about the permissions defined in the policy. For example, “Grants access to production DynamoDB tables.”

The policy description is immutable. After a value is assigned, it cannot be changed.

Return type `iam.Policy`**Returns** Policy resource**create_role** (***kwargs*)

Creates a new role for your AWS account. For more information about roles, go to [Working with Roles](#) . For information about limitations on role names and the number of roles you can create, go to [Limitations on IAM Entities](#) in the *IAM User Guide* .

The policy in the following example grants permission to an EC2 instance to assume the role.

Request Syntax

```
role = iam.create_role(
    Path='string',
    RoleName='string',
    AssumeRolePolicyDocument='string'
)
```

Parameters

- **Path** (*string*) – The path to the role. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

This parameter is optional. If it is not included, it defaults to a slash (/).

- **RoleName** (*string*) – [REQUIRED]

The name of the role to create.

- **AssumeRolePolicyDocument** (*string*) – [REQUIRED]

The policy that grants an entity permission to assume the role.

Return type `iam.Role`**Returns** Role resource**create_saml_provider** (***kwargs*)

Creates an IAM entity to describe an identity provider (IdP) that supports SAML 2.0.

The SAML provider that you create with this operation can be used as a principal in a role's trust policy to establish a trust relationship between AWS and a SAML identity provider. You can create an IAM role that supports Web-based single sign-on (SSO) to the AWS Management Console or one that supports API access to AWS.

When you create the SAML provider, you upload an a SAML metadata document that you get from your IdP and that includes the issuer's name, expiration information, and keys that can be used to validate the SAML authentication response (assertions) that are received from the IdP. You must generate the metadata document using the identity management software that is used as your organization's IdP.

Note: This operation requires [Signature Version 4](#) .

For more information, see [Enabling SAML 2.0 Federated Users to Access the AWS Management Console](#) and [About SAML 2.0-based Federation](#) in the *IAM User Guide* .

Request Syntax

```
saml_provider = iam.create_saml_provider(
    SAMLMetadataDocument='string',
    Name='string'
)
```

Parameters

- **SAMLMetadataDocument** (*string*) – [REQUIRED]

An XML document generated by an identity provider (IdP) that supports SAML 2.0. The document includes the issuer's name, expiration information, and keys that can be used to validate the SAML authentication response (assertions) that are received from the IdP. You must generate the metadata document using the identity management software that is used as your organization's IdP.

For more information, see [About SAML 2.0-based Federation](#) in the *IAM User Guide*.

- **Name** (*string*) – [REQUIRED]

The name of the provider to create.

Return type `iam.SamlProvider`

Returns `SamlProvider` resource

create_server_certificate (***kwargs*)

Uploads a server certificate entity for the AWS account. The server certificate entity includes a public key certificate, a private key, and an optional certificate chain, which should all be PEM-encoded.

For information about the number of server certificates you can upload, see [Limitations on IAM Entities](#) in the *IAM User Guide*.

Note: Because the body of the public key certificate, private key, and the certificate chain can be large, you should use POST rather than GET when calling `UploadServerCertificate`. For information about setting up signatures and authorization through the API, go to [Signing AWS API Requests](#) in the *AWS General Reference*. For general information about using the Query API with IAM, go to [Making Query Requests](#) in the *IAM User Guide*.

Request Syntax

```
server_certificate = iam.create_server_certificate(
    Path='string',
    ServerCertificateName='string',
    CertificateBody='string',
    PrivateKey='string',
    CertificateChain='string'
)
```

Parameters

- **Path** (*string*) – The path for the server certificate. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

This parameter is optional. If it is not included, it defaults to a slash (/).

Note: If you are uploading a server certificate specifically for use with Amazon CloudFront distributions, you must specify a path using the `--path` option. The path must begin with `/cloudfront` and must include a trailing slash (for example, `/cloudfront/test/`).

- **ServerCertificateName** (*string*) – [REQUIRED]

The name for the server certificate. Do not include the path in this value. The name of the certificate cannot contain any spaces.

- **CertificateBody** (*string*) – [REQUIRED]

The contents of the public key certificate in PEM-encoded format.

- **PrivateKey** (*string*) – [REQUIRED]

The contents of the private key in PEM-encoded format.

- **CertificateChain** (*string*) – The contents of the certificate chain. This is typically a concatenation of the PEM-encoded public key certificates of the chain.

Return type `iam.ServerCertificate`

Returns ServerCertificate resource

create_signing_certificate (***kwargs*)

Uploads an X.509 signing certificate and associates it with the specified user. Some AWS services use X.509 signing certificates to validate requests that are signed with a corresponding private key. When you upload the certificate, its default status is `Active`.

If the `UserName` field is not specified, the user name is determined implicitly based on the AWS access key ID used to sign the request. Because this action works for access keys under the AWS account, you can use this action to manage root credentials even if the AWS account has no associated users.

Note: Because the body of a X.509 certificate can be large, you should use POST rather than GET when calling `UploadSigningCertificate`. For information about setting up signatures and authorization through the API, go to [Signing AWS API Requests](#) in the *AWS General Reference*. For general information about using the Query API with IAM, go to [Making Query Requests](#) in the *Using IAM* guide.

Request Syntax

```
signing_certificate = iam.create_signing_certificate(
    UserName='string',
    CertificateBody='string'
)
```

Parameters

- **UserName** (*string*) – The name of the user the signing certificate is for.
- **CertificateBody** (*string*) – **[REQUIRED]**

The contents of the signing certificate.

Return type `iam.SigningCertificate`

Returns SigningCertificate resource

create_user (***kwargs*)

Creates a new user for your AWS account.

For information about limitations on the number of users you can create, see [Limitations on IAM Entities](#) in the *IAM User Guide*.

Request Syntax

```
user = iam.create_user(
    Path='string',
    UserName='string'
)
```

Parameters

- **Path** (*string*) – The path for the user name. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

This parameter is optional. If it is not included, it defaults to a slash (/).

- **UserName** (*string*) – **[REQUIRED]**

The name of the user to create.

Return type `iam.User`

Returns User resource

create_virtual_mfa_device (**kwargs)

Creates a new virtual MFA device for the AWS account. After creating the virtual MFA, use `EnableMFADevice` to attach the MFA device to an IAM user. For more information about creating and working with virtual MFA devices, go to [Using a Virtual MFA Device](#) in the *Using IAM* guide.

For information about limits on the number of MFA devices you can create, see [Limitations on Entities](#) in the *Using IAM* guide.

Warning: The seed information contained in the QR code and the Base32 string should be treated like any other secret access information, such as your AWS access keys or your passwords. After you provision your virtual device, you should ensure that the information is destroyed following secure procedures.

Request Syntax

```
virtual_mfa_device = iam.create_virtual_mfa_device(  
    Path='string',  
    VirtualMFADeviceName='string'  
)
```

Parameters

- **Path** (*string*) – The path for the virtual MFA device. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

This parameter is optional. If it is not included, it defaults to a slash (/).

- **VirtualMFADeviceName** (*string*) – **[REQUIRED]**

The name of the virtual MFA device. Use with path to uniquely identify a virtual MFA device.

Return type `iam.VirtualMfaDevice`

Returns VirtualMfaDevice resource

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

AccessKey (*user_name, id*)

Creates a AccessKey resource.:

```
access_key = iam.AccessKey('user_name', 'id')
```

Parameters

- **user_name** (*string*) – The AccessKey's user_name identifier. This **must** be set.
- **id** (*string*) – The AccessKey's id identifier. This **must** be set.

Return type `IAM.AccessKey`

Returns A AccessKey resource

AccessKeyPair (*user_name, id, secret*)

Creates a AccessKeyPair resource.:

```
access_key_pair = iam.AccessKeyPair('user_name', 'id', 'secret')
```

Parameters

- **user_name** (*string*) – The AccessKeyPair's user_name identifier. This **must** be set.
- **id** (*string*) – The AccessKeyPair's id identifier. This **must** be set.
- **secret** (*string*) – The AccessKeyPair's secret identifier. This **must** be set.

Return type `IAM.AccessKeyPair`

Returns A AccessKeyPair resource

AccountPasswordPolicy()

Creates a AccountPasswordPolicy resource.:

```
account_password_policy = iam.AccountPasswordPolicy()
```

Return type `IAM.AccountPasswordPolicy`

Returns A AccountPasswordPolicy resource

AccountSummary()

Creates a AccountSummary resource.:

```
account_summary = iam.AccountSummary()
```

Return type `IAM.AccountSummary`

Returns A AccountSummary resource

AssumeRolePolicy(role_name)

Creates a AssumeRolePolicy resource.:

```
assume_role_policy = iam.AssumeRolePolicy('role_name')
```

Parameters `role_name` (*string*) – The AssumeRolePolicy’s `role_name` identifier. This **must** be set.

Return type `IAM.AssumeRolePolicy`

Returns A AssumeRolePolicy resource

CurrentUser()

Creates a CurrentUser resource.:

```
current_user = iam.CurrentUser()
```

Return type `IAM.CurrentUser`

Returns A CurrentUser resource

Group(name)

Creates a Group resource.:

```
group = iam.Group('name')
```

Parameters `name` (*string*) – The Group’s `name` identifier. This **must** be set.

Return type `IAM.Group`

Returns A Group resource

GroupPolicy(group_name, name)

Creates a GroupPolicy resource.:

```
group_policy = iam.GroupPolicy('group_name', 'name')
```

Parameters

- **group_name** (*string*) – The GroupPolicy’s `group_name` identifier. This **must** be set.
- **name** (*string*) – The GroupPolicy’s `name` identifier. This **must** be set.

Return type `IAM.GroupPolicy`

Returns A GroupPolicy resource

InstanceProfile (*name*)

Creates a InstanceProfile resource.:

```
instance_profile = iam.InstanceProfile('name')
```

Parameters *name* (*string*) – The InstanceProfile’s name identifier. This **must** be set.

Return type `IAM.InstanceProfile`

Returns A InstanceProfile resource

LoginProfile (*user_name*)

Creates a LoginProfile resource.:

```
login_profile = iam.LoginProfile('user_name')
```

Parameters *user_name* (*string*) – The LoginProfile’s user_name identifier. This **must** be set.

Return type `IAM.LoginProfile`

Returns A LoginProfile resource

MfaDevice (*user_name*, *serial_number*)

Creates a MfaDevice resource.:

```
mfa_device = iam.MfaDevice('user_name', 'serial_number')
```

Parameters

- **user_name** (*string*) – The MfaDevice’s user_name identifier. This **must** be set.
- **serial_number** (*string*) – The MfaDevice’s serial_number identifier. This **must** be set.

Return type `IAM.MfaDevice`

Returns A MfaDevice resource

Policy (*policy_arn*)

Creates a Policy resource.:

```
policy = iam.Policy('policy_arn')
```

Parameters *policy_arn* (*string*) – The Policy’s policy_arn identifier. This **must** be set.

Return type `IAM.Policy`

Returns A Policy resource

PolicyVersion (*arn*, *version_id*)

Creates a PolicyVersion resource.:

```
policy_version = iam.PolicyVersion('arn', 'version_id')
```

Parameters

- **arn** (*string*) – The PolicyVersion’s arn identifier. This **must** be set.
- **version_id** (*string*) – The PolicyVersion’s version_id identifier. This **must** be set.

Return type `IAM.PolicyVersion`

Returns A PolicyVersion resource

Role (*name*)

Creates a Role resource.:

```
role = iam.Role('name')
```

Parameters *name* (*string*) – The Role’s name identifier. This **must** be set.

Return type `IAM.Role`

Returns A Role resource

RolePolicy (*role_name*, *name*)

Creates a RolePolicy resource.:

```
role_policy = iam.RolePolicy('role_name', 'name')
```

Parameters

- **role_name** (*string*) – The RolePolicy’s role_name identifier. This **must** be set.
- **name** (*string*) – The RolePolicy’s name identifier. This **must** be set.

Return type `IAM.RolePolicy`

Returns A RolePolicy resource

SamlProvider (*arn*)

Creates a SamlProvider resource.:

```
saml_provider = iam.SamlProvider('arn')
```

Parameters *arn* (*string*) – The SamlProvider’s arn identifier. This **must** be set.

Return type `IAM.SamlProvider`

Returns A SamlProvider resource

ServerCertificate (*name*)

Creates a ServerCertificate resource.:

```
server_certificate = iam.ServerCertificate('name')
```

Parameters *name* (*string*) – The ServerCertificate’s name identifier. This **must** be set.

Return type `IAM.ServerCertificate`

Returns A ServerCertificate resource

SigningCertificate (*user_name*, *id*)

Creates a SigningCertificate resource.:

```
signing_certificate = iam.SigningCertificate('user_name', 'id')
```

Parameters

- **user_name** (*string*) – The SigningCertificate’s user_name identifier. This **must** be set.
- **id** (*string*) – The SigningCertificate’s id identifier. This **must** be set.

Return type `IAM.SigningCertificate`

Returns A SigningCertificate resource

User (*name*)

Creates a User resource.:

```
user = iam.User('name')
```

Parameters *name* (*string*) – The User’s name identifier. This **must** be set.

Return type `IAM.User`

Returns A User resource

UserPolicy (*user_name*, *name*)

Creates a UserPolicy resource.:

```
user_policy = iam.UserPolicy('user_name', 'name')
```

Parameters

- **user_name** (*string*) – The UserPolicy’s user_name identifier. This **must** be set.
- **name** (*string*) – The UserPolicy’s name identifier. This **must** be set.

Return type `IAM.UserPolicy`

Returns A UserPolicy resource

VirtualMfaDevice (*serial_number*)

Creates a VirtualMfaDevice resource.:

```
virtual_mfa_device = iam.VirtualMfaDevice('serial_number')
```

Parameters **serial_number** (*string*) – The VirtualMfaDevice’s serial_number identifier. This **must** be set.

Return type `IAM.VirtualMfaDevice`

Returns A VirtualMfaDevice resource

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

groups

all ()

Creates an iterable of all Group resources in the collection.

Request Syntax

```
group_iterator = iam.groups.all()
```

Return type `list(iam.Group)`

Returns A list of Group resources

filter (***kwargs*)

Creates an iterable of all Group resources in the collection filtered by kwargs passed to method.

Request Syntax

```
group_iterator = iam.groups.filter(  
    PathPrefix='string',  
    Marker='string',  
    MaxItems=123  
)
```

Parameters

- **PathPrefix** (*string*) – The path prefix for filtering the results. For example, the prefix `/division_abc/subdivision_xyz/` gets all groups whose path starts with `/division_abc/subdivision_xyz/`.

This parameter is optional. If it is not included, it defaults to a slash (/), listing all groups.

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.

- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.Group)`

Returns A list of Group resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Group resources in the collection.

Request Syntax

```
group_iterator = iam.groups.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(iam.Group)`

Returns A list of Group resources

page_size (***kwargs*)

Creates an iterable of all Group resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
group_iterator = iam.groups.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(iam.Group)`

Returns A list of Group resources

instance_profiles

all ()

Creates an iterable of all InstanceProfile resources in the collection.

Request Syntax

```
instance_profile_iterator = iam.instance_profiles.all()
```

Return type `list(iam.InstanceProfile)`

Returns A list of InstanceProfile resources

filter (***kwargs*)

Creates an iterable of all InstanceProfile resources in the collection filtered by kwargs passed to method.

Request Syntax

```
instance_profile_iterator = iam.instance_profiles.filter(  
    PathPrefix='string',  
    Marker='string',  
    MaxItems=123  
)
```

Parameters

- **PathPrefix** (*string*) – The path prefix for filtering the results. For example, the prefix `/application_abc/component_xyz/` gets all instance profiles whose path starts with `/application_abc/component_xyz/`.

This parameter is optional. If it is not included, it defaults to a slash (`/`), listing all instance profiles.

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.InstanceProfile)`

Returns A list of `InstanceProfile` resources

limit (***kwargs*)

Creates an iterable up to a specified amount of `InstanceProfile` resources in the collection.

Request Syntax

```
instance_profile_iterator = iam.instance_profiles.limit(  
    count=123  
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(iam.InstanceProfile)`

Returns A list of `InstanceProfile` resources

page_size (***kwargs*)

Creates an iterable of all `InstanceProfile` resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
instance_profile_iterator = iam.instance_profiles.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(iam.InstanceProfile)`

Returns A list of `InstanceProfile` resources

policies**all()**

Creates an iterable of all Policy resources in the collection.

Request Syntax

```
policy_iterator = iam.policies.all()
```

Return type list(iam.Policy)

Returns A list of Policy resources

filter(kwargs)**

Creates an iterable of all Policy resources in the collection filtered by kwargs passed to method.

Request Syntax

```
policy_iterator = iam.policies.filter(
    Scope='All'|'AWS'|'Local',
    OnlyAttached=True|False,
    PathPrefix='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **Scope** (*string*) – The scope to use for filtering the results.

To list only AWS managed policies, set `Scope` to `AWS` . To list only the customer managed policies in your AWS account, set `Scope` to `Local` .

This parameter is optional. If it is not included, or if it is set to `All` , all policies are returned.

- **OnlyAttached** (*boolean*) – A flag to filter the results to only the attached policies.

When `OnlyAttached` is `true` , the returned list contains only the policies that are attached to a user, group, or role. When `OnlyAttached` is `false` , or when the parameter is not included, all policies are returned.

- **PathPrefix** (*string*) – The path prefix for filtering the results. This parameter is optional. If it is not included, it defaults to a slash (/), listing all policies.
- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true` .

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type list(iam.Policy)

Returns A list of Policy resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Policy resources in the collection.

Request Syntax

```
policy_iterator = iam.policies.limit(  
    count=123  
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type list(iam.Policy)

Returns A list of Policy resources

page_size (***kwargs*)

Creates an iterable of all Policy resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
policy_iterator = iam.policies.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type list(iam.Policy)

Returns A list of Policy resources

roles**all** ()

Creates an iterable of all Role resources in the collection.

Request Syntax

```
role_iterator = iam.roles.all()
```

Return type list(iam.Role)

Returns A list of Role resources

filter (***kwargs*)

Creates an iterable of all Role resources in the collection filtered by kwargs passed to method.

Request Syntax

```
role_iterator = iam.roles.filter(  
    PathPrefix='string',  
    Marker='string',  
    MaxItems=123  
)
```

Parameters

- **PathPrefix** (*string*) – The path prefix for filtering the results. For example, the prefix `/application_abc/component_xyz/` gets all roles whose path starts with `/application_abc/component_xyz/`.

This parameter is optional. If it is not included, it defaults to a slash (/), listing all roles.

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.Role)`

Returns A list of Role resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Role resources in the collection.

Request Syntax

```
role_iterator = iam.roles.limit(
    count=123
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(iam.Role)`

Returns A list of Role resources

page_size (***kwargs*)

Creates an iterable of all Role resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
role_iterator = iam.roles.page_size(
    count=123
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(iam.Role)`

Returns A list of Role resources

saml_providers

all ()

Creates an iterable of all SamlProvider resources in the collection.

Request Syntax

```
saml_provider_iterator = iam.saml_providers.all()
```

Return type `list(iam.SamlProvider)`

Returns A list of SamlProvider resources

filter()

Creates an iterable of all `SamlProvider` resources in the collection filtered by kwargs passed to method.

Request Syntax

```
saml_provider_iterator = iam.saml_providers.filter()
```

Return type `list(iam.SamlProvider)`

Returns A list of `SamlProvider` resources

limit(kwargs)**

Creates an iterable up to a specified amount of `SamlProvider` resources in the collection.

Request Syntax

```
saml_provider_iterator = iam.saml_providers.limit(  
    count=123  
)
```

Parameters `count (integer)` – The limit to the number of resources in the iterable.

Return type `list(iam.SamlProvider)`

Returns A list of `SamlProvider` resources

page_size(kwargs)**

Creates an iterable of all `SamlProvider` resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
saml_provider_iterator = iam.saml_providers.page_size(  
    count=123  
)
```

Parameters `count (integer)` – The number of items returned by each service call

Return type `list(iam.SamlProvider)`

Returns A list of `SamlProvider` resources

server_certificates**all()**

Creates an iterable of all `ServerCertificate` resources in the collection.

Request Syntax

```
server_certificate_iterator = iam.server_certificates.all()
```

Return type `list(iam.ServerCertificate)`

Returns A list of `ServerCertificate` resources

filter(kwargs)**

Creates an iterable of all `ServerCertificate` resources in the collection filtered by kwargs passed to method.

Request Syntax

```
server_certificate_iterator = iam.server_certificates.filter(
    PathPrefix='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **PathPrefix** (*string*) – The path prefix for filtering the results. For example: `/company/servercerts` would get all server certificates for which the path starts with `/company/servercerts`.

This parameter is optional. If it is not included, it defaults to a slash (/), listing all server certificates.

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.ServerCertificate)`

Returns A list of `ServerCertificate` resources

limit (***kwargs*)

Creates an iterable up to a specified amount of `ServerCertificate` resources in the collection.

Request Syntax

```
server_certificate_iterator = iam.server_certificates.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(iam.ServerCertificate)`

Returns A list of `ServerCertificate` resources

page_size (***kwargs*)

Creates an iterable of all `ServerCertificate` resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
server_certificate_iterator = iam.server_certificates.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(iam.ServerCertificate)`

Returns A list of `ServerCertificate` resources

users**all()**

Creates an iterable of all User resources in the collection.

Request Syntax

```
user_iterator = iam.users.all()
```

Return type `list(iam.User)`

Returns A list of User resources

filter(kwargs)**

Creates an iterable of all User resources in the collection filtered by kwargs passed to method.

Request Syntax

```
user_iterator = iam.users.filter(
    PathPrefix='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **PathPrefix** (*string*) – The path prefix for filtering the results. For example: `/division_abc/subdivision_xyz/`, which would get all user names whose path starts with `/division_abc/subdivision_xyz/`.

This parameter is optional. If it is not included, it defaults to a slash (/), listing all user names.

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.User)`

Returns A list of User resources

limit(kwargs)**

Creates an iterable up to a specified amount of User resources in the collection.

Request Syntax

```
user_iterator = iam.users.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(iam.User)`

Returns A list of User resources

page_size (***kwargs*)

Creates an iterable of all User resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
user_iterator = iam.users.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(iam.User)`

Returns A list of User resources

virtual_mfa_devices

all ()

Creates an iterable of all VirtualMfaDevice resources in the collection.

Request Syntax

```
virtual_mfa_device_iterator = iam.virtual_mfa_devices.all()
```

Return type `list(iam.VirtualMfaDevice)`

Returns A list of VirtualMfaDevice resources

filter (***kwargs*)

Creates an iterable of all VirtualMfaDevice resources in the collection filtered by kwargs passed to method.

Request Syntax

```
virtual_mfa_device_iterator = iam.virtual_mfa_devices.filter(
    AssignmentStatus='Assigned'|'Unassigned'|'Any',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **AssignmentStatus** (*string*) – The status (unassigned or assigned) of the devices to list. If you do not specify an `AssignmentStatus`, the action defaults to `Any` which lists both assigned and unassigned virtual MFA devices.
- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns

`true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.VirtualMfaDevice)`

Returns A list of `VirtualMfaDevice` resources

limit (***kwargs*)

Creates an iterable up to a specified amount of `VirtualMfaDevice` resources in the collection.

Request Syntax

```
virtual_mfa_device_iterator = iam.virtual_mfa_devices.limit(  
    count=123  
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(iam.VirtualMfaDevice)`

Returns A list of `VirtualMfaDevice` resources

page_size (***kwargs*)

Creates an iterable of all `VirtualMfaDevice` resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
virtual_mfa_device_iterator = iam.virtual_mfa_devices.page_size(  
    count=123  
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(iam.VirtualMfaDevice)`

Returns A list of `VirtualMfaDevice` resources

AccessKey

class `IAM.AccessKey` (*user_name*, *id*)

A resource representing an AWS Identity and Access Management (IAM) `AccessKey`:

```
import boto3
```

```
iam = boto3.resource('iam')  
access_key = iam.AccessKey('user_name', 'id')
```

Parameters

- **user_name** (*string*) – The `AccessKey`'s `user_name` identifier. This **must** be set.
- **id** (*string*) – The `AccessKey`'s `id` identifier. This **must** be set.

These are the resource's available identifiers:

- `user_name`
- `id`

These are the resource's available attributes:

- `access_key_id`
- `create_date`
- `status`

These are the resource's available actions:

- `activate()`
- `deactivate()`

- `delete()`

These are the resource's available sub-resources:

- `User()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

user_name

(*string*) The AccessKey's user_name identifier. This **must** be set.

id

(*string*) The AccessKey's id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

access_key_id

(*string*)

The ID for this access key.

create_date

(*datetime*)

The date when the access key was created.

status

(*string*)

The status of the access key. *Active* means the key is valid for API calls; *Inactive* means it is not.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

activate()

Changes the status of the specified access key from Active to Inactive, or vice versa. This action can be used to disable a user's key as part of a key rotation work flow.

If the `UserName` field is not specified, the `UserName` is determined implicitly based on the AWS access key ID used to sign the request. Because this action works for access keys under the AWS account, you can use this action to manage root credentials even if the AWS account has no associated users.

For information about rotating keys, see [Managing Keys and Certificates](#) in the *IAM User Guide*.

Request Syntax

```
response = access_key.activate()
```

Returns None

deactivate()

Changes the status of the specified access key from Active to Inactive, or vice versa. This action can be used to disable a user's key as part of a key rotation work flow.

If the `UserName` field is not specified, the `UserName` is determined implicitly based on the AWS access key ID used to sign the request. Because this action works for access keys under the AWS account, you can use this action to manage root credentials even if the AWS account has no associated users.

For information about rotating keys, see [Managing Keys and Certificates](#) in the *IAM User Guide*.

Request Syntax

```
response = access_key.deactivate()
```

Returns None

`delete()`

Deletes the access key associated with the specified user.

If you do not specify a user name, IAM determines the user name implicitly based on the AWS access key ID signing the request. Because this action works for access keys under the AWS account, you can use this action to manage root credentials even if the AWS account has no associated users.

Request Syntax

```
response = access_key.delete()
```

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

`User()`

Creates a User resource.:

```
user = access_key.User()
```

Return type `IAM.User`

Returns A User resource

AccessKeyPair

```
class IAM.AccessKeyPair (user_name, id, secret)
```

A resource representing an AWS Identity and Access Management (IAM) AccessKeyPair:

```
import boto3
```

```
iam = boto3.resource('iam')
access_key_pair = iam.AccessKeyPair('user_name', 'id', 'secret')
```

Parameters

- **user_name** (*string*) – The AccessKeyPair's user_name identifier. This **must** be set.
- **id** (*string*) – The AccessKeyPair's id identifier. This **must** be set.
- **secret** (*string*) – The AccessKeyPair's secret identifier. This **must** be set.

These are the resource's available identifiers:

- `user_name`
- `id`
- `secret`

These are the resource's available attributes:

- `access_key_id`
- `create_date`
- `secret_access_key`
- `status`

These are the resource's available actions:

- `activate()`
- `deactivate()`
- `delete()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

user_name

(*string*) The AccessKeyPair's user_name identifier. This **must** be set.

id

(*string*) The AccessKeyPair's id identifier. This **must** be set.

secret

(*string*) The AccessKeyPair's secret identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

access_key_id

(*string*)

The ID for this access key.

create_date

(*datetime*)

The date when the access key was created.

secret_access_key

(*string*)

The secret key used to sign requests.

status

(*string*)

The status of the access key. `Active` means the key is valid for API calls, while `Inactive` means it is not.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

activate()

Changes the status of the specified access key from `Active` to `Inactive`, or vice versa. This action can be used to disable a user's key as part of a key rotation work flow.

If the `UserName` field is not specified, the `UserName` is determined implicitly based on the AWS access key ID used to sign the request. Because this action works for access keys under the AWS account, you can use this action to manage root credentials even if the AWS account has no associated users.

For information about rotating keys, see [Managing Keys and Certificates](#) in the *IAM User Guide*.

Request Syntax

```
response = access_key_pair.activate()
```

Returns None

deactivate()

Changes the status of the specified access key from Active to Inactive, or vice versa. This action can be used to disable a user's key as part of a key rotation work flow.

If the `UserName` field is not specified, the `UserName` is determined implicitly based on the AWS access key ID used to sign the request. Because this action works for access keys under the AWS account, you can use this action to manage root credentials even if the AWS account has no associated users.

For information about rotating keys, see [Managing Keys and Certificates](#) in the *IAM User Guide*.

Request Syntax

```
response = access_key_pair.deactivate()
```

Returns None

delete()

Deletes the access key associated with the specified user.

If you do not specify a user name, IAM determines the user name implicitly based on the AWS access key ID signing the request. Because this action works for access keys under the AWS account, you can use this action to manage root credentials even if the AWS account has no associated users.

Request Syntax

```
response = access_key_pair.delete()
```

Returns None

AccountPasswordPolicy**class IAM.AccountPasswordPolicy**

A resource representing an AWS Identity and Access Management (IAM) AccountPasswordPolicy:

```
import boto3
```

```
iam = boto3.resource('iam')
account_password_policy = iam.AccountPasswordPolicy()
```

These are the resource's available attributes:

- `allow_users_to_change_password`
- `expire_passwords`
- `hard_expiry`
- `max_password_age`
- `minimum_password_length`
- `password_reuse_prevention`
- `require_lowercase_characters`
- `require_numbers`
- `require_symbols`
- `require_uppercase_characters`

These are the resource's available actions:

- `delete()`
- `load()`
- `reload()`
- `update()`

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

allow_users_to_change_password

(boolean)

Specifies whether IAM users are allowed to change their own password.

expire_passwords

(boolean)

Specifies whether IAM users are required to change their password after a specified number of days.

hard_expiry

(boolean)

Specifies whether IAM users are prevented from setting a new password after their password has expired.

max_password_age

(integer)

The number of days that an IAM user password is valid.

minimum_password_length

(integer)

Minimum length to require for IAM user passwords.

password_reuse_prevention

(integer)

Specifies the number of previous passwords that IAM users are prevented from reusing.

require_lowercase_characters

(boolean)

Specifies whether to require lowercase characters for IAM user passwords.

require_numbers

(boolean)

Specifies whether to require numbers for IAM user passwords.

require_symbols

(boolean)

Specifies whether to require symbols for IAM user passwords.

require_uppercase_characters

(boolean)

Specifies whether to require uppercase characters for IAM user passwords.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

delete()

Deletes the password policy for the AWS account.

Request Syntax

```
response = account_password_policy.delete()
```

Returns None

load()

Calls `iam.Client.get_account_password_policy()` to update the attributes of the Account-PasswordPolicy resource

Request Syntax

```
account_password_policy.load()
```

Returns None

reload()

Calls `iam.Client.get_account_password_policy()` to update the attributes of the Account-PasswordPolicy resource

Request Syntax

```
account_password_policy.reload()
```

Returns None

update (***kwargs*)

Updates the password policy settings for the AWS account.

Note: This action does not support partial updates. No parameters are required, but if you do not specify a parameter, that parameter's value reverts to its default value. See the **Request Parameters** section for each parameter's default value.

For more information about using a password policy, see [Managing an IAM Password Policy](#) in the *IAM User Guide*.

Request Syntax

```
response = account_password_policy.update(
    MinimumPasswordLength=123,
    RequireSymbols=True|False,
    RequireNumbers=True|False,
    RequireUppercaseCharacters=True|False,
    RequireLowercaseCharacters=True|False,
    AllowUsersToChangePassword=True|False,
    MaxPasswordAge=123,
    PasswordReusePrevention=123,
    HardExpiry=True|False
)
```

Parameters

- **MinimumPasswordLength** (*integer*) – The minimum number of characters allowed in an IAM user password.

Default value: 6

- **RequireSymbols** (*boolean*) – Specifies whether IAM user passwords must contain at least one of the following non-alphanumeric characters:

! @ # \$ % ^ & * () _ + - = [] { } | ' `

Default value: false

- **RequireNumbers** (*boolean*) – Specifies whether IAM user passwords must contain at least one numeric character (0 to 9).

Default value: false

- **RequireUppercaseCharacters** (*boolean*) – Specifies whether IAM user passwords must contain at least one uppercase character from the ISO basic Latin alphabet (A to Z).

Default value: false

- **RequireLowercaseCharacters** (*boolean*) – Specifies whether IAM user passwords must contain at least one lowercase character from the ISO basic Latin alphabet (a to z).

Default value: false

- **AllowUsersToChangePassword** (*boolean*) – Allows all IAM users in your account to use the AWS Management Console to change their own passwords. For more information, see [Letting IAM Users Change Their Own Passwords](#) in the *IAM User Guide*.

Default value: false

- **MaxPasswordAge** (*integer*) – The number of days that an IAM user password is valid. The default value of 0 means IAM user passwords never expire.

Default value: 0

- **PasswordReusePrevention** (*integer*) – Specifies the number of previous passwords that IAM users are prevented from reusing. The default value of 0 means IAM users are not prevented from reusing previous passwords.

Default value: 0

- **HardExpiry** (*boolean*) – Prevents IAM users from setting a new password after their password has expired.

Default value: false

Returns None

AccountSummary

class `IAM.AccountSummary`

A resource representing an AWS Identity and Access Management (IAM) AccountSummary:

```
import boto3
```

```
iam = boto3.resource('iam')
account_summary = iam.AccountSummary()
```

These are the resource's available attributes:

- `summary_map`

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

summary_map
(*dict*)

A set of key value pairs containing information about IAM entity usage and IAM quotas.

SummaryMap contains the following keys:

- AccessKeysPerUserQuota** The maximum number of active access keys allowed for each IAM user.
- AccountAccessKeysPresent** This value is 1 if the AWS account (root) has an access key, otherwise it is 0.
- AccountMFAEnabled** This value is 1 if the AWS account (root) has an MFA device assigned, otherwise it is 0.
- AccountSigningCertificatesPresent** This value is 1 if the AWS account (root) has a signing certificate, otherwise it is 0.
- AssumeRolePolicySizeQuota** The maximum allowed size for assume role policy documents (trust policies), in non-whitespace characters.
- AttachedPoliciesPerGroupQuota** The maximum number of managed policies that can be attached to an IAM group.
- AttachedPoliciesPerRoleQuota** The maximum number of managed policies that can be attached to an IAM role.
- AttachedPoliciesPerUserQuota** The maximum number of managed policies that can be attached to an IAM user.
- GroupPolicySizeQuota** The maximum allowed size for the aggregate of all inline policies embedded in an IAM group, in non-whitespace characters.
- Groups** The number of IAM groups in the AWS account.
- GroupsPerUserQuota** The maximum number of IAM groups each IAM user can belong to.
- GroupsQuota** The maximum number of IAM groups allowed in the AWS account.
- InstanceProfiles** The number of instance profiles in the AWS account.
- InstanceProfilesQuota** The maximum number of instance profiles allowed in the AWS account.
- MFADevices** The number of MFA devices in the AWS account, including those assigned and unsigned.
- MFADevicesInUse** The number of MFA devices that have been assigned to an IAM user or to the AWS account (root).
- Policies** The number of customer managed policies in the AWS account.
- PoliciesQuota** The maximum number of customer managed policies allowed in the AWS account.
- PolicySizeQuota** The maximum allowed size of a customer managed policy, in non-whitespace characters.
- PolicyVersionsInUse** The number of managed policies that are attached to IAM users, groups, or roles in the AWS account.
- PolicyVersionsInUseQuota** The maximum number of managed policies that can be attached to IAM users, groups, or roles in the AWS account.
- Providers** The number of identity providers in the AWS account.
- RolePolicySizeQuota** The maximum allowed size for the aggregate of all inline policies (access policies, not the trust policy) embedded in an IAM role, in non-whitespace characters.
- Roles** The number of IAM roles in the AWS account.
- RolesQuota** The maximum number of IAM roles allowed in the AWS account.
- ServerCertificates** The number of server certificates in the AWS account.
- ServerCertificatesQuota** The maximum number of server certificates allowed in the AWS account.
- SigningCertificatesPerUserQuota** The maximum number of X.509 signing certificates allowed for each IAM user.
- UserPolicySizeQuota** The maximum allowed size for the aggregate of all inline policies embedded in an IAM user, in non-whitespace characters.
- Users** The number of IAM users in the AWS account.
- UsersQuota** The maximum number of IAM users allowed in the AWS account.
- VersionsPerPolicyQuota** The maximum number of policy versions allowed for each managed policy.

AssumeRolePolicy

class `IAM.AssumeRolePolicy` (*role_name*)

A resource representing an AWS Identity and Access Management (IAM) AssumeRolePolicy:

```
import boto3
```

```
iam = boto3.resource('iam')
assume_role_policy = iam.AssumeRolePolicy('role_name')
```

Parameters `role_name` (*string*) – The AssumeRolePolicy’s `role_name` identifier. This **must** be set.

These are the resource’s available identifiers:

- `role_name`

These are the resource’s available actions:

- `update()`

These are the resource’s available sub-resources:

- `Role()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

role_name

(*string*) The AssumeRolePolicy’s `role_name` identifier. This **must** be set.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

update (***kwargs*)

Updates the policy that grants an entity permission to assume a role. For more information about roles, go to [Using Roles to Delegate Permissions and Federate Identities](#).

Request Syntax

```
response = assume_role_policy.update(
    PolicyDocument='string'
)
```

Parameters `PolicyDocument` (*string*) – [REQUIRED]

The policy that grants an entity permission to assume the role.

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource’s identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Role ()

Creates a Role resource.:

```
role = assume_role_policy.Role()
```

Return type `IAM.Role`

Returns A Role resource

CurrentUser

class `IAM.CurrentUser`

A resource representing an AWS Identity and Access Management (IAM) CurrentUser:

```
import boto3

iam = boto3.resource('iam')
current_user = iam.CurrentUser()
```

These are the resource's available attributes:

- `arn`
- `create_date`
- `password_last_used`
- `path`
- `user_id`
- `user_name`

These are the resource's available references:

- `user`

These are the resource's available collections:

- `access_keys`
- `mfa_devices`
- `signing_certificates`

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

`arn`

(string)

The Amazon Resource Name (ARN) that identifies the user. For more information about ARNs and how to use ARNs in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

`create_date`

(datetime)

The date and time, in [ISO 8601 date-time format](#) , when the user was created.

`password_last_used`

(datetime)

The date and time, in [ISO 8601 date-time format](#) , when the user's password was last used to sign in to an AWS website. For a list of AWS websites that capture a user's last sign-in time, see the [Credential Reports](#) topic in the *Using IAM* guide. If a password is used more than once in a five-minute span, only the first use is returned in this field. This field is null (not present) when:

- The user does not have a password
- The password exists but has never been used (at least not since IAM started tracking this information on October 20th, 2014)
- there is no sign-in data associated with the user

This value is returned only in the `GetUser` and `ListUsers` actions.

`path`

(string)

The path to the user. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

`user_id`

(string)

The stable and unique string identifying the user. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

user_name
(string)

The friendly name identifying the user.

References

References are related resource instances that have a belongs-to relationship. For more information about references refer to the [Resources Introduction Guide](#).

user
(User) The related user if set, otherwise None.

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

access_keys

all()
Creates an iterable of all AccessKey resources in the collection.

Request Syntax

```
access_key_iterator = current_user.access_keys.all()
```

Return type list(iam.AccessKey)
Returns A list of AccessKey resources

filter(**kwargs)
Creates an iterable of all AccessKey resources in the collection filtered by kwargs passed to method.

Request Syntax

```
access_key_iterator = current_user.access_keys.filter(
    UserName='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **UserName** (string) – The name of the user.
- **Marker** (string) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the Marker element in the response you received to inform the next call about where to start.
- **MaxItems** (integer) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the IsTruncated response element is true.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the IsTruncated response element returns true and Marker contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.AccessKey)`

Returns A list of `AccessKey` resources

limit (***kwargs*)

Creates an iterable up to a specified amount of `AccessKey` resources in the collection.

Request Syntax

```
access_key_iterator = current_user.access_keys.limit(  
    count=123  
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(iam.AccessKey)`

Returns A list of `AccessKey` resources

page_size (***kwargs*)

Creates an iterable of all `AccessKey` resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
access_key_iterator = current_user.access_keys.page_size(  
    count=123  
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(iam.AccessKey)`

Returns A list of `AccessKey` resources

mfa_devices

all ()

Creates an iterable of all `MfaDevice` resources in the collection.

Request Syntax

```
mfa_device_iterator = current_user.mfa_devices.all()
```

Return type `list(iam.MfaDevice)`

Returns A list of `MfaDevice` resources

filter (***kwargs*)

Creates an iterable of all `MfaDevice` resources in the collection filtered by `kwargs` passed to method.

Request Syntax

```
mfa_device_iterator = current_user.mfa_devices.filter(  
    UserName='string',  
    Marker='string',  
    MaxItems=123  
)
```

Parameters

- **UserName** (*string*) – The name of the user whose MFA devices you want to list.

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.MfaDevice)`

Returns A list of `MfaDevice` resources

limit (***kwargs*)

Creates an iterable up to a specified amount of `MfaDevice` resources in the collection.

Request Syntax

```
mfa_device_iterator = current_user.mfa_devices.limit(
    count=123
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(iam.MfaDevice)`

Returns A list of `MfaDevice` resources

page_size (***kwargs*)

Creates an iterable of all `MfaDevice` resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
mfa_device_iterator = current_user.mfa_devices.page_size(
    count=123
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(iam.MfaDevice)`

Returns A list of `MfaDevice` resources

signing_certificates

all ()

Creates an iterable of all `SigningCertificate` resources in the collection.

Request Syntax

```
signing_certificate_iterator = current_user.signing_certificates.all()
```

Return type `list(iam.SigningCertificate)`

Returns A list of `SigningCertificate` resources

filter (***kwargs*)

Creates an iterable of all `SigningCertificate` resources in the collection filtered by `kwargs` passed to method.

Request Syntax

```
signing_certificate_iterator = current_user.signing_certificates.filter(  
    UserName='string',  
    Marker='string',  
    MaxItems=123  
)
```

Parameters

- **UserName** (*string*) – The name of the user.
- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.SigningCertificate)`

Returns A list of `SigningCertificate` resources

limit (***kwargs*)

Creates an iterable up to a specified amount of `SigningCertificate` resources in the collection.

Request Syntax

```
signing_certificate_iterator = current_user.signing_certificates.limit(  
    count=123  
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(iam.SigningCertificate)`

Returns A list of `SigningCertificate` resources

page_size (***kwargs*)

Creates an iterable of all `SigningCertificate` resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
signing_certificate_iterator = current_user.signing_certificates.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(iam.SigningCertificate)`

Returns A list of `SigningCertificate` resources

Group

class `IAM.Group` (*name*)

A resource representing an AWS Identity and Access Management (IAM) Group:

```
import boto3

iam = boto3.resource('iam')
group = iam.Group('name')
```

Parameters `name` (*string*) – The Group’s name identifier. This **must** be set.

These are the resource’s available identifiers:

- `name`

These are the resource’s available attributes:

- `arn`
- `create_date`
- `group_id`
- `group_name`
- `path`

These are the resource’s available actions:

- `add_user()`
- `attach_policy()`
- `create()`
- `create_policy()`
- `delete()`
- `detach_policy()`
- `load()`
- `reload()`
- `remove_user()`
- `update()`

These are the resource’s available sub-resources:

- `Policy()`

These are the resource’s available collections:

- `attached_policies`
- `policies`
- `users`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

name

(*string*) The Group’s name identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

arn

(*string*)

The Amazon Resource Name (ARN) specifying the group. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

create_date

(*datetime*)

The date and time, in [ISO 8601 date-time format](#) , when the group was created.

group_id
(string)

The stable and unique string identifying the group. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

group_name
(string)

The friendly name that identifies the group.

path
(string)

The path to the group. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

add_user (**kwargs)

Adds the specified user to the specified group.

Request Syntax

```
response = group.add_user(  
    UserName='string'  
)
```

Parameters **UserName** (string) – [REQUIRED]

The name of the user to add.

Returns None

attach_policy (**kwargs)

Attaches the specified managed policy to the specified group.

You use this API to attach a managed policy to a group. To embed an inline policy in a group, use `PutGroupPolicy` .

For more information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide* .

Request Syntax

```
response = group.attach_policy(  
    PolicyArn='string'  
)
```

Parameters **PolicyArn** (string) – [REQUIRED]

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference* .

Returns None

create (**kwargs)

Creates a new group.

For information about the number of groups you can create, see [Limitations on IAM Entities](#) in the *IAM User Guide*.

Request Syntax

```
group = group.create(
    Path='string',
)
```

Parameters **Path** (*string*) – The path to the group. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

This parameter is optional. If it is not included, it defaults to a slash (/).

Return type `iam.Group`

Returns Group resource

`create_policy` (**kwargs)

Adds (or updates) an inline policy document that is embedded in the specified group.

A user can also have managed policies attached to it. To attach a managed policy to a group, use `AttachGroupPolicy`. To create a new managed policy, use `CreatePolicy`. For information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

For information about limits on the number of inline policies that you can embed in a group, see [Limitations on IAM Entities](#) in the *IAM User Guide*.

Note: Because policy documents can be large, you should use POST rather than GET when calling `PutGroupPolicy`. For general information about using the Query API with IAM, go to [Making Query Requests](#) in the *Using IAM* guide.

Request Syntax

```
group_policy = group.create_policy(
    PolicyName='string',
    PolicyDocument='string'
)
```

Parameters

- **PolicyName** (*string*) – [REQUIRED]
The name of the policy document.
- **PolicyDocument** (*string*) – [REQUIRED]
The policy document.

Return type `iam.GroupPolicy`

Returns GroupPolicy resource

`delete` ()

Deletes the specified group. The group must not contain any users or have any attached policies.

Request Syntax

```
response = group.delete()
```

Returns None

`detach_policy` (**kwargs)

Removes the specified managed policy from the specified group.

A group can also have inline policies embedded with it. To delete an inline policy, use the `DeleteGroupPolicy` API. For information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide* .

Request Syntax

```
response = group.detach_policy(  
    PolicyArn='string'  
)
```

Parameters `PolicyArn` (*string*) – [REQUIRED]

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference* .

Returns None

`load()`

Calls `iam.Client.get_group()` to update the attributes of the Group resource

Request Syntax

```
group.load()
```

Returns None

`reload()`

Calls `iam.Client.get_group()` to update the attributes of the Group resource

Request Syntax

```
group.reload()
```

Returns None

`remove_user` (***kwargs*)

Removes the specified user from the specified group.

Request Syntax

```
response = group.remove_user(  
    UserName='string'  
)
```

Parameters `UserName` (*string*) – [REQUIRED]

The name of the user to remove.

Returns None

`update` (***kwargs*)

Updates the name and/or the path of the specified group.

Warning: You should understand the implications of changing a group's path or name. For more information, see [Renaming Users and Groups](#) in the *IAM User Guide* .

Note: To change a group name the requester must have appropriate permissions on both the source object and the target object. For example, to change Managers to MGRs, the entity making the request must have permission on Managers and MGRs, or must have permission on all (*). For more information about permissions, see [Permissions and Policies](#) .

Request Syntax

```
group = group.update(
    NewPath='string',
    NewGroupName='string'
)
```

Parameters

- **NewPath** (*string*) – New path for the group. Only include this if changing the group's path.
- **NewGroupName** (*string*) – New name for the group. Only include this if changing the group's name.

Return type `iam.Group`

Returns Group resource

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Policy (*name*)

Creates a GroupPolicy resource.:

```
group_policy = group.Policy('name')
```

Parameters **name** (*string*) – The Policy's name identifier. This **must** be set.

Return type `IAM.GroupPolicy`

Returns A GroupPolicy resource

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

attached_policies

all ()

Creates an iterable of all Policy resources in the collection.

Request Syntax

```
policy_iterator = group.attached_policies.all()
```

Return type `list(iam.Policy)`

Returns A list of Policy resources

filter (***kwargs*)

Creates an iterable of all Policy resources in the collection filtered by kwargs passed to method.

Request Syntax

```
policy_iterator = group.attached_policies.filter(
    PathPrefix='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **PathPrefix** (*string*) – The path prefix for filtering the results. This parameter is optional. If it is not included, it defaults to a slash (/), listing all policies.
- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.Policy)`

Returns A list of Policy resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Policy resources in the collection.

Request Syntax

```
policy_iterator = group.attached_policies.limit(  
    count=123  
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(iam.Policy)`

Returns A list of Policy resources

page_size (***kwargs*)

Creates an iterable of all Policy resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
policy_iterator = group.attached_policies.page_size(  
    count=123  
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(iam.Policy)`

Returns A list of Policy resources

policies

all ()

Creates an iterable of all GroupPolicy resources in the collection.

Request Syntax

```
group_policy_iterator = group.policies.all()
```

Return type `list(iam.GroupPolicy)`

Returns A list of GroupPolicy resources

filter (**kwargs)

Creates an iterable of all GroupPolicy resources in the collection filtered by kwargs passed to method.

Request Syntax

```
group_policy_iterator = group.policies.filter(
    Marker='string',
    MaxItems=123
)
```

Parameters

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.GroupPolicy)`

Returns A list of GroupPolicy resources

limit (**kwargs)

Creates an iterable up to a specified amount of GroupPolicy resources in the collection.

Request Syntax

```
group_policy_iterator = group.policies.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(iam.GroupPolicy)`

Returns A list of GroupPolicy resources

page_size (**kwargs)

Creates an iterable of all GroupPolicy resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
group_policy_iterator = group.policies.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(iam.GroupPolicy)`

Returns A list of GroupPolicy resources

users

all()

Creates an iterable of all User resources in the collection.

Request Syntax

```
user_iterator = group.users.all()
```

Return type `list(iam.User)`

Returns A list of User resources

filter(kwargs)**

Creates an iterable of all User resources in the collection filtered by kwargs passed to method.

Request Syntax

```
user_iterator = group.users.filter(
    Marker='string',
    MaxItems=123
)
```

Parameters

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.User)`

Returns A list of User resources

limit(kwargs)**

Creates an iterable up to a specified amount of User resources in the collection.

Request Syntax

```
user_iterator = group.users.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(iam.User)`

Returns A list of User resources

page_size(kwargs)**

Creates an iterable of all User resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```

user_iterator = group.users.page_size(
    count=123
)

```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(iam.User)`

Returns A list of User resources

GroupPolicy

class `IAM.GroupPolicy` (*group_name, name*)

A resource representing an AWS Identity and Access Management (IAM) GroupPolicy:

```
import boto3
```

```
iam = boto3.resource('iam')
group_policy = iam.GroupPolicy('group_name', 'name')
```

Parameters

- **group_name** (*string*) – The GroupPolicy’s `group_name` identifier. This **must** be set.
- **name** (*string*) – The GroupPolicy’s `name` identifier. This **must** be set.

These are the resource’s available identifiers:

- `group_name`
- `name`

These are the resource’s available attributes:

- `policy_document`
- `policy_name`

These are the resource’s available actions:

- `delete()`
- `load()`
- `put()`
- `reload()`

These are the resource’s available sub-resources:

- `Group()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

`group_name`

(*string*) The GroupPolicy’s `group_name` identifier. This **must** be set.

`name`

(*string*) The GroupPolicy’s `name` identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

`policy_document`

(*string*)

The policy document.

`policy_name`

(*string*)

The name of the policy.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

`delete()`

Deletes the specified inline policy that is embedded in the specified group.

A group can also have managed policies attached to it. To detach a managed policy from a group, use `DetachGroupPolicy`. For more information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = group_policy.delete()
```

Returns None

`load()`

Calls `iam.Client.get_group_policy()` to update the attributes of the `GroupPolicy` resource

Request Syntax

```
group_policy.load()
```

Returns None

`put(**kwargs)`

Adds (or updates) an inline policy document that is embedded in the specified group.

A user can also have managed policies attached to it. To attach a managed policy to a group, use `AttachGroupPolicy`. To create a new managed policy, use `CreatePolicy`. For information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

For information about limits on the number of inline policies that you can embed in a group, see [Limitations on IAM Entities](#) in the *IAM User Guide*.

Note: Because policy documents can be large, you should use POST rather than GET when calling `PutGroupPolicy`. For general information about using the Query API with IAM, go to [Making Query Requests](#) in the *Using IAM* guide.

Request Syntax

```
response = group_policy.put(
    PolicyDocument='string'
)
```

Parameters `PolicyDocument` (*string*) – [REQUIRED]

The policy document.

Returns None

`reload()`

Calls `iam.Client.get_group_policy()` to update the attributes of the `GroupPolicy` resource

Request Syntax

```
group_policy.reload()
```

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Group()

Creates a Group resource.:

```
group = group_policy.Group()
```

Return type `IAM.Group`

Returns A Group resource

InstanceProfile

class `IAM.InstanceProfile(name)`

A resource representing an AWS Identity and Access Management (IAM) InstanceProfile:

```
import boto3
```

```
iam = boto3.resource('iam')
instance_profile = iam.InstanceProfile('name')
```

Parameters `name` (*string*) – The InstanceProfile's name identifier. This **must** be set.

These are the resource's available identifiers:

- `name`

These are the resource's available attributes:

- `arn`
- `create_date`
- `instance_profile_id`
- `instance_profile_name`
- `path`
- `roles_attribute`

These are the resource's available references:

- `roles`

These are the resource's available actions:

- `add_role()`
- `delete()`
- `load()`
- `reload()`
- `remove_role()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

name

(*string*) The InstanceProfile's name identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

arn*(string)*

The Amazon Resource Name (ARN) specifying the instance profile. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

create_date*(datetime)*

The date when the instance profile was created.

instance_profile_id*(string)*

The stable and unique string identifying the instance profile. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

instance_profile_name*(string)*

The name identifying the instance profile.

path*(string)*

The path to the instance profile. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

roles_attribute*(list)*

The role associated with the instance profile.

References

References are related resource instances that have a belongs-to relationship. For more information about references refer to the [Resources Introduction Guide](#).

roles

(Role) The related roles if set, otherwise None.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

add_role (**kwargs)

Adds the specified role to the specified instance profile. For more information about roles, go to [Working with Roles](#) . For more information about instance profiles, go to [About Instance Profiles](#) .

Request Syntax

```
response = instance_profile.add_role(
    RoleName='string'
)
```

Parameters **RoleName** (*string*) – [REQUIRED]

The name of the role to add.

Returns None**delete** ()

Deletes the specified instance profile. The instance profile must not have an associated role.

Warning: Make sure you do not have any Amazon EC2 instances running with the instance profile you are about to delete. Deleting a role or instance profile that is associated with a running instance will break any applications running on the instance.

For more information about instance profiles, go to [About Instance Profiles](#) .

Request Syntax

```
response = instance_profile.delete()
```

Returns None

load()

Calls `iam.Client.get_instance_profile()` to update the attributes of the InstanceProfile resource

Request Syntax

```
instance_profile.load()
```

Returns None

reload()

Calls `iam.Client.get_instance_profile()` to update the attributes of the InstanceProfile resource

Request Syntax

```
instance_profile.reload()
```

Returns None

remove_role(kwargs)**

Removes the specified role from the specified instance profile.

Warning: Make sure you do not have any Amazon EC2 instances running with the role you are about to remove from the instance profile. Removing a role from an instance profile that is associated with a running instance will break any applications running on the instance.

For more information about roles, go to [Working with Roles](#) . For more information about instance profiles, go to [About Instance Profiles](#) .

Request Syntax

```
response = instance_profile.remove_role(
    RoleName='string'
)
```

Parameters `RoleName` (*string*) – **[REQUIRED]**

The name of the role to remove.

Returns None

LoginProfile

class `IAM.LoginProfile` (*user_name*)

A resource representing an AWS Identity and Access Management (IAM) LoginProfile:

```
import boto3

iam = boto3.resource('iam')
login_profile = iam.LoginProfile('user_name')
```

Parameters `user_name` (*string*) – The LoginProfile’s `user_name` identifier. This **must** be set.

These are the resource’s available identifiers:

- `user_name`

These are the resource’s available attributes:

- `create_date`
- `password_reset_required`

These are the resource’s available actions:

- `create()`
- `delete()`
- `load()`
- `reload()`
- `update()`

These are the resource’s available sub-resources:

- `User()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

user_name

(*string*) The LoginProfile’s `user_name` identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

create_date

(*datetime*)

The date when the password for the user was created.

password_reset_required

(*boolean*)

Specifies whether the user is required to set a new password on next sign-in.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

create (***kwargs*)

Creates a password for the specified user, giving the user the ability to access AWS services through the AWS Management Console. For more information about managing passwords, see [Managing Passwords](#) in the *Using IAM* guide.

Request Syntax

```
login_profile = login_profile.create(
    Password='string',
    PasswordResetRequired=True|False
)
```

Parameters

- **Password** (*string*) – [REQUIRED]
The new password for the user.
- **PasswordResetRequired** (*boolean*) – Specifies whether the user is required to set a new password on next sign-in.

Return type `iam.LoginProfile`

Returns LoginProfile resource

delete()

Deletes the password for the specified user, which terminates the user’s ability to access AWS services through the AWS Management Console.

Warning: Deleting a user’s password does not prevent a user from accessing IAM through the command line interface or the API. To prevent all user access you must also either make the access key inactive or delete it. For more information about making keys inactive or deleting them, see [UpdateAccessKey](#) and [DeleteAccessKey](#) .

Request Syntax

```
response = login_profile.delete()
```

Returns None

load()

Calls `iam.Client.get_login_profile()` to update the attributes of the LoginProfile resource

Request Syntax

```
login_profile.load()
```

Returns None

reload()

Calls `iam.Client.get_login_profile()` to update the attributes of the LoginProfile resource

Request Syntax

```
login_profile.reload()
```

Returns None

update(kwargs)**

Changes the password for the specified user.

Users can change their own passwords by calling `ChangePassword` . For more information about modifying passwords, see [Managing Passwords](#) in the *IAM User Guide* .

Request Syntax

```
response = login_profile.update(  
    Password='string',  
    PasswordResetRequired=True|False  
)
```

Parameters

- **Password** (*string*) – The new password for the specified user.
- **PasswordResetRequired** (*boolean*) – Require the specified user to set a new password on next sign-in.

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

User()

Creates a User resource.:

```
user = login_profile.User()
```

Return type `IAM.User`**Returns** A User resource**MfaDevice****class** `IAM.MfaDevice` (*user_name*, *serial_number*)

A resource representing an AWS Identity and Access Management (IAM) MfaDevice:

```
import boto3
```

```
iam = boto3.resource('iam')  
mfa_device = iam.MfaDevice('user_name', 'serial_number')
```

Parameters

- **user_name** (*string*) – The MfaDevice's user_name identifier. This **must** be set.
- **serial_number** (*string*) – The MfaDevice's serial_number identifier. This **must** be set.

These are the resource's available identifiers:

- `user_name`
- `serial_number`

These are the resource's available attributes:

- `enable_date`

These are the resource's available actions:

- `associate()`
- `disassociate()`
- `resync()`

These are the resource's available sub-resources:

- `User()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

user_name

(*string*) The MfaDevice's user_name identifier. This **must** be set.

serial_number

(*string*) The MfaDevice's serial_number identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

enable_date

(*datetime*)

The date when the MFA device was enabled for the user.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

associate (***kwargs*)

Enables the specified MFA device and associates it with the specified user name. When enabled, the MFA device is required for every subsequent login by the user name associated with the device.

Request Syntax

```
response = mfa_device.associate(
    AuthenticationCode1='string',
    AuthenticationCode2='string'
)
```

Parameters

- **AuthenticationCode1** (*string*) – [REQUIRED]
An authentication code emitted by the device.
- **AuthenticationCode2** (*string*) – [REQUIRED]
A subsequent authentication code emitted by the device.

Returns None**disassociate** ()

Deactivates the specified MFA device and removes it from association with the user name for which it was originally enabled.

For more information about creating and working with virtual MFA devices, go to [Using a Virtual MFA Device](#) in the *Using IAM* guide.

Request Syntax

```
response = mfa_device.disassociate()
```

Returns None**resync** (***kwargs*)

Synchronizes the specified MFA device with AWS servers.

For more information about creating and working with virtual MFA devices, go to [Using a Virtual MFA Device](#) in the *Using IAM* guide.

Request Syntax

```
response = mfa_device.resync(
    AuthenticationCode1='string',
```

```
AuthenticationCode2='string'  
)
```

Parameters

- **AuthenticationCode1** (*string*) – [REQUIRED]
An authentication code emitted by the device.
- **AuthenticationCode2** (*string*) – [REQUIRED]
A subsequent authentication code emitted by the device.

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

User ()

Creates a User resource.:

```
user = mfa_device.User()
```

Return type `IAM.User`**Returns** A User resource**Policy****class** `IAM.Policy` (*arn*)

A resource representing an AWS Identity and Access Management (IAM) Policy:

```
import boto3
```

```
iam = boto3.resource('iam')  
policy = iam.Policy('arn')
```

Parameters **arn** (*string*) – The Policy's arn identifier. This **must** be set.

These are the resource's available identifiers:

- `arn`

These are the resource's available attributes:

- `attachment_count`
- `create_date`
- `default_version_id`
- `description`
- `is_attachable`
- `path`
- `policy_id`
- `policy_name`
- `update_date`

These are the resource's available references:

- `default_version`

These are the resource's available actions:

- `attach_group()`
- `attach_role()`
- `attach_user()`
- `create_version()`

- `delete()`
- `detach_group()`
- `detach_role()`
- `detach_user()`
- `load()`
- `reload()`

These are the resource's available collections:

- `attached_groups`
- `attached_roles`
- `attached_users`
- `versions`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

arn

(*string*) The Policy's arn identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

attachment_count

(*integer*)

The number of entities (users, groups, and roles) that the policy is attached to.

create_date

(*datetime*)

The date and time, in [ISO 8601 date-time format](#) , when the policy was created.

default_version_id

(*string*)

The identifier for the version of the policy that is set as the default version.

description

(*string*)

A friendly description of the policy.

This element is included in the response to the `GetPolicy` operation. It is not included in the response to the `ListPolicies` operation.

is_attachable

(*boolean*)

Specifies whether the policy can be attached to an IAM user, group, or role.

path

(*string*)

The path to the policy.

For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

policy_id

(*string*)

The stable and unique string identifying the policy.

For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

policy_name

(string)

The friendly name (not ARN) identifying the policy.

update_date

(datetime)

The date and time, in [ISO 8601 date-time format](#) , when the policy was last updated.

When a policy has only one version, this field contains the date and time when the policy was created.

When a policy has more than one version, this field contains the date and time when the most recent policy version was created.

References

References are related resource instances that have a belongs-to relationship. For more information about references refer to the [Resources Introduction Guide](#).

default_version

(PolicyVersion) The related default_version if set, otherwise None.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

attach_group (**kwargs)

Attaches the specified managed policy to the specified group.

You use this API to attach a managed policy to a group. To embed an inline policy in a group, use `PutGroupPolicy` .

For more information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide* .

Request Syntax

```
response = policy.attach_group(  
    GroupName='string',  
  
)
```

Parameters GroupName (string) – [REQUIRED]

The name (friendly name, not ARN) of the group to attach the policy to.

Returns None**attach_role** (**kwargs)

Attaches the specified managed policy to the specified role.

When you attach a managed policy to a role, the managed policy is used as the role's access (permissions) policy. You cannot use a managed policy as the role's trust policy. The role's trust policy is created at the same time as the role, using `CreateRole` . You can update a role's trust policy using `UpdateAssumeRolePolicy` .

Use this API to attach a managed policy to a role. To embed an inline policy in a role, use `PutRolePolicy` . For more information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide* .

Request Syntax


```
response = policy.attach_role(
    RoleName='string',
)

```

Parameters `RoleName` (*string*) – [REQUIRED]

The name (friendly name, not ARN) of the role to attach the policy to.

Returns None

attach_user (***kwargs*)

Attaches the specified managed policy to the specified user.

You use this API to attach a managed policy to a user. To embed an inline policy in a user, use `PutUserPolicy`.

For more information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = policy.attach_user(
    UserName='string',
)

```

Parameters `UserName` (*string*) – [REQUIRED]

The name (friendly name, not ARN) of the user to attach the policy to.

Returns None

create_version (***kwargs*)

Creates a new version of the specified managed policy. To update a managed policy, you create a new policy version. A managed policy can have up to five versions. If the policy has five versions, you must delete an existing version using `DeletePolicyVersion` before you create a new version.

Optionally, you can set the new version as the policy's default version. The default version is the operative version; that is, the version that is in effect for the IAM users, groups, and roles that the policy is attached to.

For more information about managed policy versions, see [Versioning for Managed Policies](#) in the *IAM User Guide*.

Request Syntax

```
policy_version = policy.create_version(
    PolicyDocument='string',
    SetAsDefault=True|False
)

```

Parameters

- **PolicyDocument** (*string*) – [REQUIRED]

The policy document.

- **SetAsDefault** (*boolean*) – Specifies whether to set this version as the policy's default version.

When this parameter is `true`, the new policy version becomes the operative version; that is, the version that is in effect for the IAM users, groups, and roles that the policy is attached to.

For more information about managed policy versions, see [Versioning for Managed Policies](#) in the *IAM User Guide* .

Return type `iam.PolicyVersion`

Returns PolicyVersion resource

delete()

Deletes the specified managed policy.

Before you can delete a managed policy, you must detach the policy from all users, groups, and roles that it is attached to, and you must delete all of the policy's versions. The following steps describe the process for deleting a managed policy:

- Detach the policy from all users, groups, and roles that the policy is attached to, using the `DetachUserPolicy` , `DetachGroupPolicy` , or `DetachRolePolicy` APIs. To list all the users, groups, and roles that a policy is attached to, use `ListEntitiesForPolicy` .
- Delete all versions of the policy using `DeletePolicyVersion` . To list the policy's versions, use `ListPolicyVersions` . You cannot use `DeletePolicyVersion` to delete the version that is marked as the default version. You delete the policy's default version in the next step of the process.
- Delete the policy (this automatically deletes the policy's default version) using this API.

For information about managed policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide* .

Request Syntax

```
response = policy.delete()
```

Returns None

detach_group(kwargs)**

Removes the specified managed policy from the specified group.

A group can also have inline policies embedded with it. To delete an inline policy, use the `DeleteGroupPolicy` API. For information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide* .

Request Syntax

```
response = policy.detach_group(  
    GroupName='string',  
)
```

Parameters `GroupName` (*string*) – [REQUIRED]

The name (friendly name, not ARN) of the group to detach the policy from.

Returns None

detach_role(kwargs)**

Removes the specified managed policy from the specified role.

A role can also have inline policies embedded with it. To delete an inline policy, use the `DeleteRolePolicy` API. For information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide* .

Request Syntax

```
response = policy.detach_role(  
    RoleName='string',  
)
```

Parameters `RoleName` (*string*) – [REQUIRED]

The name (friendly name, not ARN) of the role to detach the policy from.

Returns None

detach_user (***kwargs*)

Removes the specified managed policy from the specified user.

A user can also have inline policies embedded with it. To delete an inline policy, use the `DeleteUserPolicy` API. For information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

.

Request Syntax

```
response = policy.detach_user(
    UserName='string',
)
```

Parameters `UserName` (*string*) – [REQUIRED]

The name (friendly name, not ARN) of the user to detach the policy from.

Returns None

load ()

Calls `iam.Client.get_policy()` to update the attributes of the Policy resource

Request Syntax

```
policy.load()
```

Returns None

reload ()

Calls `iam.Client.get_policy()` to update the attributes of the Policy resource

Request Syntax

```
policy.reload()
```

Returns None

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

attached_groups**all** ()

Creates an iterable of all Group resources in the collection.

Request Syntax

```
group_iterator = policy.attached_groups.all()
```

Return type `list(iam.Group)`

Returns A list of Group resources

filter (***kwargs*)

Creates an iterable of all Group resources in the collection filtered by kwargs passed to method.

Request Syntax

```
group_iterator = policy.attached_groups.filter(  
    PathPrefix='string',  
    Marker='string',  
    MaxItems=123  
)
```

Parameters

- **PathPrefix** (*string*) – The path prefix for filtering the results. This parameter is optional. If it is not included, it defaults to a slash (/), listing all entities.
- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.Group)`

Returns A list of Group resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Group resources in the collection.

Request Syntax

```
group_iterator = policy.attached_groups.limit(  
    count=123  
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(iam.Group)`

Returns A list of Group resources

page_size (***kwargs*)

Creates an iterable of all Group resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
group_iterator = policy.attached_groups.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(iam.Group)`

Returns A list of Group resources

attached_roles**all()**

Creates an iterable of all Role resources in the collection.

Request Syntax

```
role_iterator = policy.attached_roles.all()
```

Return type `list(iam.Role)`**Returns** A list of Role resources**filter(**kwargs)**

Creates an iterable of all Role resources in the collection filtered by kwargs passed to method.

Request Syntax

```
role_iterator = policy.attached_roles.filter(
    PathPrefix='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **PathPrefix** (*string*) – The path prefix for filtering the results. This parameter is optional. If it is not included, it defaults to a slash (/), listing all entities.
- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.Role)`**Returns** A list of Role resources**limit(**kwargs)**

Creates an iterable up to a specified amount of Role resources in the collection.

Request Syntax

```
role_iterator = policy.attached_roles.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.**Return type** `list(iam.Role)`**Returns** A list of Role resources

page_size (***kwargs*)

Creates an iterable of all Role resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
role_iterator = policy.attached_roles.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type list(iam.Role)

Returns A list of Role resources

attached_users**all** ()

Creates an iterable of all User resources in the collection.

Request Syntax

```
user_iterator = policy.attached_users.all()
```

Return type list(iam.User)

Returns A list of User resources

filter (***kwargs*)

Creates an iterable of all User resources in the collection filtered by kwargs passed to method.

Request Syntax

```
user_iterator = policy.attached_users.filter(  
    PathPrefix='string',  
    Marker='string',  
    MaxItems=123  
)
```

Parameters

- **PathPrefix** (*string*) – The path prefix for filtering the results. This parameter is optional. If it is not included, it defaults to a slash (/), listing all entities.
- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type list(iam.User)

Returns A list of User resources

limit (***kwargs*)

Creates an iterable up to a specified amount of User resources in the collection.

Request Syntax

```
user_iterator = policy.attached_users.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(iam.User)`

Returns A list of User resources

page_size (***kwargs*)

Creates an iterable of all User resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
user_iterator = policy.attached_users.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(iam.User)`

Returns A list of User resources

versions

all ()

Creates an iterable of all PolicyVersion resources in the collection.

Request Syntax

```
policy_version_iterator = policy.versions.all()
```

Return type `list(iam.PolicyVersion)`

Returns A list of PolicyVersion resources

filter (***kwargs*)

Creates an iterable of all PolicyVersion resources in the collection filtered by kwargs passed to method.

Request Syntax

```
policy_version_iterator = policy.versions.filter(
    Marker='string',
    MaxItems=123
)
```

Parameters

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.

- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.PolicyVersion)`

Returns A list of `PolicyVersion` resources

limit (***kwargs*)

Creates an iterable up to a specified amount of `PolicyVersion` resources in the collection.

Request Syntax

```
policy_version_iterator = policy.versions.limit(  
    count=123  
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(iam.PolicyVersion)`

Returns A list of `PolicyVersion` resources

page_size (***kwargs*)

Creates an iterable of all `PolicyVersion` resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
policy_version_iterator = policy.versions.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(iam.PolicyVersion)`

Returns A list of `PolicyVersion` resources

PolicyVersion

class `IAM.PolicyVersion` (*arn*, *version_id*)

A resource representing an AWS Identity and Access Management (IAM) `PolicyVersion`:

```
import boto3
```

```
iam = boto3.resource('iam')  
policy_version = iam.PolicyVersion('arn', 'version_id')
```

Parameters

- **arn** (*string*) – The `PolicyVersion`'s `arn` identifier. This **must** be set.
- **version_id** (*string*) – The `PolicyVersion`'s `version_id` identifier. This **must** be set.

These are the resource's available identifiers:

- `arn`
- `version_id`

These are the resource's available attributes:

- `create_date`
- `document`
- `is_default_version`

These are the resource's available actions:

- `delete()`
- `load()`
- `reload()`
- `set_as_default()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

arn

(*string*) The PolicyVersion's arn identifier. This **must** be set.

version_id

(*string*) The PolicyVersion's version_id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

create_date

(*datetime*)

The date and time, in [ISO 8601 date-time format](#) , when the policy version was created.

document

(*string*)

The policy document.

The policy document is returned in the response to the `GetPolicyVersion` and `GetAccountAuthorizationDetails` operations. It is not returned in the response to the `CreatePolicyVersion` or `ListPolicyVersions` operations.

is_default_version

(*boolean*)

Specifies whether the policy version is set as the policy's default version.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

delete()

Deletes the specified version of the specified managed policy.

You cannot delete the default version of a policy using this API. To delete the default version of a policy, use `DeletePolicy` . To find out which version of a policy is marked as the default version, use `ListPolicyVersions` .

For information about versions for managed policies, refer to [Versioning for Managed Policies](#) in the *IAM User Guide* .

Request Syntax

```
response = policy_version.delete()
```

Returns None

load()

Calls `iam.Client.get_policy_version()` to update the attributes of the PolicyVersion resource

Request Syntax

```
policy_version.load()
```

Returns None

reload()

Calls `iam.Client.get_policy_version()` to update the attributes of the PolicyVersion resource

Request Syntax

```
policy_version.reload()
```

Returns None

set_as_default()

Sets the specified version of the specified policy as the policy's default (operative) version.

This action affects all users, groups, and roles that the policy is attached to. To list the users, groups, and roles that the policy is attached to, use the `ListEntitiesForPolicy` API.

For information about managed policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = policy_version.set_as_default()
```

Returns None

Role

class `IAM.Role`(*name*)

A resource representing an AWS Identity and Access Management (IAM) Role:

```
import boto3
```

```
iam = boto3.resource('iam')
```

```
role = iam.Role('name')
```

Parameters *name* (*string*) – The Role's name identifier. This **must** be set.

These are the resource's available identifiers:

- `name`

These are the resource's available attributes:

- `arn`
- `assume_role_policy_document`
- `create_date`
- `path`
- `role_id`
- `role_name`

These are the resource's available actions:

- `attach_policy()`

- `delete()`
- `detach_policy()`
- `load()`
- `reload()`

These are the resource's available sub-resources:

- `AssumeRolePolicy()`
- `Policy()`

These are the resource's available collections:

- `attached_policies`
- `instance_profiles`
- `policies`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

name

(*string*) The Role's name identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

arn

(*string*)

The Amazon Resource Name (ARN) specifying the role. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

assume_role_policy_document

(*string*)

The policy that grants an entity permission to assume the role.

create_date

(*datetime*)

The date and time, in [ISO 8601 date-time format](#) , when the role was created.

path

(*string*)

The path to the role. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

role_id

(*string*)

The stable and unique string identifying the role. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

role_name

(*string*)

The friendly name that identifies the role.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

attach_policy (***kwargs*)

Attaches the specified managed policy to the specified role.

When you attach a managed policy to a role, the managed policy is used as the role's access (permissions) policy. You cannot use a managed policy as the role's trust policy. The role's trust policy is created at the same time as the role, using `CreateRole`. You can update a role's trust policy using `UpdateAssumeRolePolicy`.

Use this API to attach a managed policy to a role. To embed an inline policy in a role, use `PutRolePolicy`. For more information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = role.attach_policy(
    PolicyArn='string'
)
```

Parameters `PolicyArn` (*string*) – [REQUIRED]

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

Returns None

`delete()`

Deletes the specified role. The role must not have any policies attached. For more information about roles, go to [Working with Roles](#).

Warning: Make sure you do not have any Amazon EC2 instances running with the role you are about to delete. Deleting a role or instance profile that is associated with a running instance will break any applications running on the instance.

Request Syntax

```
response = role.delete()
```

Returns None

`detach_policy(**kwargs)`

Removes the specified managed policy from the specified role.

A role can also have inline policies embedded with it. To delete an inline policy, use the `DeleteRolePolicy` API. For information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = role.detach_policy(
    PolicyArn='string'
)
```

Parameters `PolicyArn` (*string*) – [REQUIRED]

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

Returns None

load()

Calls `iam.Client.get_role()` to update the attributes of the Role resource

Request Syntax

```
role.load()
```

Returns None

reload()

Calls `iam.Client.get_role()` to update the attributes of the Role resource

Request Syntax

```
role.reload()
```

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

AssumeRolePolicy()

Creates a AssumeRolePolicy resource.:

```
assume_role_policy = role.AssumeRolePolicy()
```

Return type `IAM.AssumeRolePolicy`

Returns A AssumeRolePolicy resource

Policy(name)

Creates a RolePolicy resource.:

```
role_policy = role.Policy('name')
```

Parameters `name (string)` – The Policy's name identifier. This **must** be set.

Return type `IAM.RolePolicy`

Returns A RolePolicy resource

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

attached_policies**all()**

Creates an iterable of all Policy resources in the collection.

Request Syntax

```
policy_iterator = role.attached_policies.all()
```

Return type `list(iam.Policy)`

Returns A list of Policy resources

filter (***kwargs*)

Creates an iterable of all Policy resources in the collection filtered by kwargs passed to method.

Request Syntax

```
policy_iterator = role.attached_policies.filter(  
    PathPrefix='string',  
    Marker='string',  
    MaxItems=123  
)
```

Parameters

- **PathPrefix** (*string*) – The path prefix for filtering the results. This parameter is optional. If it is not included, it defaults to a slash (/), listing all policies.
- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.Policy)`

Returns A list of Policy resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Policy resources in the collection.

Request Syntax

```
policy_iterator = role.attached_policies.limit(  
    count=123  
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(iam.Policy)`

Returns A list of Policy resources

page_size (***kwargs*)

Creates an iterable of all Policy resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
policy_iterator = role.attached_policies.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(iam.Policy)`

Returns A list of Policy resources

instance_profiles**all()**

Creates an iterable of all InstanceProfile resources in the collection.

Request Syntax

```
instance_profile_iterator = role.instance_profiles.all()
```

Return type list(iam.InstanceProfile)

Returns A list of InstanceProfile resources

filter(kwargs)**

Creates an iterable of all InstanceProfile resources in the collection filtered by kwargs passed to method.

Request Syntax

```
instance_profile_iterator = role.instance_profiles.filter(
    Marker='string',
    MaxItems=123
)
```

Parameters

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type list(iam.InstanceProfile)

Returns A list of InstanceProfile resources

limit(kwargs)**

Creates an iterable up to a specified amount of InstanceProfile resources in the collection.

Request Syntax

```
instance_profile_iterator = role.instance_profiles.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type list(iam.InstanceProfile)

Returns A list of InstanceProfile resources

page_size(kwargs)**

Creates an iterable of all InstanceProfile resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
instance_profile_iterator = role.instance_profiles.page_size(
    count=123
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(iam.InstanceProfile)`

Returns A list of InstanceProfile resources

policies

all()

Creates an iterable of all RolePolicy resources in the collection.

Request Syntax

```
role_policy_iterator = role.policies.all()
```

Return type `list(iam.RolePolicy)`

Returns A list of RolePolicy resources

filter (***kwargs*)

Creates an iterable of all RolePolicy resources in the collection filtered by kwargs passed to method.

Request Syntax

```
role_policy_iterator = role.policies.filter(
    Marker='string',
    MaxItems=123
)
```

Parameters

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.RolePolicy)`

Returns A list of RolePolicy resources

limit (***kwargs*)

Creates an iterable up to a specified amount of RolePolicy resources in the collection.

Request Syntax

```
role_policy_iterator = role.policies.limit(
    count=123
)
```


Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(iam.RolePolicy)`

Returns A list of RolePolicy resources

page_size (***kwargs*)

Creates an iterable of all RolePolicy resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
role_policy_iterator = role.policies.page_size(
    count=123
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(iam.RolePolicy)`

Returns A list of RolePolicy resources

RolePolicy

class `IAM.RolePolicy` (*role_name, name*)

A resource representing an AWS Identity and Access Management (IAM) RolePolicy:

```
import boto3
```

```
iam = boto3.resource('iam')
role_policy = iam.RolePolicy('role_name', 'name')
```

Parameters

- **role_name** (*string*) – The RolePolicy’s role_name identifier. This **must** be set.
- **name** (*string*) – The RolePolicy’s name identifier. This **must** be set.

These are the resource’s available identifiers:

- `role_name`
- `name`

These are the resource’s available attributes:

- `policy_document`
- `policy_name`

These are the resource’s available actions:

- `delete()`
- `load()`
- `put()`
- `reload()`

These are the resource’s available sub-resources:

- `Role()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

role_name

(*string*) The RolePolicy’s role_name identifier. This **must** be set.

name

(*string*) The RolePolicy’s name identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

policy_document

(string)

The policy document.

policy_name

(string)

The name of the policy.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

delete()

Deletes the specified inline policy that is embedded in the specified role.

A role can also have managed policies attached to it. To detach a managed policy from a role, use `DetachRolePolicy`. For more information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = role_policy.delete()
```

Returns None

load()

Calls `iam.Client.get_role_policy()` to update the attributes of the `RolePolicy` resource

Request Syntax

```
role_policy.load()
```

Returns None

put(kwargs)**

Adds (or updates) an inline policy document that is embedded in the specified role.

When you embed an inline policy in a role, the inline policy is used as the role's access (permissions) policy. The role's trust policy is created at the same time as the role, using `CreateRole`. You can update a role's trust policy using `UpdateAssumeRolePolicy`. For more information about roles, go to [Using Roles to Delegate Permissions and Federate Identities](#).

A role can also have a managed policy attached to it. To attach a managed policy to a role, use `AttachRolePolicy`. To create a new managed policy, use `CreatePolicy`. For information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

For information about limits on the number of inline policies that you can embed with a role, see [Limitations on IAM Entities](#) in the *IAM User Guide*.

Note: Because policy documents can be large, you should use POST rather than GET when calling `PutRolePolicy`. For general information about using the Query API with IAM, go to [Making Query Requests](#) in the *Using IAM* guide.

Request Syntax

```
response = role_policy.put(
    PolicyDocument='string'
)
```

Parameters **PolicyDocument** (*string*) – [REQUIRED]

The policy document.

Returns None

reload()

Calls `iam.Client.get_role_policy()` to update the attributes of the RolePolicy resource

Request Syntax

```
role_policy.reload()
```

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource’s identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Role()

Creates a Role resource.:

```
role = role_policy.Role()
```

Return type `IAM.Role`

Returns A Role resource

SamlProvider

class `IAM.SamlProvider` (*arn*)

A resource representing an AWS Identity and Access Management (IAM) SamlProvider:

```
import boto3
```

```
iam = boto3.resource('iam')
saml_provider = iam.SamlProvider('arn')
```

Parameters **arn** (*string*) – The SamlProvider’s arn identifier. This **must** be set.

These are the resource’s available identifiers:

- `arn`

These are the resource’s available attributes:

- `create_date`
- `saml_metadata_document`
- `valid_until`

These are the resource’s available actions:

- `delete()`
- `load()`
- `reload()`
- `update()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

arn

(*string*) The SamlProvider's arn identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

create_date

(*datetime*)

The date and time when the SAML provider was created.

saml_metadata_document

(*string*)

The XML metadata document that includes information about an identity provider.

valid_until

(*datetime*)

The expiration date and time for the SAML provider.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

delete()

Deletes a SAML provider.

Deleting the provider does not update any roles that reference the SAML provider as a principal in their trust policies. Any attempt to assume a role that references a SAML provider that has been deleted will fail.

Note: This operation requires [Signature Version 4](#) .

Request Syntax

```
response = saml_provider.delete()
```

Returns None

load()

Calls `iam.Client.get_saml_provider()` to update the attributes of the SamlProvider resource

Request Syntax

```
saml_provider.load()
```

Returns None

reload()

Calls `iam.Client.get_saml_provider()` to update the attributes of the SamlProvider resource

Request Syntax

```
saml_provider.reload()
```

Returns None

update (***kwargs*)

Updates the metadata document for an existing SAML provider.

Note: This operation requires [Signature Version 4](#).

Request Syntax

```
response = saml_provider.update(
    SAMLMetadataDocument='string',
)
```

Parameters **SAMLMetadataDocument** (*string*) – [REQUIRED]

An XML document generated by an identity provider (IdP) that supports SAML 2.0. The document includes the issuer’s name, expiration information, and keys that can be used to validate the SAML authentication response (assertions) that are received from the IdP. You must generate the metadata document using the identity management software that is used as your organization’s IdP.

Return type dict

Returns

Response Syntax

```
{
    'SAMLProviderArn': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful UpdateSAMLProvider request.

- **SAMLProviderArn** (*string*) –

The Amazon Resource Name (ARN) of the SAML provider that was updated.

ServerCertificate

class IAM.**ServerCertificate** (*name*)

A resource representing an AWS Identity and Access Management (IAM) ServerCertificate:

```
import boto3
```

```
iam = boto3.resource('iam')
server_certificate = iam.ServerCertificate('name')
```

Parameters **name** (*string*) – The ServerCertificate’s name identifier. This **must** be set.

These are the resource’s available identifiers:

- **name**

These are the resource’s available attributes:

- `certificate_body`
- `certificate_chain`
- `server_certificate_metadata`

These are the resource's available actions:

- `delete()`
- `load()`
- `reload()`
- `update()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

name

(*string*) The ServerCertificate's name identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

certificate_body

(*string*)

The contents of the public key certificate.

certificate_chain

(*string*)

The contents of the public key certificate chain.

server_certificate_metadata

(*dict*)

The meta information of the server certificate, such as its name, path, ID, and ARN.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

delete()

Deletes the specified server certificate.

Warning: If you are using a server certificate with Elastic Load Balancing, deleting the certificate could have implications for your application. If Elastic Load Balancing doesn't detect the deletion of bound certificates, it may continue to use the certificates. This could cause Elastic Load Balancing to stop accepting traffic. We recommend that you remove the reference to the certificate from Elastic Load Balancing before using this command to delete the certificate. For more information, go to [DeleteLoadBalancerListeners](#) in the *Elastic Load Balancing API Reference* .

Request Syntax

```
response = server_certificate.delete()
```

Returns None

load()

Calls `iam.Client.get_server_certificate()` to update the attributes of the ServerCertificate resource

Request Syntax

```
server_certificate.load()
```

Returns None**reload()**

Calls `iam.Client.get_server_certificate()` to update the attributes of the `ServerCertificate` resource

Request Syntax

```
server_certificate.reload()
```

Returns None**update(**kwargs)**

Updates the name and/or the path of the specified server certificate.

Warning: You should understand the implications of changing a server certificate's path or name. For more information, see [Managing Server Certificates](#) in the *IAM User Guide* .

Note: To change a server certificate name the requester must have appropriate permissions on both the source object and the target object. For example, to change the name from `ProductionCert` to `ProdCert`, the entity making the request must have permission on `ProductionCert` and `ProdCert`, or must have permission on all (*). For more information about permissions, see [Permissions and Policies](#) .

Request Syntax

```
server_certificate = server_certificate.update(
    NewPath='string',
    NewServerCertificateName='string'
)
```

Parameters

- **NewPath** (*string*) – The new path for the server certificate. Include this only if you are updating the server certificate's path.
- **NewServerCertificateName** (*string*) – The new name for the server certificate. Include this only if you are updating the server certificate's name. The name of the certificate cannot contain any spaces.

Return type `iam.ServerCertificate`**Returns** `ServerCertificate` resource**SigningCertificate**

class `IAM.SigningCertificate` (*user_name*, *id*)

A resource representing an AWS Identity and Access Management (IAM) `SigningCertificate`:

```
import boto3
```

```
iam = boto3.resource('iam')
signing_certificate = iam.SigningCertificate('user_name', 'id')
```

Parameters

- **user_name** (*string*) – The `SigningCertificate`'s `user_name` identifier. This **must** be set.

- **id** (*string*) – The SigningCertificate’s id identifier. This **must** be set.

These are the resource’s available identifiers:

- `user_name`
- `id`

These are the resource’s available attributes:

- `certificate_body`
- `certificate_id`
- `status`
- `upload_date`

These are the resource’s available actions:

- `activate()`
- `deactivate()`
- `delete()`

These are the resource’s available sub-resources:

- `User()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

user_name

(*string*) The SigningCertificate’s user_name identifier. This **must** be set.

id

(*string*) The SigningCertificate’s id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

certificate_body

(*string*)

The contents of the signing certificate.

certificate_id

(*string*)

The ID for the signing certificate.

status

(*string*)

The status of the signing certificate. `Active` means the key is valid for API calls, while `Inactive` means it is not.

upload_date

(*datetime*)

The date when the signing certificate was uploaded.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

activate()

Changes the status of the specified signing certificate from active to disabled, or vice versa. This action can be used to disable a user’s signing certificate as part of a certificate rotation work flow.

If the `UserName` field is not specified, the `UserName` is determined implicitly based on the AWS access key ID used to sign the request. Because this action works for access keys under the AWS account, you can use this action to manage root credentials even if the AWS account has no associated users.

Request Syntax

```
response = signing_certificate.activate()
```

Returns None

`deactivate()`

Changes the status of the specified signing certificate from active to disabled, or vice versa. This action can be used to disable a user's signing certificate as part of a certificate rotation work flow.

If the `UserName` field is not specified, the `UserName` is determined implicitly based on the AWS access key ID used to sign the request. Because this action works for access keys under the AWS account, you can use this action to manage root credentials even if the AWS account has no associated users.

Request Syntax

```
response = signing_certificate.deactivate()
```

Returns None

`delete()`

Deletes the specified signing certificate associated with the specified user.

If you do not specify a user name, IAM determines the user name implicitly based on the AWS access key ID signing the request. Because this action works for access keys under the AWS account, you can use this action to manage root credentials even if the AWS account has no associated users.

Request Syntax

```
response = signing_certificate.delete()
```

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

`User()`

Creates a User resource.:

```
user = signing_certificate.User()
```

Return type `IAM.User`

Returns A User resource

User

`class IAM.User(name)`

A resource representing an AWS Identity and Access Management (IAM) User:

```
import boto3
```

```
iam = boto3.resource('iam')
user = iam.User('name')
```

Parameters `name` (*string*) – The User’s name identifier. This **must** be set.

These are the resource’s available identifiers:

- `name`

These are the resource’s available attributes:

- `arn`
- `create_date`
- `password_last_used`
- `path`
- `user_id`
- `user_name`

These are the resource’s available actions:

- `add_group()`
- `attach_policy()`
- `create()`
- `create_access_key_pair()`
- `create_login_profile()`
- `create_policy()`
- `delete()`
- `detach_policy()`
- `enable_mfa()`
- `load()`
- `reload()`
- `remove_group()`
- `update()`

These are the resource’s available sub-resources:

- `AccessKey()`
- `LoginProfile()`
- `MfaDevice()`
- `Policy()`
- `SigningCertificate()`

These are the resource’s available collections:

- `access_keys`
- `attached_policies`
- `groups`
- `mfa_devices`
- `policies`
- `signing_certificates`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

name

(*string*) The User’s name identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

arn

(*string*)

The Amazon Resource Name (ARN) that identifies the user. For more information about ARNs and how to use ARNs in policies, see [IAM Identifiers](#) in the *Using IAM* guide.

create_date
(datetime)

The date and time, in [ISO 8601 date-time format](#) , when the user was created.

password_last_used
(datetime)

The date and time, in [ISO 8601 date-time format](#) , when the user's password was last used to sign in to an AWS website. For a list of AWS websites that capture a user's last sign-in time, see the [Credential Reports](#) topic in the *Using IAM* guide. If a password is used more than once in a five-minute span, only the first use is returned in this field. This field is null (not present) when:

- The user does not have a password
- The password exists but has never been used (at least not since IAM started tracking this information on October 20th, 2014)
- there is no sign-in data associated with the user

This value is returned only in the GetUser and ListUsers actions.

path
(string)

The path to the user. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

user_id
(string)

The stable and unique string identifying the user. For more information about IDs, see [IAM Identifiers](#) in the *Using IAM* guide.

user_name
(string)

The friendly name identifying the user.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

add_group (**kwargs)
Adds the specified user to the specified group.

Request Syntax

```
response = user.add_group(
    GroupName='string',
)
```

Parameters GroupName (string) – [REQUIRED]

The name of the group to update.

Returns None

attach_policy (**kwargs)
Attaches the specified managed policy to the specified user.

You use this API to attach a managed policy to a user. To embed an inline policy in a user, use PutUserPolicy .

For more information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = user.attach_policy(  
    PolicyArn='string'  
)
```

Parameters `PolicyArn` (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

Returns None

create (***kwargs*)

Creates a new user for your AWS account.

For information about limitations on the number of users you can create, see [Limitations on IAM Entities](#) in the *IAM User Guide*.

Request Syntax

```
user = user.create(  
    Path='string',  
)
```

Parameters `Path` (*string*) – The path for the user name. For more information about paths, see [IAM Identifiers](#) in the *Using IAM* guide.

This parameter is optional. If it is not included, it defaults to a slash (/).

Return type `iam.User`

Returns User resource

create_access_key_pair ()

Creates a new AWS secret access key and corresponding AWS access key ID for the specified user. The default status for new keys is `Active`.

If you do not specify a user name, IAM determines the user name implicitly based on the AWS access key ID signing the request. Because this action works for access keys under the AWS account, you can use this action to manage root credentials even if the AWS account has no associated users.

For information about limits on the number of keys you can create, see [Limitations on IAM Entities](#) in the *IAM User Guide*.

Warning: To ensure the security of your AWS account, the secret access key is accessible only during key and user creation. You must save the key (for example, in a text file) if you want to be able to access it again. If a secret key is lost, you can delete the access keys for the associated user and then create new keys.

Request Syntax

```
access_key_pair = user.create_access_key_pair()
```

Return type `iam.AccessKeyPair`

Returns AccessKeyPair resource

create_login_profile (***kwargs*)

Creates a password for the specified user, giving the user the ability to access AWS services through the

AWS Management Console. For more information about managing passwords, see [Managing Passwords](#) in the *Using IAM* guide.

Request Syntax

```
login_profile = user.create_login_profile(
    Password='string',
    PasswordResetRequired=True|False
)
```

Parameters

- **Password** (*string*) – [REQUIRED]
The new password for the user.
- **PasswordResetRequired** (*boolean*) – Specifies whether the user is required to set a new password on next sign-in.

Return type `iam.LoginProfile`

Returns LoginProfile resource

create_policy (***kwargs*)

Adds (or updates) an inline policy document that is embedded in the specified user.

A user can also have a managed policy attached to it. To attach a managed policy to a user, use `AttachUserPolicy`. To create a new managed policy, use `CreatePolicy`. For information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

For information about limits on the number of inline policies that you can embed in a user, see [Limitations on IAM Entities](#) in the *IAM User Guide*.

Note: Because policy documents can be large, you should use POST rather than GET when calling `PutUserPolicy`. For general information about using the Query API with IAM, go to [Making Query Requests](#) in the *Using IAM* guide.

Request Syntax

```
user_policy = user.create_policy(
    PolicyName='string',
    PolicyDocument='string'
)
```

Parameters

- **PolicyName** (*string*) – [REQUIRED]
The name of the policy document.
- **PolicyDocument** (*string*) – [REQUIRED]
The policy document.

Return type `iam.UserPolicy`

Returns UserPolicy resource

delete ()

Deletes the specified user. The user must not belong to any groups, have any keys or signing certificates, or have any attached policies.

Request Syntax

```
response = user.delete()
```

Returns None

detach_policy (**kwargs)

Removes the specified managed policy from the specified user.

A user can also have inline policies embedded with it. To delete an inline policy, use the `DeleteUserPolicy` API. For information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

.

Request Syntax

```
response = user.detach_policy(  
    PolicyArn='string'  
)
```

Parameters **PolicyArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN). ARNs are unique identifiers for AWS resources.

For more information about ARNs, go to [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

Returns None

enable_mfa (**kwargs)

Enables the specified MFA device and associates it with the specified user name. When enabled, the MFA device is required for every subsequent login by the user name associated with the device.

Request Syntax

```
mfa_device = user.enable_mfa(  
    SerialNumber='string',  
    AuthenticationCode1='string',  
    AuthenticationCode2='string'  
)
```

Parameters

- **SerialNumber** (*string*) – [REQUIRED]

The serial number that uniquely identifies the MFA device. For virtual MFA devices, the serial number is the device ARN.

- **AuthenticationCode1** (*string*) – [REQUIRED]

An authentication code emitted by the device.

- **AuthenticationCode2** (*string*) – [REQUIRED]

A subsequent authentication code emitted by the device.

Return type `iam.MfaDevice`

Returns MfaDevice resource

load()

Calls `iam.Client.get_user()` to update the attributes of the User resource

Request Syntax

```
user.load()
```

Returns None

reload()

Calls `iam.Client.get_user()` to update the attributes of the User resource

Request Syntax

```
user.reload()
```

Returns None

remove_group (***kwargs*)

Removes the specified user from the specified group.

Request Syntax

```
response = user.remove_group(
    GroupName='string',
)
```

Parameters **GroupName** (*string*) – [REQUIRED]

The name of the group to update.

Returns None

update (***kwargs*)

Updates the name and/or the path of the specified user.

Warning: You should understand the implications of changing a user’s path or name. For more information, see [Renaming Users and Groups](#) in the *IAM User Guide* .

Note: To change a user name the requester must have appropriate permissions on both the source object and the target object. For example, to change Bob to Robert, the entity making the request must have permission on Bob and Robert, or must have permission on all (*). For more information about permissions, see [Permissions and Policies](#) .

Request Syntax

```
user = user.update(
    NewPath='string',
    NewUserName='string'
)
```

Parameters

- **NewPath** (*string*) – New path for the user. Include this parameter only if you’re changing the user’s path.
- **NewUserName** (*string*) – New name for the user. Include this parameter only if you’re changing the user’s name.

Return type `iam.User`

Returns User resource

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource’s identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

AccessKey (*id*)

Creates a AccessKey resource.:

```
access_key = user.AccessKey('id')
```

Parameters **id** (*string*) – The AccessKey’s id identifier. This **must** be set.

Return type `IAM.AccessKey`

Returns A AccessKey resource

LoginProfile()

Creates a LoginProfile resource.:

```
login_profile = user.LoginProfile()
```

Return type `IAM.LoginProfile`

Returns A LoginProfile resource

MfaDevice (*serial_number*)

Creates a MfaDevice resource.:

```
mfa_device = user.MfaDevice('serial_number')
```

Parameters **serial_number** (*string*) – The MfaDevice’s serial_number identifier. This **must** be set.

Return type `IAM.MfaDevice`

Returns A MfaDevice resource

Policy (*name*)

Creates a UserPolicy resource.:

```
user_policy = user.Policy('name')
```

Parameters **name** (*string*) – The Policy’s name identifier. This **must** be set.

Return type `IAM.UserPolicy`

Returns A UserPolicy resource

SigningCertificate (*id*)

Creates a SigningCertificate resource.:

```
signing_certificate = user.SigningCertificate('id')
```

Parameters **id** (*string*) – The SigningCertificate’s id identifier. This **must** be set.

Return type `IAM.SigningCertificate`

Returns A SigningCertificate resource

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

access_keys

all()

Creates an iterable of all AccessKey resources in the collection.

Request Syntax

```
access_key_iterator = user.access_keys.all()
```

Return type `list(iam.AccessKey)`

Returns A list of AccessKey resources

filter (***kwargs*)

Creates an iterable of all AccessKey resources in the collection filtered by kwargs passed to method.

Request Syntax


```
access_key_iterator = user.access_keys.filter(
    Marker='string',
    MaxItems=123
)
```

Parameters

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.AccessKey)`

Returns A list of `AccessKey` resources

limit (***kwargs*)

Creates an iterable up to a specified amount of `AccessKey` resources in the collection.

Request Syntax

```
access_key_iterator = user.access_keys.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(iam.AccessKey)`

Returns A list of `AccessKey` resources

page_size (***kwargs*)

Creates an iterable of all `AccessKey` resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
access_key_iterator = user.access_keys.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(iam.AccessKey)`

Returns A list of `AccessKey` resources

attached_policies

all ()

Creates an iterable of all `Policy` resources in the collection.

Request Syntax

```
policy_iterator = user.attached_policies.all()
```

Return type `list(iam.Policy)`

Returns A list of Policy resources

filter (***kwargs*)

Creates an iterable of all Policy resources in the collection filtered by kwargs passed to method.

Request Syntax

```
policy_iterator = user.attached_policies.filter(  
    PathPrefix='string',  
    Marker='string',  
    MaxItems=123  
)
```

Parameters

- **PathPrefix** (*string*) – The path prefix for filtering the results. This parameter is optional. If it is not included, it defaults to a slash (/), listing all policies.
- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.Policy)`

Returns A list of Policy resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Policy resources in the collection.

Request Syntax

```
policy_iterator = user.attached_policies.limit(  
    count=123  
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(iam.Policy)`

Returns A list of Policy resources

page_size (***kwargs*)

Creates an iterable of all Policy resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
policy_iterator = user.attached_policies.page_size(
    count=123
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(iam.Policy)`

Returns A list of Policy resources

groups

all ()

Creates an iterable of all Group resources in the collection.

Request Syntax

```
group_iterator = user.groups.all()
```

Return type `list(iam.Group)`

Returns A list of Group resources

filter (**kwargs)

Creates an iterable of all Group resources in the collection filtered by kwargs passed to method.

Request Syntax

```
group_iterator = user.groups.filter(
    Marker='string',
    MaxItems=123
)
```

Parameters

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.Group)`

Returns A list of Group resources

limit (**kwargs)

Creates an iterable up to a specified amount of Group resources in the collection.

Request Syntax

```
group_iterator = user.groups.limit(
    count=123
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(iam.Group)`

Returns A list of Group resources

page_size (***kwargs*)

Creates an iterable of all Group resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
group_iterator = user.groups.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(iam.Group)`

Returns A list of Group resources

mfa_devices

all ()

Creates an iterable of all MfaDevice resources in the collection.

Request Syntax

```
mfa_device_iterator = user.mfa_devices.all()
```

Return type `list(iam.MfaDevice)`

Returns A list of MfaDevice resources

filter (***kwargs*)

Creates an iterable of all MfaDevice resources in the collection filtered by kwargs passed to method.

Request Syntax

```
mfa_device_iterator = user.mfa_devices.filter(  
    Marker='string',  
    MaxItems=123  
)
```

Parameters

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.MfaDevice)`

Returns A list of MfaDevice resources

limit (***kwargs*)

Creates an iterable up to a specified amount of MfaDevice resources in the collection.

Request Syntax

```
mfa_device_iterator = user.mfa_devices.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type list(iam.MfaDevice)

Returns A list of MfaDevice resources

page_size (***kwargs*)

Creates an iterable of all MfaDevice resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
mfa_device_iterator = user.mfa_devices.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type list(iam.MfaDevice)

Returns A list of MfaDevice resources

policies

all ()

Creates an iterable of all UserPolicy resources in the collection.

Request Syntax

```
user_policy_iterator = user.policies.all()
```

Return type list(iam.UserPolicy)

Returns A list of UserPolicy resources

filter (***kwargs*)

Creates an iterable of all UserPolicy resources in the collection filtered by kwargs passed to method.

Request Syntax

```
user_policy_iterator = user.policies.filter(
    Marker='string',
    MaxItems=123
)
```

Parameters

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.UserPolicy)`

Returns A list of `UserPolicy` resources

limit (***kwargs*)

Creates an iterable up to a specified amount of `UserPolicy` resources in the collection.

Request Syntax

```
user_policy_iterator = user.policies.limit(  
    count=123  
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(iam.UserPolicy)`

Returns A list of `UserPolicy` resources

page_size (***kwargs*)

Creates an iterable of all `UserPolicy` resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
user_policy_iterator = user.policies.page_size(  
    count=123  
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(iam.UserPolicy)`

Returns A list of `UserPolicy` resources

signing_certificates

all ()

Creates an iterable of all `SigningCertificate` resources in the collection.

Request Syntax

```
signing_certificate_iterator = user.signing_certificates.all()
```

Return type `list(iam.SigningCertificate)`

Returns A list of `SigningCertificate` resources

filter (***kwargs*)

Creates an iterable of all `SigningCertificate` resources in the collection filtered by `kwargs` passed to method.

Request Syntax

```
signing_certificate_iterator = user.signing_certificates.filter(  
    Marker='string',  
    MaxItems=123  
)
```

Parameters

- **Marker** (*string*) – Use this parameter only when paginating results and only after you receive a response indicating that the results are truncated. Set it to the value of the `Marker` element in the response you received to inform the next call about where to start.
- **MaxItems** (*integer*) – Use this only when paginating results to indicate the maximum number of items you want in the response. If there are additional items beyond the maximum you specify, the `IsTruncated` response element is `true`.

This parameter is optional. If you do not include it, it defaults to 100. Note that IAM might return fewer results, even when there are more results available. If this is the case, the `IsTruncated` response element returns `true` and `Marker` contains a value to include in the subsequent call that tells the service where to continue from.

Return type `list(iam.SigningCertificate)`

Returns A list of `SigningCertificate` resources

limit (***kwargs*)

Creates an iterable up to a specified amount of `SigningCertificate` resources in the collection.

Request Syntax

```
signing_certificate_iterator = user.signing_certificates.limit(
    count=123
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(iam.SigningCertificate)`

Returns A list of `SigningCertificate` resources

page_size (***kwargs*)

Creates an iterable of all `SigningCertificate` resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
signing_certificate_iterator = user.signing_certificates.page_size(
    count=123
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(iam.SigningCertificate)`

Returns A list of `SigningCertificate` resources

UserPolicy

class `IAM.UserPolicy` (*user_name*, *name*)

A resource representing an AWS Identity and Access Management (IAM) `UserPolicy`:

```
import boto3
```

```
iam = boto3.resource('iam')
user_policy = iam.UserPolicy('user_name', 'name')
```

Parameters

- **user_name** (*string*) – The `UserPolicy`'s `user_name` identifier. This **must** be set.

- **name** (*string*) – The UserPolicy’s name identifier. This **must** be set.

These are the resource’s available identifiers:

- `user_name`
- `name`

These are the resource’s available attributes:

- `policy_document`
- `policy_name`

These are the resource’s available actions:

- `delete()`
- `load()`
- `put()`
- `reload()`

These are the resource’s available sub-resources:

- `User()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

user_name

(*string*) The UserPolicy’s user_name identifier. This **must** be set.

name

(*string*) The UserPolicy’s name identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

policy_document

(*string*)

The policy document.

policy_name

(*string*)

The name of the policy.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

delete()

Deletes the specified inline policy that is embedded in the specified user.

A user can also have managed policies attached to it. To detach a managed policy from a user, use `DetachUserPolicy`. For more information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Request Syntax

```
response = user_policy.delete()
```

Returns None

load()

Calls `iam.Client.get_user_policy()` to update the attributes of the UserPolicy resource

Request Syntax

```
user_policy.load()
```

Returns None**put** (***kwargs*)

Adds (or updates) an inline policy document that is embedded in the specified user.

A user can also have a managed policy attached to it. To attach a managed policy to a user, use `AttachUserPolicy`. To create a new managed policy, use `CreatePolicy`. For information about policies, refer to [Managed Policies and Inline Policies](#) in the *IAM User Guide*.

For information about limits on the number of inline policies that you can embed in a user, see [Limitations on IAM Entities](#) in the *IAM User Guide*.

Note: Because policy documents can be large, you should use POST rather than GET when calling `PutUserPolicy`. For general information about using the Query API with IAM, go to [Making Query Requests](#) in the *Using IAM* guide.

Request Syntax

```
response = user_policy.put(
    PolicyDocument='string'
)
```

Parameters `PolicyDocument` (*string*) – [REQUIRED]

The policy document.

Returns None**reload**()

Calls `iam.Client.get_user_policy()` to update the attributes of the `UserPolicy` resource

Request Syntax

```
user_policy.reload()
```

Returns None**Sub-resources**

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

User()

Creates a User resource.:

```
user = user_policy.User()
```

Return type `IAM.User`**Returns** A User resource**VirtualMfaDevice**

```
class IAM.VirtualMfaDevice(serial_number)
```

A resource representing an AWS Identity and Access Management (IAM) VirtualMfaDevice:

```
import boto3
```

```
iam = boto3.resource('iam')
virtual_mfa_device = iam.VirtualMfaDevice('serial_number')
```

Parameters `serial_number` (*string*) – The VirtualMfaDevice’s `serial_number` identifier. This **must** be set.

These are the resource’s available identifiers:

- `serial_number`

These are the resource’s available attributes:

- `base_32_string_seed`
- `enable_date`
- `qr_code_png`
- `user_attribute`

These are the resource’s available references:

- `user`

These are the resource’s available actions:

- `delete()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

serial_number

(*string*) The VirtualMfaDevice’s `serial_number` identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

base_32_string_seed

(*bytes*)

The Base32 seed defined as specified in [RFC3548](#). The Base32StringSeed is Base64-encoded.

enable_date

(*datetime*)

The date and time on which the virtual MFA device was enabled.

qr_code_png

(*bytes*)

A QR code PNG image that encodes `otpauth://totp/$virtualMFADeviceName@$AccountName?secret=$` where `$virtualMFADeviceName` is one of the `create` call arguments, `AccountName` is the user name if set (otherwise, the account ID otherwise), and `Base32String` is the seed in Base32 format. The Base32String value is Base64-encoded.

user_attribute

(*dict*)

Contains information about an IAM user entity.

This data type is used as a response element in the following actions:

- `CreateUser`
- `GetUser`
- `ListUsers`

References

References are related resource instances that have a belongs-to relationship. For more information about references refer to the [Resources Introduction Guide](#).

user

(User) The related user if set, otherwise None.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

delete()

Deletes a virtual MFA device.

Note: You must deactivate a user's virtual MFA device before you can delete it. For information about deactivating MFA devices, see `DeactivateMFADevice`.

Request Syntax

```
response = virtual_mfa_device.delete()
```

Returns None

ImportExport

Table of Contents

- [ImportExport](#)
 - [Client](#)
 - [Paginators](#)

Client

class ImportExport.Client

A low-level client representing AWS Import/Export:

```
import boto3
```

```
client = boto3.client('importexport')
```

These are the available methods:

- `can_paginate()`
- `cancel_job()`
- `create_job()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_shipping_label()`
- `get_status()`
- `get_waiter()`
- `list_jobs()`
- `update_job()`

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns `True` if the operation can be paginated, `False` otherwise.

cancel_job (***kwargs*)

This operation cancels a specified job. Only the job owner can cancel it. The operation fails if the job has already started or is complete.

Request Syntax

```
response = client.cancel_job(
    JobId='string',
    APIVersion='string'
)
```

Parameters

- **JobId** (*string*) – **[REQUIRED]** A unique identifier which refers to a particular job.
- **APIVersion** (*string*) – Specifies the version of the client tool.

Return type `dict`

Returns**Response Syntax**

```
{
    'Success': True|False
}
```

Response Structure

- (*dict*) – Output structure for the `CancelJob` operation.
 - **Success** (*boolean*) – Specifies whether (true) or not (false) AWS Import/Export updated your job.

create_job (***kwargs*)

This operation initiates the process of scheduling an upload or download of your data. You include in the request a manifest that describes the data transfer specifics. The response to the request includes a job ID, which you can use in other operations, a signature that you use to identify your storage device, and the address where you should ship your storage device.

Request Syntax

```
response = client.create_job(
    JobType='Import'|'Export',
    Manifest='string',
    ManifestAddendum='string',
    ValidateOnly=True|False,
    APIVersion='string'
)
```

Parameters

- **JobType** (*string*) – **[REQUIRED]** Specifies whether the job to initiate is an import or export job.
- **Manifest** (*string*) – **[REQUIRED]** The UTF-8 encoded text of the manifest file.

- **ManifestAddendum** (*string*) – For internal use only.
- **ValidateOnly** (*boolean*) – **[REQUIRED]** Validate the manifest and parameter values in the request but do not actually create a job.
- **APIVersion** (*string*) – Specifies the version of the client tool.

Return type dict

Returns

Response Syntax

```
{
  'JobId': 'string',
  'JobType': 'Import'|'Export',
  'Signature': 'string',
  'SignatureFileContents': 'string',
  'WarningMessage': 'string',
  'ArtifactList': [
    {
      'Description': 'string',
      'URL': 'string'
    },
  ]
}
```

Response Structure

- (*dict*) – Output structure for the CreateJob operation.
 - **JobId** (*string*) – A unique identifier which refers to a particular job.
 - **JobType** (*string*) – Specifies whether the job to initiate is an import or export job.
 - **Signature** (*string*) – An encrypted code used to authenticate the request and response, for example, “DV+TpDfx1/TdSE9ktyK9k/bDTVI=”. Only use this value is you want to create the signature file yourself. Generally you should use the SignatureFileContents value.
 - **SignatureFileContents** (*string*) – The actual text of the SIGNATURE file to be written to disk.
 - **WarningMessage** (*string*) – An optional message notifying you of non-fatal issues with the job, such as use of an incompatible Amazon S3 bucket name.
 - **ArtifactList** (*list*) – A collection of artifacts.
 - * (*dict*) – A discrete item that contains the description and URL of an artifact (such as a PDF).
 - **Description** (*string*) – The associated description for this object.
 - **URL** (*string*) – The URL for a given Artifact.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to *ClientMethod*.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method’s model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters `operation_name` (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises `OperationNotPageableError` Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_shipping_label (***kwargs*)

This operation generates a pre-paid UPS shipping label that you will use to ship your device to AWS for processing.

Request Syntax

```
response = client.get_shipping_label(
    jobIds=[
        'string',
    ],
    name='string',
    company='string',
    phoneNumber='string',
    country='string',
    stateOrProvince='string',
    city='string',
    postalCode='string',
    street1='string',
    street2='string',
    street3='string',
    APIVersion='string'
)
```

Parameters

- **jobIds** (*list*) – **[REQUIRED]**
 - (*string*) –
- **name** (*string*) – Specifies the name of the person responsible for shipping this package.
- **company** (*string*) – Specifies the name of the company that will ship this package.
- **phoneNumber** (*string*) – Specifies the phone number of the person responsible for shipping this package.
- **country** (*string*) – Specifies the name of your country for the return address.
- **stateOrProvince** (*string*) – Specifies the name of your state or your province for the return address.
- **city** (*string*) – Specifies the name of your city for the return address.
- **postalCode** (*string*) – Specifies the postal code for the return address.
- **street1** (*string*) – Specifies the first part of the street address for the return address, for example 1234 Main Street.
- **street2** (*string*) – Specifies the optional second part of the street address for the return address, for example Suite 100.
- **street3** (*string*) – Specifies the optional third part of the street address for the return address, for example c/o Jane Doe.
- **APIVersion** (*string*) – Specifies the version of the client tool.

Return type `dict`

Returns

Response Syntax

```
{
    'ShippingLabelURL': 'string',
    'Warning': 'string'
}
```

Response Structure

- (*dict*) –
 - **ShippingLabelURL** (*string*) –
 - **Warning** (*string*) –

get_status (***kwargs*)

This operation returns information about a job, including where the job is in the processing pipeline, the status of the results, and the signature value associated with the job. You can only return information about jobs you own.

Request Syntax

```
response = client.get_status(
    JobId='string',
    APIVersion='string'
)
```

Parameters

- **JobId** (*string*) – **[REQUIRED]** A unique identifier which refers to a particular job.
- **APIVersion** (*string*) – Specifies the version of the client tool.

Return type dict

Returns**Response Syntax**

```
{
    'JobId': 'string',
    'JobType': 'Import'|'Export',
    'LocationCode': 'string',
    'LocationMessage': 'string',
    'ProgressCode': 'string',
    'ProgressMessage': 'string',
    'Carrier': 'string',
    'TrackingNumber': 'string',
    'LogBucket': 'string',
    'LogKey': 'string',
    'ErrorCount': 123,
    'Signature': 'string',
    'SignatureFileContents': 'string',
    'CurrentManifest': 'string',
    'CreationDate': datetime(2015, 1, 1),
    'ArtifactList': [
        {
            'Description': 'string',
            'URL': 'string'
        },
    ],
}
```

Response Structure

- (*dict*) – Output structure for the GetStatus operation.
 - **JobId** (*string*) – A unique identifier which refers to a particular job.
 - **JobType** (*string*) – Specifies whether the job to initiate is an import or export job.
 - **LocationCode** (*string*) – A token representing the location of the storage device, such as “AtAWS”.
 - **LocationMessage** (*string*) – A more human readable form of the physical location of the storage device.
 - **ProgressCode** (*string*) – A token representing the state of the job, such as “Started”.
 - **ProgressMessage** (*string*) – A more human readable form of the job status.
 - **Carrier** (*string*) – Name of the shipping company. This value is included when the LocationCode is “Returned”.
 - **TrackingNumber** (*string*) – The shipping tracking number assigned by AWS Import/Export to the storage device when it’s returned to you. We return this value when the LocationCode is “Returned”.
 - **LogBucket** (*string*) – Amazon S3 bucket for user logs.
 - **LogKey** (*string*) – The key where the user logs were stored.
 - **ErrorCount** (*integer*) – Number of errors. We return this value when the ProgressCode is Success or SuccessWithErrors.
 - **Signature** (*string*) – An encrypted code used to authenticate the request and response, for example, “DV+TpDfx1/TdSE9ktyK9k/bDTVl=”. Only use this value if you want to create the signature file yourself. Generally you should use the SignatureFileContents value.
 - **SignatureFileContents** (*string*) – An encrypted code used to authenticate the request and response, for example, “DV+TpDfx1/TdSE9ktyK9k/bDTVl=”. Only use this value if you want to create the signature file yourself. Generally you should use the SignatureFileContents value.
 - **CurrentManifest** (*string*) – The last manifest submitted, which will be used to process the job.
 - **CreationDate** (*datetime*) – Timestamp of the CreateJob request in ISO8601 date format. For example “2010-03-28T20:27:35Z”.
 - **ArtifactList** (*list*) – A collection of artifacts.
 - * (*dict*) – A discrete item that contains the description and URL of an artifact (such as a PDF).
 - **Description** (*string*) – The associated description for this object.
 - **URL** (*string*) – The URL for a given Artifact.

`get_waiter` (*waiter_name*)

`list_jobs` (***kwargs*)

This operation returns the jobs associated with the requester. AWS Import/Export lists the jobs in reverse chronological order based on the date of creation. For example if Job Test1 was created 2009Dec30 and Test2 was created 2010Feb05, the ListJobs operation would return Test2 followed by Test1.

Request Syntax

```
response = client.list_jobs(
    MaxJobs=123,
    Marker='string',
    APIVersion='string'
)
```


Parameters

- **MaxJobs** (*integer*) – Sets the maximum number of jobs returned in the response. If there are additional jobs that were not returned because MaxJobs was exceeded, the response contains IsTruncatedtrue/IsTruncated. To return the additional jobs, see Marker.
- **Marker** (*string*) – Specifies the JOBID to start after when listing the jobs created with your account. AWS Import/Export lists your jobs in reverse chronological order. See MaxJobs.
- **APIVersion** (*string*) – Specifies the version of the client tool.

Return type dict**Returns****Response Syntax**

```
{
  'Jobs': [
    {
      'JobId': 'string',
      'CreationDate': datetime(2015, 1, 1),
      'IsCanceled': True|False,
      'JobType': 'Import'|'Export'
    },
  ],
  'IsTruncated': True|False
}
```

Response Structure

- (*dict*) – Output structure for the ListJobs operation.
 - **Jobs** (*list*) – A list container for Jobs returned by the ListJobs operation.
 - * (*dict*) – Representation of a job returned by the ListJobs operation.
 - **JobId** (*string*) – A unique identifier which refers to a particular job.
 - **CreationDate** (*datetime*) – Timestamp of the CreateJob request in ISO8601 date format. For example “2010-03-28T20:27:35Z”.
 - **IsCanceled** (*boolean*) – Indicates whether the job was canceled.
 - **JobType** (*string*) – Specifies whether the job to initiate is an import or export job.
 - **IsTruncated** (*boolean*) – Indicates whether the list of jobs was truncated. If true, then call ListJobs again using the last JobId element as the marker.

update_job (***kwargs*)

You use this operation to change the parameters specified in the original manifest file by supplying a new manifest file. The manifest file attached to this request replaces the original manifest file. You can only use the operation after a CreateJob request but before the data transfer starts and you can only use it on jobs you own.

Request Syntax

```
response = client.update_job(
    JobId='string',
    Manifest='string',
    JobType='Import'|'Export',
    ValidateOnly=True|False,
    APIVersion='string'
)
```

Parameters

- **JobId** (*string*) – **[REQUIRED]** A unique identifier which refers to a particular job.
- **Manifest** (*string*) – **[REQUIRED]** The UTF-8 encoded text of the manifest file.
- **JobType** (*string*) – **[REQUIRED]** Specifies whether the job to initiate is an import or export job.
- **ValidateOnly** (*boolean*) – **[REQUIRED]** Validate the manifest and parameter values in the request but do not actually create a job.
- **APIVersion** (*string*) – Specifies the version of the client tool.

Return type dict**Returns****Response Syntax**

```
{
  'Success': True|False,
  'WarningMessage': 'string',
  'ArtifactList': [
    {
      'Description': 'string',
      'URL': 'string'
    },
  ],
}
```

Response Structure

- (*dict*) – Output structure for the UpateJob operation.
 - **Success** (*boolean*) – Specifies whether (true) or not (false) AWS Import/Export updated your job.
 - **WarningMessage** (*string*) – An optional message notifying you of non-fatal issues with the job, such as use of an incompatible Amazon S3 bucket name.
 - **ArtifactList** (*list*) – A collection of artifacts.
 - * (*dict*) – A discrete item that contains the description and URL of an artifact (such as a PDF).
 - **Description** (*string*) – The associated description for this object.
 - **URL** (*string*) – The URL for a given Artifact.

Paginators

The available paginators are:

- `ImportExport.Paginator.ListJobs`

class `ImportExport.Paginator.ListJobs`

```
paginator = client.get_paginator('list_jobs')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `ImportExport.Client.list_jobs()`.

Request Syntax

```

response_iterator = paginator.paginate(
    APIVersion='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)

```

Parameters

- **APIVersion** (*string*) – Specifies the version of the client tool.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```

{
    'Jobs': [
        {
            'JobId': 'string',
            'CreationDate': datetime(2015, 1, 1),
            'IsCanceled': True|False,
            'JobType': 'Import'|'Export'
        },
    ],
    'IsTruncated': True|False,
    'NextToken': 'string'
}

```

Response Structure

- (*dict*) – Output structure for the `ListJobs` operation.
 - **Jobs** (*list*) – A list container for Jobs returned by the `ListJobs` operation.
 - * (*dict*) – Representation of a job returned by the `ListJobs` operation.
 - **JobId** (*string*) – A unique identifier which refers to a particular job.
 - **CreationDate** (*datetime*) – Timestamp of the `CreateJob` request in ISO8601 date format. For example “2010-03-28T20:27:35Z”.
 - **IsCanceled** (*boolean*) – Indicates whether the job was canceled.
 - **JobType** (*string*) – Specifies whether the job to initiate is an import or export job.

- **IsTruncated** (*boolean*) – Indicates whether the list of jobs was truncated. If true, then call ListJobs again using the last JobId element as the marker.
- **NextToken** (*string*) –
A token to resume pagination.

Kinesis

Table of Contents

- Kinesis
 - Client
 - Paginators
 - Waiters

Client

`class Kinesis.Client`

A low-level client representing Amazon Kinesis:

```
import boto3

client = boto3.client('kinesis')
```

These are the available methods:

- `add_tags_to_stream()`
- `can_paginate()`
- `create_stream()`
- `delete_stream()`
- `describe_stream()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_records()`
- `get_shard_iterator()`
- `get_waiter()`
- `list_streams()`
- `list_tags_for_stream()`
- `merge_shards()`
- `put_record()`
- `put_records()`
- `remove_tags_from_stream()`
- `split_shard()`

`add_tags_to_stream(**kwargs)`

Adds or updates tags for the specified Amazon Kinesis stream. Each stream can have up to 10 tags.

If tags have already been assigned to the stream, `AddTagsToStream` overwrites any existing tags that correspond to the specified tag keys.

Request Syntax

```
response = client.add_tags_to_stream(
    StreamName='string',
    Tags={
```

```

        'string': 'string'
    }
)

```

Parameters

- **StreamName** (*string*) – [REQUIRED]
The name of the stream.
- **Tags** (*dict*) – [REQUIRED]
The set of key-value pairs to use to create the tags.
– (*string*) –
 * (*string*) –

Returns None

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

create_stream (***kwargs*)

Creates a Amazon Kinesis stream. A stream captures and transports data records that are continuously emitted from different data sources or *producers*. Scale-out within an Amazon Kinesis stream is explicitly supported by means of shards, which are uniquely identified groups of data records in an Amazon Kinesis stream.

You specify and control the number of shards that a stream is composed of. Each shard can support reads up to 5 transactions per second, up to a maximum data read total of 2 MB per second. Each shard can support writes up to 1,000 records per second, up to a maximum data write total of 1 MB per second. You can add shards to a stream if the amount of data input increases and you can remove shards if the amount of data input decreases.

The stream name identifies the stream. The name is scoped to the AWS account used by the application. It is also scoped by region. That is, two streams in two different accounts can have the same name, and two streams in the same account, but in two different regions, can have the same name.

CreateStream is an asynchronous operation. Upon receiving a CreateStream request, Amazon Kinesis immediately returns and sets the stream status to `CREATING`. After the stream is created, Amazon Kinesis sets the stream status to `ACTIVE`. You should perform read and write operations only on an `ACTIVE` stream.

You receive a `LimitExceededException` when making a CreateStream request if you try to do one of the following:

- Have more than five streams in the `CREATING` state at any point in time.
- Create more shards than are authorized for your account.

For the default shard limit for an AWS account, see [Amazon Kinesis Limits](#). If you need to increase this limit, [contact AWS Support](#).

You can use `DescribeStream` to check the stream status, which is returned in `StreamStatus`.

CreateStream has a limit of 5 transactions per second per account.

Request Syntax

```

response = client.create_stream(
    StreamName='string',

```

```
ShardCount=123
)
```

Parameters

- **StreamName** (*string*) – [REQUIRED]

A name to identify the stream. The stream name is scoped to the AWS account used by the application that creates the stream. It is also scoped by region. That is, two streams in two different AWS accounts can have the same name, and two streams in the same AWS account, but in two different regions, can have the same name.

- **ShardCount** (*integer*) – [REQUIRED]

The number of shards that the stream will use. The throughput of the stream is a function of the number of shards; more shards are required for greater provisioned throughput.

DefaultShardLimit;

Returns None**delete_stream** (***kwargs*)

Deletes a stream and all its shards and data. You must shut down any applications that are operating on the stream before you delete the stream. If an application attempts to operate on a deleted stream, it will receive the exception `ResourceNotFoundException`.

If the stream is in the `ACTIVE` state, you can delete it. After a `DeleteStream` request, the specified stream is in the `DELETING` state until Amazon Kinesis completes the deletion.

Note: Amazon Kinesis might continue to accept data read and write operations, such as `PutRecord`, `PutRecords`, and `GetRecords`, on a stream in the `DELETING` state until the stream deletion is complete.

When you delete a stream, any shards in that stream are also deleted, and any tags are dissociated from the stream.

You can use the `DescribeStream` operation to check the state of the stream, which is returned in `StreamStatus`.

`DeleteStream` has a limit of 5 transactions per second per account.

Request Syntax

```
response = client.delete_stream(
    StreamName='string'
)
```

Parameters **StreamName** (*string*) – [REQUIRED]

The name of the stream to delete.

Returns None**describe_stream** (***kwargs*)

Describes the specified stream.

The information about the stream includes its current status, its Amazon Resource Name (ARN), and an array of shard objects. For each shard object, there is information about the hash key and sequence number ranges that the shard spans, and the IDs of any earlier shards that played in a role in creating the shard. A sequence number is the identifier associated with every record ingested in the Amazon Kinesis stream. The sequence number is assigned when a record is put into the stream.

You can limit the number of returned shards using the `Limit` parameter. The number of shards in a stream may be too large to return from a single call to `DescribeStream`. You can detect this by using

the `HasMoreShards` flag in the returned output. `HasMoreShards` is set to `true` when there is more data available.

`DescribeStream` is a paginated operation. If there are more shards available, you can request them using the shard ID of the last shard returned. Specify this ID in the `ExclusiveStartShardId` parameter in a subsequent request to `DescribeStream`.

`DescribeStream` has a limit of 10 transactions per second per account.

Request Syntax

```
response = client.describe_stream(
    StreamName='string',
    Limit=123,
    ExclusiveStartShardId='string'
)
```

Parameters

- **StreamName** (*string*) – [REQUIRED]
The name of the stream to describe.
- **Limit** (*integer*) – The maximum number of shards to return.
- **ExclusiveStartShardId** (*string*) – The shard ID of the shard to start with.

Return type dict

Returns

Response Syntax

```
{
  'StreamDescription': {
    'StreamName': 'string',
    'StreamARN': 'string',
    'StreamStatus': 'CREATING'|'DELETING'|'ACTIVE'|'UPDATING',
    'Shards': [
      {
        'ShardId': 'string',
        'ParentShardId': 'string',
        'AdjacentParentShardId': 'string',
        'HashKeyRange': {
          'StartingHashKey': 'string',
          'EndingHashKey': 'string'
        },
        'SequenceNumberRange': {
          'StartingSequenceNumber': 'string',
          'EndingSequenceNumber': 'string'
        }
      },
    ],
    'HasMoreShards': True|False
  }
}
```

Response Structure

- (*dict*) –

Represents the output for `DescribeStream`.

- **StreamDescription** (*dict*) –

The current status of the stream, the stream ARN, an array of shard objects that comprise the stream, and states whether there are more shards available.

* **StreamName** (*string*) –

The name of the stream being described.

* **StreamARN** (*string*) –

The Amazon Resource Name (ARN) for the stream being described.

* **StreamStatus** (*string*) –

The current status of the stream being described.

The stream status is one of the following states:

- **CREATING** - The stream is being created. Amazon Kinesis immediately returns and sets `StreamStatus` to `CREATING`.
- **DELETING** - The stream is being deleted. The specified stream is in the `DELETING` state until Amazon Kinesis completes the deletion.
- **ACTIVE** - The stream exists and is ready for read and write operations or deletion. You should perform read and write operations only on an `ACTIVE` stream.
- **UPDATING** - Shards in the stream are being merged or split. Read and write operations continue to work while the stream is in the `UPDATING` state.

* **Shards** (*list*) –

The shards that comprise the stream.

· (*dict*) –

A uniquely identified group of data records in an Amazon Kinesis stream.

· **ShardId** (*string*) –

The unique identifier of the shard within the Amazon Kinesis stream.

· **ParentShardId** (*string*) –

The shard Id of the shard's parent.

· **AdjacentParentShardId** (*string*) –

The shard Id of the shard adjacent to the shard's parent.

· **HashKeyRange** (*dict*) –

The range of possible hash key values for the shard, which is a set of ordered contiguous positive integers.

· **StartingHashKey** (*string*) –

The starting hash key of the hash key range.

· **EndingHashKey** (*string*) –

The ending hash key of the hash key range.

· **SequenceNumberRange** (*dict*) –

The range of possible sequence numbers for the shard.

· **StartingSequenceNumber** (*string*) –

The starting sequence number for the range.

· **EndingSequenceNumber** (*string*) –

The ending sequence number for the range. Shards that are in the `OPEN` state have an ending sequence number of `null`.

* **HasMoreShards** (*boolean*) –

If set to `true`, more shards in the stream are available to describe.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_records (***kwargs*)

Gets data records from a shard.

Specify a shard iterator using the `ShardIterator` parameter. The shard iterator specifies the position in the shard from which you want to start reading data records sequentially. If there are no records available in the portion of the shard that the iterator points to, `GetRecords` returns an empty list. Note that it might take multiple calls to `get` to a portion of the shard that contains records.

You can scale by provisioning multiple shards. Your application should have one thread per shard, each reading continuously from its stream. To read from a stream continually, call `GetRecords` in a loop. Use `GetShardIterator` to get the shard iterator to specify in the first `GetRecords` call. `GetRecords` returns a new shard iterator in `NextShardIterator`. Specify the shard iterator returned in `NextShardIterator` in subsequent calls to `GetRecords`. Note that if the shard has been closed, the shard iterator can't return more data and `GetRecords` returns `null` in `NextShardIterator`. You can terminate the loop when the shard is closed, or when the shard iterator reaches the record with the sequence number or other attribute that marks it as the last record to process.

Each data record can be up to 1 MB in size, and each shard can read up to 2 MB per second. You can ensure that your calls don't exceed the maximum supported size or throughput by using the `Limit` parameter to specify the maximum number of records that `GetRecords` can return. Consider your average record size when determining this limit.

The size of the data returned by `GetRecords` will vary depending on the utilization of the shard. The maximum size of data that `GetRecords` can return is 10 MB. If a call returns this amount of data, subsequent calls made within the next 5 seconds throw `ProvisionedThroughputExceededException`. If there is insufficient provisioned throughput on the shard, subsequent calls made within the next 1 second throw `ProvisionedThroughputExceededException`. Note that `GetRecords` won't return any data when it throws an exception. For this reason, we recommend that you wait one second between calls to `GetRecords`; however, it's possible that the application will get exceptions for longer than 1 second.

To detect whether the application is falling behind in processing, you can use the

`MillisBehindLatest` response attribute. You can also monitor the stream using CloudWatch metrics (see [Monitoring Amazon Kinesis](#) in the *Amazon Kinesis Developer Guide*).

Each Amazon Kinesis record includes a value, `ApproximateArrivalTimestamp`, that is set when an Amazon Kinesis stream successfully receives and stores a record. This is commonly referred to as a server-side timestamp, which is different than a client-side timestamp, where the timestamp is set when a data producer creates or sends the record to a stream. The timestamp has millisecond precision. There are no guarantees about the timestamp accuracy, or that the timestamp is always increasing. For example, records in a shard or across a stream might have timestamps that are out of order.

Request Syntax

```
response = client.get_records(
    ShardIterator='string',
    Limit=123
)
```

Parameters

- **ShardIterator** (*string*) – **[REQUIRED]**

The position in the shard from which you want to start sequentially reading data records. A shard iterator specifies this position using the sequence number of a data record in the shard.

- **Limit** (*integer*) – The maximum number of records to return. Specify a value of up to 10,000. If you specify a value that is greater than 10,000, `GetRecords` throws `InvalidArgumentException`.

Return type dict

Returns

Response Syntax

```
{
    'Records': [
        {
            'SequenceNumber': 'string',
            'ApproximateArrivalTimestamp': datetime(2015, 1, 1),
            'Data': b'bytes',
            'PartitionKey': 'string'
        },
    ],
    'NextShardIterator': 'string',
    'MillisBehindLatest': 123
}
```

Response Structure

- (*dict*) –

Represents the output for `GetRecords`.

- **Records** (*list*) –

The data records retrieved from the shard.

- * (*dict*) –

The unit of data of the Amazon Kinesis stream, which is composed of a sequence number, a partition key, and a data blob.

- **SequenceNumber** (*string*) –

The unique identifier of the record in the stream.

- **ApproximateArrivalTimestamp** (*datetime*) –

The approximate time that the record was inserted into the stream.

- **Data** (*bytes*) –

The data blob. The data in the blob is both opaque and immutable to the Amazon Kinesis service, which does not inspect, interpret, or change the data in the blob in any way. When the data blob (the payload before base64-encoding) is added to the partition key size, the total size must not exceed the maximum record size (1 MB).

- **PartitionKey** (*string*) –

Identifies which shard in the stream the data record is assigned to.

- **NextShardIterator** (*string*) –

The next position in the shard from which to start sequentially reading data records. If set to `null`, the shard has been closed and the requested iterator will not return any more data.

- **MillisBehindLatest** (*integer*) –

The number of milliseconds the GetRecords response is from the tip of the stream, indicating how far behind current time the consumer is. A value of zero indicates record processing is caught up, and there are no new records to process at this moment.

get_shard_iterator (***kwargs*)

Gets a shard iterator. A shard iterator expires five minutes after it is returned to the requester.

A shard iterator specifies the position in the shard from which to start reading data records sequentially. A shard iterator specifies this position using the sequence number of a data record in a shard. A sequence number is the identifier associated with every record ingested in the Amazon Kinesis stream. The sequence number is assigned when a record is put into the stream.

You must specify the shard iterator type. For example, you can set the `ShardIteratorType` parameter to read exactly from the position denoted by a specific sequence number by using the `AT_SEQUENCE_NUMBER` shard iterator type, or right after the sequence number by using the `AFTER_SEQUENCE_NUMBER` shard iterator type, using sequence numbers returned by earlier calls to `PutRecord`, `PutRecords`, `GetRecords`, or `DescribeStream`. You can specify the shard iterator type `TRIM_HORIZON` in the request to cause `ShardIterator` to point to the last untrimmed record in the shard in the system, which is the oldest data record in the shard. Or you can point to just after the most recent record in the shard, by using the shard iterator type `LATEST`, so that you always read the most recent data in the shard.

When you repeatedly read from an Amazon Kinesis stream use a `GetShardIterator` request to get the first shard iterator for use in your first `GetRecords` request and then use the shard iterator returned by the `GetRecords` request in `NextShardIterator` for subsequent reads. A new shard iterator is returned by every `GetRecords` request in `NextShardIterator`, which you use in the `ShardIterator` parameter of the next `GetRecords` request.

If a `GetShardIterator` request is made too often, you receive a `ProvisionedThroughputExceededException`. For more information about throughput limits, see `GetRecords`.

If the shard is closed, the iterator can't return more data, and `GetShardIterator` returns `null` for its `ShardIterator`. A shard can be closed using `SplitShard` or `MergeShards`.

`GetShardIterator` has a limit of 5 transactions per second per account per open shard.

Request Syntax

```
response = client.get_shard_iterator(
    StreamName='string',
    ShardId='string',
    ShardIteratorType='AT_SEQUENCE_NUMBER' | 'AFTER_SEQUENCE_NUMBER' | 'TRIM_HORIZON' | 'LATEST',
    StartingSequenceNumber='string'
)
```

Parameters

- **StreamName** (*string*) – [REQUIRED]

The name of the stream.

- **ShardId** (*string*) – [REQUIRED]

The shard ID of the shard to get the iterator for.

- **ShardIteratorType** (*string*) – [REQUIRED]

Determines how the shard iterator is used to start reading data records from the shard.

The following are the valid shard iterator types:

- AT_SEQUENCE_NUMBER - Start reading exactly from the position denoted by a specific sequence number.
- AFTER_SEQUENCE_NUMBER - Start reading right after the position denoted by a specific sequence number.
- TRIM_HORIZON - Start reading at the last untrimmed record in the shard in the system, which is the oldest data record in the shard.
- LATEST - Start reading just after the most recent record in the shard, so that you always read the most recent data in the shard.

- **StartingSequenceNumber** (*string*) – The sequence number of the data record in the shard from which to start reading from.

Return type dict

Returns

Response Syntax

```
{
    'ShardIterator': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output for `GetShardIterator`.

- **ShardIterator** (*string*) –

The position in the shard from which to start reading data records sequentially. A shard iterator specifies this position using the sequence number of a data record in a shard.

get_waiter (*waiter_name*)

list_streams (***kwargs*)

Lists your streams.

The number of streams may be too large to return from a single call to `ListStreams`. You can limit the number of returned streams using the `Limit` parameter. If you do not specify a value for the `Limit` parameter, Amazon Kinesis uses the default limit, which is currently 10.

You can detect if there are more streams available to list by using the `HasMoreStreams` flag from the returned output. If there are more streams available, you can request more streams by using the name of the last stream returned by the `ListStreams` request in the `ExclusiveStartStreamName` parameter in a subsequent request to `ListStreams`. The group of stream names returned by the subsequent request is then added to the list. You can continue this process until all the stream names have been collected in the list.

`ListStreams` has a limit of 5 transactions per second per account.

Request Syntax

```
response = client.list_streams(
    Limit=123,
    ExclusiveStartStreamName='string'
)
```

Parameters

- **Limit** (*integer*) – The maximum number of streams to list.
- **ExclusiveStartStreamName** (*string*) – The name of the stream to start the list with.

Return type dict

Returns

Response Syntax

```
{
    'StreamNames': [
        'string',
    ],
    'HasMoreStreams': True|False
}
```

Response Structure

- (*dict*) –

Represents the output for `ListStreams`.

- **StreamNames** (*list*) –

The names of the streams that are associated with the AWS account making the `ListStreams` request.

* (*string*) –

- **HasMoreStreams** (*boolean*) –

If set to `true`, there are more streams available to list.

list_tags_for_stream (***kwargs*)

Lists the tags for the specified Amazon Kinesis stream.

Request Syntax

```
response = client.list_tags_for_stream(
    StreamName='string',
    ExclusiveStartTagKey='string',
    Limit=123
)
```

Parameters

- **StreamName** (*string*) – [REQUIRED]
The name of the stream.

- **ExclusiveStartTagKey** (*string*) – The key to use as the starting point for the list of tags. If this parameter is set, `ListTagsForStream` gets all tags that occur after `ExclusiveStartTagKey`.
- **Limit** (*integer*) – The number of tags to return. If this number is less than the total number of tags associated with the stream, `HasMoreTags` is set to `true`. To list additional tags, set `ExclusiveStartTagKey` to the last key in the response.

Return type dict

Returns

Response Syntax

```
{
  'Tags': [
    {
      'Key': 'string',
      'Value': 'string'
    },
  ],
  'HasMoreTags': True|False
}
```

Response Structure

- (*dict*) –

Represents the output for `ListTagsForStream`.

- **Tags** (*list*) –

A list of tags associated with `StreamName`, starting with the first tag after `ExclusiveStartTagKey` and up to the specified `Limit`.

- * (*dict*) –

Metadata assigned to the stream, consisting of a key-value pair.

- **Key** (*string*) –

A unique identifier for the tag. Maximum length: 128 characters. Valid characters: Unicode letters, digits, white space, `_`, `.`, `/`, `=`, `+`, `-`, `%`, `@`

- **Value** (*string*) –

An optional string, typically used to describe or define the tag. Maximum length: 256 characters. Valid characters: Unicode letters, digits, white space, `_`, `.`, `/`, `=`, `+`, `-`, `%`, `@`

- **HasMoreTags** (*boolean*) –

If set to `true`, more tags are available. To request additional tags, set `ExclusiveStartTagKey` to the key of the last tag returned.

merge_shards (***kwargs*)

Merges two adjacent shards in a stream and combines them into a single shard to reduce the stream's capacity to ingest and transport data. Two shards are considered adjacent if the union of the hash key ranges for the two shards form a contiguous set with no gaps. For example, if you have two shards, one with a hash key range of 276...381 and the other with a hash key range of 382...454, then you could merge these two shards into a single shard that would have a hash key range of 276...454. After the merge, the single child shard receives data for all hash key values covered by the two parent shards.

`MergeShards` is called when there is a need to reduce the overall capacity of a stream because of excess capacity that is not being used. You must specify the shard to be merged and the adjacent shard

for a stream. For more information about merging shards, see [Merge Two Shards](#) in the *Amazon Kinesis Developer Guide*.

If the stream is in the `ACTIVE` state, you can call `MergeShards`. If a stream is in the `CREATING`, `UPDATING`, or `DELETING` state, `MergeShards` returns a `ResourceInUseException`. If the specified stream does not exist, `MergeShards` returns a `ResourceNotFoundException`.

You can use `DescribeStream` to check the state of the stream, which is returned in `StreamStatus`.

`MergeShards` is an asynchronous operation. Upon receiving a `MergeShards` request, Amazon Kinesis immediately returns a response and sets the `StreamStatus` to `UPDATING`. After the operation is completed, Amazon Kinesis sets the `StreamStatus` to `ACTIVE`. Read and write operations continue to work while the stream is in the `UPDATING` state.

You use `DescribeStream` to determine the shard IDs that are specified in the `MergeShards` request.

If you try to operate on too many streams in parallel using `CreateStream`, `DeleteStream`, `MergeShards` or `SplitShard`, you will receive a `LimitExceededException`.

`MergeShards` has limit of 5 transactions per second per account.

Request Syntax

```
response = client.merge_shards(
    StreamName='string',
    ShardToMerge='string',
    AdjacentShardToMerge='string'
)
```

Parameters

- **StreamName** (*string*) – [REQUIRED]
The name of the stream for the merge.
- **ShardToMerge** (*string*) – [REQUIRED]
The shard ID of the shard to combine with the adjacent shard for the merge.
- **AdjacentShardToMerge** (*string*) – [REQUIRED]
The shard ID of the adjacent shard for the merge.

Returns None

`put_record` (**kwargs)

Writes a single data record from a producer into an Amazon Kinesis stream. Call `PutRecord` to send data from the producer into the Amazon Kinesis stream for real-time ingestion and subsequent processing, one record at a time. Each shard can support writes up to 1,000 records per second, up to a maximum data write total of 1 MB per second.

You must specify the name of the stream that captures, stores, and transports the data; a partition key; and the data blob itself.

The data blob can be any type of data; for example, a segment from a log file, geographic/location data, website clickstream data, and so on.

The partition key is used by Amazon Kinesis to distribute data across shards. Amazon Kinesis segregates the data records that belong to a data stream into multiple shards, using the partition key associated with each data record to determine which shard a given data record belongs to.

Partition keys are Unicode strings, with a maximum length limit of 256 characters for each key. An MD5 hash function is used to map partition keys to 128-bit integer values and to map associated data records to shards using the hash key ranges of the shards. You can override hashing the partition key to determine the shard by explicitly specifying a hash value using the `ExplicitHashKey` parameter. For more information, see [Adding Data to a Stream](#) in the *Amazon Kinesis Developer Guide*.

`PutRecord` returns the shard ID of where the data record was placed and the sequence number that was assigned to the data record.

Sequence numbers generally increase over time. To guarantee strictly increasing ordering, use the `SequenceNumberForOrdering` parameter. For more information, see [Adding Data to a Stream](#) in the *Amazon Kinesis Developer Guide*.

If a `PutRecord` request cannot be processed because of insufficient provisioned throughput on the shard involved in the request, `PutRecord` throws `ProvisionedThroughputExceededException`.

Data records are accessible for only 24 hours from the time that they are added to an Amazon Kinesis stream.

Request Syntax

```
response = client.put_record(
    StreamName='string',
    Data=b'bytes',
    PartitionKey='string',
    ExplicitHashKey='string',
    SequenceNumberForOrdering='string'
)
```

Parameters

- **StreamName** (*string*) – [REQUIRED]

The name of the stream to put the data record into.

- **Data** (*bytes*) – [REQUIRED]

The data blob to put into the record, which is base64-encoded when the blob is serialized. When the data blob (the payload before base64-encoding) is added to the partition key size, the total size must not exceed the maximum record size (1 MB).

- **PartitionKey** (*string*) – [REQUIRED]

Determines which shard in the stream the data record is assigned to. Partition keys are Unicode strings with a maximum length limit of 256 characters for each key. Amazon Kinesis uses the partition key as input to a hash function that maps the partition key and associated data to a specific shard. Specifically, an MD5 hash function is used to map partition keys to 128-bit integer values and to map associated data records to shards. As a result of this hashing mechanism, all data records with the same partition key will map to the same shard within the stream.

- **ExplicitHashKey** (*string*) – The hash value used to explicitly determine the shard the data record is assigned to by overriding the partition key hash.
- **SequenceNumberForOrdering** (*string*) – Guarantees strictly increasing sequence numbers, for puts from the same client and to the same partition key. Usage: set the `SequenceNumberForOrdering` of record *n* to the sequence number of record *n-1* (as returned in the result when putting record *n-1*). If this parameter is not set, records will be coarsely ordered based on arrival time.

Return type dict

Returns

Response Syntax

```
{
    'ShardId': 'string',
    'SequenceNumber': 'string'
}
```


Response Structure

- (*dict*) –

Represents the output for `PutRecord`.

- **ShardId** (*string*) –

The shard ID of the shard where the data record was placed.

- **SequenceNumber** (*string*) –

The sequence number identifier that was assigned to the put data record. The sequence number for the record is unique across all records in the stream. A sequence number is the identifier associated with every record put into the stream.

put_records (***kwargs*)

Writes multiple data records from a producer into an Amazon Kinesis stream in a single call (also referred to as a `PutRecords` request). Use this operation to send data from a data producer into the Amazon Kinesis stream for data ingestion and processing.

Each `PutRecords` request can support up to 500 records. Each record in the request can be as large as 1 MB, up to a limit of 5 MB for the entire request, including partition keys. Each shard can support writes up to 1,000 records per second, up to a maximum data write total of 1 MB per second.

You must specify the name of the stream that captures, stores, and transports the data; and an array of request `Records`, with each record in the array requiring a partition key and data blob. The record size limit applies to the total size of the partition key and data blob.

The data blob can be any type of data; for example, a segment from a log file, geographic/location data, website clickstream data, and so on.

The partition key is used by Amazon Kinesis as input to a hash function that maps the partition key and associated data to a specific shard. An MD5 hash function is used to map partition keys to 128-bit integer values and to map associated data records to shards. As a result of this hashing mechanism, all data records with the same partition key map to the same shard within the stream. For more information, see [Adding Data to a Stream](#) in the *Amazon Kinesis Developer Guide*.

Each record in the `Records` array may include an optional parameter, `ExplicitHashKey`, which overrides the partition key to shard mapping. This parameter allows a data producer to determine explicitly the shard where the record is stored. For more information, see [Adding Multiple Records with PutRecords](#) in the *Amazon Kinesis Developer Guide*.

The `PutRecords` response includes an array of response `Records`. Each record in the response array directly correlates with a record in the request array using natural ordering, from the top to the bottom of the request and response. The response `Records` array always includes the same number of records as the request array.

The response `Records` array includes both successfully and unsuccessfully processed records. Amazon Kinesis attempts to process all records in each `PutRecords` request. A single record failure does not stop the processing of subsequent records.

A successfully-processed record includes `ShardId` and `SequenceNumber` values. The `ShardId` parameter identifies the shard in the stream where the record is stored. The `SequenceNumber` parameter is an identifier assigned to the put record, unique to all records in the stream.

An unsuccessfully-processed record includes `ErrorCode` and `ErrorMessage` values. `ErrorCode` reflects the type of error and can be one of the following values: `ProvisionedThroughputExceededException` or `InternalFailure`. `ErrorMessage` provides more detailed information about the `ProvisionedThroughputExceededException` exception including the account ID, stream name, and shard ID of the record that was throttled. For more

information about partially successful responses, see [Adding Multiple Records with PutRecords](#) in the *Amazon Kinesis Developer Guide*.

Data records are accessible for only 24 hours from the time that they are added to an Amazon Kinesis stream.

Request Syntax

```
response = client.put_records(
    Records=[
        {
            'Data': b'bytes',
            'ExplicitHashKey': 'string',
            'PartitionKey': 'string'
        },
    ],
    StreamName='string'
)
```

Parameters

- **Records** (*list*) – [REQUIRED]

The records associated with the request.

– (*dict*) –

Represents the output for PutRecords.

- * **Data** (*bytes*) – [REQUIRED]

The data blob to put into the record, which is base64-encoded when the blob is serialized. When the data blob (the payload before base64-encoding) is added to the partition key size, the total size must not exceed the maximum record size (1 MB).

- * **ExplicitHashKey** (*string*) –

The hash value used to determine explicitly the shard that the data record is assigned to by overriding the partition key hash.

- * **PartitionKey** (*string*) – [REQUIRED]

Determines which shard in the stream the data record is assigned to. Partition keys are Unicode strings with a maximum length limit of 256 characters for each key. Amazon Kinesis uses the partition key as input to a hash function that maps the partition key and associated data to a specific shard. Specifically, an MD5 hash function is used to map partition keys to 128-bit integer values and to map associated data records to shards. As a result of this hashing mechanism, all data records with the same partition key map to the same shard within the stream.

- **StreamName** (*string*) – [REQUIRED]

The stream name associated with the request.

Return type dict

Returns

Response Syntax

```
{
    'FailedRecordCount': 123,
    'Records': [
        {
```

```

        'SequenceNumber': 'string',
        'ShardId': 'string',
        'ErrorCode': 'string',
        'ErrorMessage': 'string'
    },
]
}

```

Response Structure

- (*dict*) –

PutRecords results.

- **FailedRecordCount** (*integer*) –

The number of unsuccessfully processed records in a PutRecords request.

- **Records** (*list*) –

An array of successfully and unsuccessfully processed record results, correlated with the request by natural ordering. A record that is successfully added to your Amazon Kinesis stream includes `SequenceNumber` and `ShardId` in the result. A record that fails to be added to your Amazon Kinesis stream includes `ErrorCode` and `ErrorMessage` in the result.

- * (*dict*) –

Represents the result of an individual record from a PutRecords request. A record that is successfully added to your Amazon Kinesis stream includes `SequenceNumber` and `ShardId` in the result. A record that fails to be added to your Amazon Kinesis stream includes `ErrorCode` and `ErrorMessage` in the result.

- **SequenceNumber** (*string*) –

The sequence number for an individual record result.

- **ShardId** (*string*) –

The shard ID for an individual record result.

- **ErrorCode** (*string*) –

The error code for an individual record result. ErrorCodes can be either `ProvisionedThroughputExceededException` or `InternalFailure`.

- **ErrorMessage** (*string*) –

The error message for an individual record result. An `ErrorCode` value of `ProvisionedThroughputExceededException` has an error message that includes the account ID, stream name, and shard ID. An `ErrorCode` value of `InternalFailure` has the error message "Internal Service Failure".

remove_tags_from_stream (***kwargs*)

Deletes tags from the specified Amazon Kinesis stream.

If you specify a tag that does not exist, it is ignored.

Request Syntax

```
response = client.remove_tags_from_stream(
    StreamName='string',
    TagKeys=[
        'string',
    ]
)
```

Parameters

- **StreamName** (*string*) – [REQUIRED]

The name of the stream.

- **TagKeys** (*list*) – [REQUIRED]

A list of tag keys. Each corresponding tag is removed from the stream.

– (*string*) –

Returns None**split_shard** (***kwargs*)

Splits a shard into two new shards in the stream, to increase the stream’s capacity to ingest and transport data. `SplitShard` is called when there is a need to increase the overall capacity of stream because of an expected increase in the volume of data records being ingested.

You can also use `SplitShard` when a shard appears to be approaching its maximum utilization, for example, when the set of producers sending data into the specific shard are suddenly sending more than previously anticipated. You can also call `SplitShard` to increase stream capacity, so that more Amazon Kinesis applications can simultaneously read data from the stream for real-time processing.

You must specify the shard to be split and the new hash key, which is the position in the shard where the shard gets split in two. In many cases, the new hash key might simply be the average of the beginning and ending hash key, but it can be any hash key value in the range being mapped into the shard. For more information about splitting shards, see [Split a Shard](#) in the *Amazon Kinesis Developer Guide*.

You can use `DescribeStream` to determine the shard ID and hash key values for the `ShardToSplit` and `NewStartingHashKey` parameters that are specified in the `SplitShard` request.

`SplitShard` is an asynchronous operation. Upon receiving a `SplitShard` request, Amazon Kinesis immediately returns a response and sets the stream status to `UPDATING`. After the operation is completed, Amazon Kinesis sets the stream status to `ACTIVE`. Read and write operations continue to work while the stream is in the `UPDATING` state.

You can use `DescribeStream` to check the status of the stream, which is returned in `StreamStatus`. If the stream is in the `ACTIVE` state, you can call `SplitShard`. If a stream is in `CREATING` or `UPDATING` or `DELETING` states, `DescribeStream` returns a `ResourceInUseException`.

If the specified stream does not exist, `DescribeStream` returns a `ResourceNotFoundException`. If you try to create more shards than are authorized for your account, you receive a `LimitExceededException`.

For the default shard limit for an AWS account, see [Amazon Kinesis Limits](#). If you need to increase this limit, [contact AWS Support](#).

If you try to operate on too many streams in parallel using `CreateStream`, `DeleteStream`, `MergeShards` or `SplitShard`, you receive a `LimitExceededException`.

`SplitShard` has limit of 5 transactions per second per account.

Request Syntax

```
response = client.split_shard(
    StreamName='string',
```

```

    ShardToSplit='string',
    NewStartingHashKey='string'
)

```

Parameters

- **StreamName** (*string*) – [REQUIRED]
The name of the stream for the shard split.
- **ShardToSplit** (*string*) – [REQUIRED]
The shard ID of the shard to split.
- **NewStartingHashKey** (*string*) – [REQUIRED]
A hash key value for the starting hash key of one of the child shards created by the split. The hash key range for a given shard constitutes a set of ordered contiguous positive integers. The value for `NewStartingHashKey` must be in the range of hash keys being mapped into the shard. The `NewStartingHashKey` hash key value and all higher hash key values in hash key range are distributed to one of the child shards. All the lower hash key values in the range are distributed to the other child shard.

Returns None

Paginators

The available paginators are:

- `Kinesis.Paginator.DescribeStream`
- `Kinesis.Paginator.ListStreams`

class `Kinesis.Paginator.DescribeStream`

```
paginator = client.get_paginator('describe_stream')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Kinesis.Client.describe_stream()`.

Request Syntax

```

response_iterator = paginator.paginate(
    StreamName='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)

```

Parameters

- **StreamName** (*string*) – [REQUIRED]
The name of the stream to describe.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'StreamDescription': {
    'StreamName': 'string',
    'StreamARN': 'string',
    'StreamStatus': 'CREATING'|'DELETING'|'ACTIVE'|'UPDATING',
    'Shards': [
      {
        'ShardId': 'string',
        'ParentShardId': 'string',
        'AdjacentParentShardId': 'string',
        'HashKeyRange': {
          'StartingHashKey': 'string',
          'EndingHashKey': 'string'
        },
        'SequenceNumberRange': {
          'StartingSequenceNumber': 'string',
          'EndingSequenceNumber': 'string'
        }
      },
    ],
    'HasMoreShards': True|False
  },
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output for `DescribeStream`.

- **StreamDescription** (*dict*) –

The current status of the stream, the stream ARN, an array of shard objects that comprise the stream, and states whether there are more shards available.

- * **StreamName** (*string*) –

The name of the stream being described.

- * **StreamARN** (*string*) –

The Amazon Resource Name (ARN) for the stream being described.

- * **StreamStatus** (*string*) –

The current status of the stream being described.

The stream status is one of the following states:

- **CREATING** - The stream is being created. Amazon Kinesis immediately returns and sets `StreamStatus` to `CREATING`.
- **DELETING** - The stream is being deleted. The specified stream is in the `DELETING` state until Amazon Kinesis completes the deletion.
- **ACTIVE** - The stream exists and is ready for read and write operations or deletion. You should perform read and write operations only on an `ACTIVE` stream.
- **UPDATING** - Shards in the stream are being merged or split. Read and write operations continue to work while the stream is in the `UPDATING` state.

* **Shards** (*list*) –

The shards that comprise the stream.

- (*dict*) –

A uniquely identified group of data records in an Amazon Kinesis stream.

- **ShardId** (*string*) –

The unique identifier of the shard within the Amazon Kinesis stream.

- **ParentShardId** (*string*) –

The shard Id of the shard's parent.

- **AdjacentParentShardId** (*string*) –

The shard Id of the shard adjacent to the shard's parent.

- **HashKeyRange** (*dict*) –

The range of possible hash key values for the shard, which is a set of ordered contiguous positive integers.

- **StartingHashKey** (*string*) –

The starting hash key of the hash key range.

- **EndingHashKey** (*string*) –

The ending hash key of the hash key range.

- **SequenceNumberRange** (*dict*) –

The range of possible sequence numbers for the shard.

- **StartingSequenceNumber** (*string*) –

The starting sequence number for the range.

- **EndingSequenceNumber** (*string*) –

The ending sequence number for the range. Shards that are in the `OPEN` state have an ending sequence number of `null`.

* **HasMoreShards** (*boolean*) –

If set to `true`, more shards in the stream are available to describe.

– **NextToken** (*string*) –

A token to resume pagination.

class `Kinesis.Paginator.ListStreams`

```
paginator = client.get_paginator('list_streams')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Kinesis.Client.list_streams()`.

Request Syntax

```
response_iterator = paginator.paginate(
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns**Response Syntax**

```
{
    'StreamNames': [
        'string',
    ],
    'HasMoreStreams': True|False,
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output for `ListStreams`.

- **StreamNames** (*list*) –

The names of the streams that are associated with the AWS account making the `ListStreams` request.

* (*string*) –

- **HasMoreStreams** (*boolean*) –

If set to `true`, there are more streams available to list.

- **NextToken** (*string*) –

A token to resume pagination.

Waiters

The available waiters are:

- `Kinesis.Waiter.StreamExists`

class `Kinesis.Waiter.StreamExists`

```
waiter = client.get_waiter('stream_exists')
```

wait (***kwargs*)

Polls `Kinesis.Client.describe_stream()` every 10 seconds until a successful state is reached. An error is returned after 18 failed checks.

Request Syntax

```
waiter.wait(
    StreamName='string',
    Limit=123,
    ExclusiveStartShardId='string'
)
```

Parameters

- **StreamName** (*string*) – [REQUIRED]
The name of the stream to describe.
- **Limit** (*integer*) – The maximum number of shards to return.
- **ExclusiveStartShardId** (*string*) – The shard ID of the shard to start with.

Returns None

KMS

Table of Contents

- [KMS](#)
 - [Client](#)

Client

class `KMS.Client`

A low-level client representing AWS Key Management Service (KMS):

```
import boto3
```

```
client = boto3.client('kms')
```

These are the available methods:

- `can_paginate()`
- `create_alias()`
- `create_grant()`
- `create_key()`
- `decrypt()`

- `delete_alias()`
- `describe_key()`
- `disable_key()`
- `disable_key_rotation()`
- `enable_key()`
- `enable_key_rotation()`
- `encrypt()`
- `generate_data_key()`
- `generate_data_key_without_plaintext()`
- `generate_presigned_url()`
- `generate_random()`
- `get_key_policy()`
- `get_key_rotation_status()`
- `get_paginator()`
- `get_waiter()`
- `list_aliases()`
- `list_grants()`
- `list_key_policies()`
- `list_keys()`
- `put_key_policy()`
- `re_encrypt()`
- `retire_grant()`
- `revoke_grant()`
- `update_alias()`
- `update_key_description()`

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you’d normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

create_alias (***kwargs*)

Creates a display name for a customer master key. An alias can be used to identify a key and should be unique. The console enforces a one-to-one mapping between the alias and a key. An alias name can contain only alphanumeric characters, forward slashes (/), underscores (_), and dashes (-). An alias must start with the word “alias” followed by a forward slash (alias/). An alias that begins with “aws” after the forward slash (alias/aws...) is reserved by Amazon Web Services (AWS).

To associate an alias with a different key, call `UpdateAlias`.

Note that you cannot create or update an alias that represents a key in another account.

Request Syntax

```
response = client.create_alias(  
    AliasName='string',  
    TargetKeyId='string'  
)
```

Parameters

- **AliasName** (*string*) – [REQUIRED]

String that contains the display name. The name must start with the word “alias” followed by a forward slash (alias/). Aliases that begin with “alias/AWS” are

reserved.

- **TargetKeyId** (*string*) – [REQUIRED]

An identifier of the key for which you are creating the alias. This value cannot be another alias but can be a globally unique identifier or a fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Returns None

create_grant (***kwargs*)

Adds a grant to a key to specify who can access the key and under what conditions. Grants are alternate permission mechanisms to key policies. For more information about grants, see [Grants](#) in the developer guide. If a grant is absent, access to the key is evaluated based on IAM policies attached to the user.

- [ListGrants](#)
- [RetireGrant](#)
- [RevokeGrant](#)

Request Syntax

```
response = client.create_grant(
    KeyId='string',
    GranteePrincipal='string',
    RetiringPrincipal='string',
    Operations=[
        'Decrypt'|'Encrypt'|'GenerateDataKey'|'GenerateDataKeyWithoutPlaintext'|'ReEncryptF
    ],
    Constraints={
        'EncryptionContextSubset': {
            'string': 'string'
        },
        'EncryptionContextEquals': {
            'string': 'string'
        }
    },
    GrantTokens=[
        'string',
    ]
)
```

Parameters

- **KeyId** (*string*) – [REQUIRED]

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

- **GranteePrincipal** (*string*) – [REQUIRED]

Principal given permission by the grant to use the key identified by the `KeyId` parameter.

- **RetiringPrincipal** (*string*) – Principal given permission to retire the grant. For more information, see [RetireGrant](#).
- **Operations** (*list*) – List of operations permitted by the grant. This can be any combination of one or more of the following values:

- Decrypt
- Encrypt
- GenerateDataKey
- GenerateDataKeyWithoutPlaintext
- ReEncryptFrom
- ReEncryptTo
- CreateGrant
- RetireGrant
- *(string)* –
- **Constraints** (*dict*) – Specifies the conditions under which the actions specified by the `Operations` parameter are allowed.
 - **EncryptionContextSubset** (*dict*) – The constraint equals the full encryption context.
 - * *(string)* –
 - *(string)* –
 - **EncryptionContextEquals** (*dict*) – The constraint contains additional key/value pairs that serve to further limit the grant.
 - * *(string)* –
 - *(string)* –
- **GrantTokens** (*list*) – For more information, see [Grant Tokens](#) .
 - *(string)* –

Return type dict

Returns

Response Syntax

```
{
    'GrantToken': 'string',
    'GrantId': 'string'
}
```

Response Structure

- (*dict*) –
 - **GrantToken** (*string*) –
For more information, see [Grant Tokens](#) .
 - **GrantId** (*string*) –

Unique grant identifier. You can use the *GrantId* value to revoke a grant.

create_key (***kwargs*)

Creates a customer master key. Customer master keys can be used to encrypt small amounts of data (less than 4K) directly, but they are most commonly used to encrypt or envelope data keys that are then used to encrypt customer data. For more information about data keys, see `GenerateDataKey` and `GenerateDataKeyWithoutPlaintext` .

Request Syntax

```
response = client.create_key(
    Policy='string',
    Description='string',
    KeyUsage='ENCRYPT_DECRYPT'
)
```

Parameters

- **Policy** (*string*) – Policy to be attached to the key. This is required and delegates back to the account. The key is the root of trust.

- **Description** (*string*) – Description of the key. We recommend that you choose a description that helps your customer decide whether the key is appropriate for a task.
- **KeyUsage** (*string*) – Specifies the intended use of the key. Currently this defaults to ENCRYPT/DECRYPT, and only symmetric encryption and decryption are supported.

Return type dict

Returns

Response Syntax

```
{
  'KeyMetadata': {
    'AWSAccountId': 'string',
    'KeyId': 'string',
    'Arn': 'string',
    'CreationDate': datetime(2015, 1, 1),
    'Enabled': True|False,
    'Description': 'string',
    'KeyUsage': 'ENCRYPT_DECRYPT'
  }
}
```

Response Structure

- (*dict*) –
 - **KeyMetadata** (*dict*) –

Metadata associated with the key.

 - * **AWSAccountId** (*string*) –

Account ID number.
 - * **KeyId** (*string*) –

Unique identifier for the key.
 - * **Arn** (*string*) –

Key ARN (Amazon Resource Name).
 - * **CreationDate** (*datetime*) –

Date the key was created.
 - * **Enabled** (*boolean*) –

Value that specifies whether the key is enabled.
 - * **Description** (*string*) –

The description of the key.
 - * **KeyUsage** (*string*) –

A value that specifies what operation(s) the key can perform.

decrypt (**kwargs)

Decrypts ciphertext. Ciphertext is plaintext that has been previously encrypted by using any of the following functions:

- `GenerateDataKey`
- `GenerateDataKeyWithoutPlaintext`
- `Encrypt`

Note that if a caller has been granted access permissions to all keys (through, for example, IAM user policies that grant `Decrypt` permission on all resources), then ciphertext encrypted by using keys in other accounts where the key grants access to the caller can be decrypted. To remedy this, we recommend that you do not grant `Decrypt` access in an IAM user policy. Instead grant `Decrypt` access only in

key policies. If you must grant `Decrypt` access in an IAM user policy, you should scope the resource to specific keys or to specific trusted accounts.

Request Syntax

```
response = client.decrypt(
    CiphertextBlob=b'bytes',
    EncryptionContext={
        'string': 'string'
    },
    GrantTokens=[
        'string',
    ]
)
```

Parameters

- **CiphertextBlob** (*bytes*) – [REQUIRED]
Ciphertext to be decrypted. The blob includes metadata.
- **EncryptionContext** (*dict*) – The encryption context. If this was specified in the `Encrypt` function, it must be specified here or the decryption operation will fail. For more information, see [Encryption Context](#) .
 - (*string*) –
 - * (*string*) –
- **GrantTokens** (*list*) – For more information, see [Grant Tokens](#) .
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'KeyId': 'string',
    'Plaintext': b'bytes'
}
```

Response Structure

- (*dict*) –
 - **KeyId** (*string*) –
ARN of the key used to perform the decryption. This value is returned if no errors are encountered during the operation.
 - **Plaintext** (*bytes*) –
Decrypted plaintext data. This value may not be returned if the customer master key is not available or if you didn’t have permission to use it.

delete_alias (***kwargs*)

Deletes the specified alias. To associate an alias with a different key, call `UpdateAlias` .

Request Syntax

```
response = client.delete_alias(
    AliasName='string'
)
```

Parameters **AliasName** (*string*) – [REQUIRED]

The alias to be deleted. The name must start with the word “alias” followed by a forward slash (alias/). Aliases that begin with “alias/AWS” are reserved.

Returns None

describe_key (***kwargs*)

Provides detailed information about the specified customer master key.

Request Syntax

```
response = client.describe_key(
    KeyId='string'
)
```

Parameters **KeyId** (*string*) – **[REQUIRED]**

A unique identifier for the customer master key. This value can be a globally unique identifier, a fully specified ARN to either an alias or a key, or an alias name prefixed by “alias/”.

- **Key ARN Example** - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- **Alias ARN Example** - arn:aws:kms:us-east-1:123456789012:alias/MyAliasName
- **Globally Unique Key ID Example** - 12345678-1234-1234-1234-123456789012
- **Alias Name Example** - alias/MyAliasName

Return type dict

Returns

Response Syntax

```
{
  'KeyMetadata': {
    'AWSAccountId': 'string',
    'KeyId': 'string',
    'Arn': 'string',
    'CreationDate': datetime(2015, 1, 1),
    'Enabled': True|False,
    'Description': 'string',
    'KeyUsage': 'ENCRYPT_DECRYPT'
  }
}
```

Response Structure

- (*dict*) –
 - **KeyMetadata** (*dict*) –

Metadata associated with the key.

 - * **AWSAccountId** (*string*) –

Account ID number.
 - * **KeyId** (*string*) –

Unique identifier for the key.
 - * **Arn** (*string*) –

Key ARN (Amazon Resource Name).
 - * **CreationDate** (*datetime*) –

Date the key was created.
 - * **Enabled** (*boolean*) –

Value that specifies whether the key is enabled.

* **Description** (*string*) –

The description of the key.

* **KeyUsage** (*string*) –

A value that specifies what operation(s) the key can perform.

disable_key (***kwargs*)

Marks a key as disabled, thereby preventing its use.

Request Syntax

```
response = client.disable_key(  
    KeyId='string'  
)
```

Parameters **KeyId** (*string*) – **[REQUIRED]**

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Returns None

disable_key_rotation (***kwargs*)

Disables rotation of the specified key.

Request Syntax

```
response = client.disable_key_rotation(  
    KeyId='string'  
)
```

Parameters **KeyId** (*string*) – **[REQUIRED]**

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Returns None

enable_key (***kwargs*)

Marks a key as enabled, thereby permitting its use. You can have up to 25 enabled keys at one time.

Request Syntax

```
response = client.enable_key(  
    KeyId='string'  
)
```

Parameters **KeyId** (*string*) – **[REQUIRED]**

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Returns None

enable_key_rotation (**kwargs)

Enables rotation of the specified customer master key.

Request Syntax

```
response = client.enable_key_rotation(
    KeyId='string'
)
```

Parameters **KeyId** (*string*) – [REQUIRED]

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Returns None

encrypt (**kwargs)

Encrypts plaintext into ciphertext by using a customer master key. The `Encrypt` function has two primary use cases:

- You can encrypt up to 4 KB of arbitrary data such as an RSA key, a database password, or other sensitive customer information.
- If you are moving encrypted data from one region to another, you can use this API to encrypt in the new region the plaintext data key that was used to encrypt the data in the original region. This provides you with an encrypted copy of the data key that can be decrypted in the new region and used there to decrypt the encrypted data.

Unless you are moving encrypted data from one region to another, you don't use this function to encrypt a generated data key within a region. You retrieve data keys already encrypted by calling the `GenerateDataKey` or `GenerateDataKeyWithoutPlaintext` function. Data keys don't need to be encrypted again by calling `Encrypt`.

If you want to encrypt data locally in your application, you can use the `GenerateDataKey` function to return a plaintext data encryption key and a copy of the key encrypted under the customer master key (CMK) of your choosing.

Request Syntax

```
response = client.encrypt(
    KeyId='string',
    Plaintext=b'bytes',
    EncryptionContext={
        'string': 'string'
    },
    GrantTokens=[
        'string',
    ]
)
```

Parameters

- **KeyId** (*string*) – [REQUIRED]

A unique identifier for the customer master key. This value can be a globally unique identifier, a fully specified ARN to either an alias or a key, or an alias name prefixed by “alias/”.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012

- Alias ARN Example - arn:aws:kms:us-east-1:123456789012:alias/MyAliasName
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012
- Alias Name Example - alias/MyAliasName
- **Plaintext** (*bytes*) – **[REQUIRED]**
Data to be encrypted.
- **EncryptionContext** (*dict*) – Name/value pair that specifies the encryption context to be used for authenticated encryption. If used here, the same value must be supplied to the `Decrypt` API or decryption will fail. For more information, see [Encryption Context](#) .
 - (*string*) –
 - * (*string*) –
- **GrantTokens** (*list*) – For more information, see [Grant Tokens](#) .
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'CiphertextBlob': b'bytes',
    'KeyId': 'string'
}
```

Response Structure

- (*dict*) –
 - **CiphertextBlob** (*bytes*) –
The encrypted plaintext. If you are using the CLI, the value is Base64 encoded. Otherwise, it is not encoded.
 - **KeyId** (*string*) –
The ID of the key used during encryption.

generate_data_key (***kwargs*)

Generates a data key that you can use in your application to locally encrypt data. This call returns a plaintext version of the key in the `Plaintext` field of the response object and an encrypted copy of the key in the `CiphertextBlob` field. The key is encrypted by using the master key specified by the `KeyId` field. To decrypt the encrypted key, pass it to the `Decrypt` API.

We recommend that you use the following pattern to locally encrypt data: call the `GenerateDataKey` API, use the key returned in the `Plaintext` response field to locally encrypt data, and then erase the plaintext data key from memory. Store the encrypted data key (contained in the `CiphertextBlob` field) alongside of the locally encrypted data.

Note: You should not call the `Encrypt` function to re-encrypt your data keys within a region. `GenerateDataKey` always returns the data key encrypted and tied to the customer master key that will be used to decrypt it. There is no need to decrypt it twice.

If you decide to use the optional `EncryptionContext` parameter, you must also store the context in full or at least store enough information along with the encrypted data to be able to reconstruct the context when submitting the ciphertext to the `Decrypt` API. It is a good practice to choose a context that you can reconstruct on the fly to better secure the ciphertext. For more information about how this parameter is used, see [Encryption Context](#) .

To decrypt data, pass the encrypted data key to the `Decrypt` API. `Decrypt` uses the associated master key to decrypt the encrypted data key and returns it as plaintext. Use the plaintext data key to locally decrypt your data and then erase the key from memory. You must specify the encryption context, if any, that you specified when you generated the key. The encryption context is logged by CloudTrail, and you can use this log to help track the use of particular data.

Request Syntax

```
response = client.generate_data_key(
    KeyId='string',
    EncryptionContext={
        'string': 'string'
    },
    NumberOfBytes=123,
    KeySpec='AES_256'|'AES_128',
    GrantTokens=[
        'string',
    ]
)
```

Parameters

- **KeyId** (*string*) – [REQUIRED]

A unique identifier for the customer master key. This value can be a globally unique identifier, a fully specified ARN to either an alias or a key, or an alias name prefixed by “alias/”.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Alias ARN Example - arn:aws:kms:us-east-1:123456789012:alias/MyAliasName
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012
- Alias Name Example - alias/MyAliasName

- **EncryptionContext** (*dict*) – Name/value pair that contains additional data to be authenticated during the encryption and decryption processes that use the key. This value is logged by AWS CloudTrail to provide context around the data encrypted by the key.

- (*string*) –
- * (*string*) –

- **NumberOfBytes** (*integer*) – Integer that contains the number of bytes to generate. Common values are 128, 256, 512, and 1024. 1024 is the current limit. We recommend that you use the `KeySpec` parameter instead.
- **KeySpec** (*string*) – Value that identifies the encryption algorithm and key size to generate a data key for. Currently this can be `AES_128` or `AES_256`.
- **GrantTokens** (*list*) – For more information, see [Grant Tokens](#).
- (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'CiphertextBlob': b'bytes',
    'Plaintext': b'bytes',
    'KeyId': 'string'
}
```

Response Structure

- *(dict)* –
 - **CiphertextBlob** (*bytes*) –

Ciphertext that contains the encrypted data key. You must store the blob and enough information to reconstruct the encryption context so that the data encrypted by using the key can later be decrypted. You must provide both the ciphertext blob and the encryption context to the Decrypt API to recover the plaintext data key and decrypt the object.

If you are using the CLI, the value is Base64 encoded. Otherwise, it is not encoded.
 - **Plaintext** (*bytes*) –

Plaintext that contains the data key. Use this for encryption and decryption and then remove it from memory as soon as possible.
 - **KeyId** (*string*) –

System generated unique identifier of the key to be used to decrypt the encrypted copy of the data key.

`generate_data_key_without_plaintext` (***kwargs*)

Returns a data key encrypted by a customer master key without the plaintext copy of that key. Otherwise, this API functions exactly like `GenerateDataKey`. You can use this API to, for example, satisfy an audit requirement that an encrypted key be made available without exposing the plaintext copy of that key.

Request Syntax

```
response = client.generate_data_key_without_plaintext(  
    KeyId='string',  
    EncryptionContext={  
        'string': 'string'  
    },  
    KeySpec='AES_256'|'AES_128',  
    NumberOfBytes=123,  
    GrantTokens=[  
        'string',  
    ]  
)
```

Parameters

- **KeyId** (*string*) – [REQUIRED]

A unique identifier for the customer master key. This value can be a globally unique identifier, a fully specified ARN to either an alias or a key, or an alias name prefixed by “alias/”.

– Key	ARN	Example	–	arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
– Alias	ARN	Example	–	arn:aws:kms:us-east-1:123456789012:alias/MyAliasName
– Globally Unique Key ID	Example	–	12345678-1234-1234-1234-123456789012	
– Alias Name	Example	–	alias/MyAliasName	
- **EncryptionContext** (*dict*) – Name:value pair that contains additional data to be authenticated during the encryption and decryption processes.
 - (*string*) –
 - * (*string*) –

- **KeySpec** (*string*) – Value that identifies the encryption algorithm and key size. Currently this can be AES_128 or AES_256.
- **NumberOfBytes** (*integer*) – Integer that contains the number of bytes to generate. Common values are 128, 256, 512, 1024 and so on. We recommend that you use the `KeySpec` parameter instead.
- **GrantTokens** (*list*) – For more information, see [Grant Tokens](#).
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'CiphertextBlob': b'bytes',
    'KeyId': 'string'
}
```

Response Structure

- (*dict*) –
 - **CiphertextBlob** (*bytes*) –

Ciphertext that contains the wrapped data key. You must store the blob and encryption context so that the key can be used in a future decrypt operation.

If you are using the CLI, the value is Base64 encoded. Otherwise, it is not encoded.
 - **KeyId** (*string*) –

System generated unique identifier of the key to be used to decrypt the encrypted copy of the data key.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

generate_random (***kwargs*)

Generates an unpredictable byte string.

Request Syntax

```
response = client.generate_random(
    NumberOfBytes=123
)
```

Parameters **NumberOfBytes** (*integer*) – Integer that contains the number of bytes to generate. Common values are 128, 256, 512, 1024 and so on. The current limit is 1024 bytes.

Return type dict

Returns

Response Syntax

```
{
    'Plaintext': b'bytes'
}
```

Response Structure

- (*dict*) –
 - **Plaintext** (*bytes*) –
Plaintext that contains the unpredictable byte string.

get_key_policy (***kwargs*)

Retrieves a policy attached to the specified key.

Request Syntax

```
response = client.get_key_policy(
    KeyId='string',
    PolicyName='string'
)
```

Parameters

- **KeyId** (*string*) – **[REQUIRED]**

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

- **PolicyName** (*string*) – **[REQUIRED]**

String that contains the name of the policy. Currently, this must be “default”. Policy names can be discovered by calling ListKeyPolicies .

Return type dict

Returns**Response Syntax**

```
{
    'Policy': 'string'
}
```

Response Structure

- (*dict*) –
 - **Policy** (*string*) –
A policy document in JSON format.

get_key_rotation_status (***kwargs*)

Retrieves a Boolean value that indicates whether key rotation is enabled for the specified key.

Request Syntax

```
response = client.get_key_rotation_status(
    KeyId='string'
)
```

Parameters `KeyId` (*string*) – [REQUIRED]

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- **Key ARN Example** - `arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012`
- **Globally Unique Key ID Example** - `12345678-1234-1234-1234-123456789012`

Return type dict

Returns**Response Syntax**

```
{
    'KeyRotationEnabled': True|False
}
```

Response Structure

- (*dict*) –
 - **KeyRotationEnabled** (*boolean*) – A Boolean value that specifies whether key rotation is enabled.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters `operation_name` (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type L{botocore.paginate.Paginator}

Returns A paginator object.

get_waiter (*waiter_name*)**list_aliases** (***kwargs*)

Lists all of the key aliases in the account.

Request Syntax

```
response = client.list_aliases(
    Limit=123,
    Marker='string'
)
```

Parameters

- **Limit** (*integer*) – Specify this parameter when paginating results to indicate the maximum number of aliases you want in each response. If there are additional aliases beyond the maximum you specify, the `Truncated` response element will be set to `true`.
- **Marker** (*string*) – Use this parameter when paginating results, and only in a subsequent request after you've received a response where the results are truncated. Set it to the value of the `NextMarker` element in the response you just received.

Return type dict

Returns**Response Syntax**

```
{
  'Aliases': [
    {
      'AliasName': 'string',
      'AliasArn': 'string',
      'TargetKeyId': 'string'
    },
  ],
  'NextMarker': 'string',
  'Truncated': True|False
}
```

Response Structure

- *(dict)* –
 - **Aliases** (*list*) –

A list of key aliases in the user’s account.

 - * *(dict)* – Contains information about an alias.
 - **AliasName** (*string*) –

String that contains the alias.
 - **AliasArn** (*string*) –

String that contains the key ARN.
 - **TargetKeyId** (*string*) –

String that contains the key identifier pointed to by the alias.
 - **NextMarker** (*string*) –

If `Truncated` is true, this value is present and contains the value to use for the `Marker` request parameter in a subsequent pagination request.
 - **Truncated** (*boolean*) –

A flag that indicates whether there are more items in the list. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more aliases in the list.

`list_grants` (**kwargs)

List the grants for a specified key.

Request Syntax

```
response = client.list_grants(
    KeyId='string',
    Limit=123,
    Marker='string'
)
```

Parameters

- **KeyId** (*string*) – [REQUIRED]

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

 - Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
 - Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012
- **Limit** (*integer*) – Specify this parameter only when paginating results to indicate the maximum number of grants you want listed in the response. If there are

additional grants beyond the maximum you specify, the `Truncated` response element will be set to `true`.

- **Marker** (*string*) – Use this parameter only when paginating results, and only in a subsequent request after you’ve received a response where the results are truncated. Set it to the value of the `NextMarker` in the response you just received.

Return type dict

Returns

Response Syntax

```
{
  'Grants': [
    {
      'GrantId': 'string',
      'GranteePrincipal': 'string',
      'RetiringPrincipal': 'string',
      'IssuingAccount': 'string',
      'Operations': [
        'Decrypt'|'Encrypt'|'GenerateDataKey'|'GenerateDataKeyWithoutPlain
      ],
      'Constraints': {
        'EncryptionContextSubset': {
          'string': 'string'
        },
        'EncryptionContextEquals': {
          'string': 'string'
        }
      }
    },
  ],
  'NextMarker': 'string',
  'Truncated': True|False
}
```

Response Structure

- (*dict*) –
 - **Grants** (*list*) –

A list of grants.

- * (*dict*) –

Contains information about each entry in the grant list.

- **GrantId** (*string*) –
Unique grant identifier.
- **GranteePrincipal** (*string*) –
The principal that receives the grant permission.
- **RetiringPrincipal** (*string*) –
The principal that can retire the account.
- **IssuingAccount** (*string*) –
The account under which the grant was issued.
- **Operations** (*list*) –
List of operations permitted by the grant. This can be any combination of one or more of the following values:
 - Decrypt

- `Encrypt`
- `GenerateDataKey`
- `GenerateDataKeyWithoutPlaintext`
- `ReEncryptFrom`
- `ReEncryptTo`
- `CreateGrant`
- *(string)* –
- **Constraints** (*dict*) –

Specifies the conditions under which the actions specified by the `Operations` parameter are allowed.

- **EncryptionContextSubset** (*dict*) – The constraint equals the full encryption context.
- *(string)* –
- *(string)* –
- **EncryptionContextEquals** (*dict*) – The constraint contains additional key/value pairs that serve to further limit the grant.
- *(string)* –
- *(string)* –

- **NextMarker** (*string*) –

If `Truncated` is true, this value is present and contains the value to use for the `Marker` request parameter in a subsequent pagination request.

- **Truncated** (*boolean*) –

A flag that indicates whether there are more items in the list. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more grants in the list.

list_key_policies (***kwargs*)

Retrieves a list of policies attached to a key.

Request Syntax

```
response = client.list_key_policies(  
    KeyId='string',  
    Limit=123,  
    Marker='string'  
)
```

Parameters

- **KeyId** (*string*) – [REQUIRED]

A unique identifier for the customer master key. This value can be a globally unique identifier, a fully specified ARN to either an alias or a key, or an alias name prefixed by “alias/”.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Alias ARN Example - arn:aws:kms:us-east-1:123456789012:alias/MyAliasName
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012
- Alias Name Example - alias/MyAliasName

- **Limit** (*integer*) – Specify this parameter only when paginating results to indicate the maximum number of policies you want listed in the response. If there are additional policies beyond the maximum you specify, the `Truncated` response element will be set to `true`.

- **Marker** (*string*) – Use this parameter only when paginating results, and only in a subsequent request after you’ve received a response where the results are truncated. Set it to the value of the `NextMarker` in the response you just received.

Return type dict

Returns

Response Syntax

```
{
    'PolicyNames': [
        'string',
    ],
    'NextMarker': 'string',
    'Truncated': True|False
}
```

Response Structure

- (*dict*) –
 - **PolicyNames** (*list*) –

A list of policy names. Currently, there is only one policy and it is named “Default”.

* (*string*) –
 - **NextMarker** (*string*) –

If `Truncated` is true, this value is present and contains the value to use for the `Marker` request parameter in a subsequent pagination request.
 - **Truncated** (*boolean*) –

A flag that indicates whether there are more items in the list. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more policies in the list.

list_keys (***kwargs*)

Lists the customer master keys.

Request Syntax

```
response = client.list_keys(
    Limit=123,
    Marker='string'
)
```

Parameters

- **Limit** (*integer*) – Specify this parameter only when paginating results to indicate the maximum number of keys you want listed in the response. If there are additional keys beyond the maximum you specify, the `Truncated` response element will be set to `true`.
- **Marker** (*string*) – Use this parameter only when paginating results, and only in a subsequent request after you’ve received a response where the results are truncated. Set it to the value of the `NextMarker` in the response you just received.

Return type dict

Returns

Response Syntax

```
{
  'Keys': [
    {
      'KeyId': 'string',
      'KeyArn': 'string'
    },
  ],
  'NextMarker': 'string',
  'Truncated': True|False
}
```

Response Structure

- *(dict)* –
 - **Keys** (*list*) –
A list of keys.
 - * *(dict)* –
Contains information about each entry in the key list.
 - **KeyId** (*string*) –
Unique identifier of the key.
 - **KeyArn** (*string*) –
ARN of the key.
 - **NextMarker** (*string*) –
If `Truncated` is true, this value is present and contains the value to use for the `Marker` request parameter in a subsequent pagination request.
 - **Truncated** (*boolean*) –
A flag that indicates whether there are more items in the list. If your results were truncated, you can make a subsequent pagination request using the `Marker` request parameter to retrieve more keys in the list.

put_key_policy (***kwargs*)

Attaches a policy to the specified key.

Request Syntax

```
response = client.put_key_policy(
    KeyId='string',
    PolicyName='string',
    Policy='string'
)
```

Parameters

- **KeyId** (*string*) – [REQUIRED]
A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.
 - Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
 - Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012
- **PolicyName** (*string*) – [REQUIRED]
Name of the policy to be attached. Currently, the only supported name is “default”.

- **Policy** (*string*) – [REQUIRED]

The policy, in JSON format, to be attached to the key.

Returns None

re_encrypt (***kwargs*)

Encrypts data on the server side with a new customer master key without exposing the plaintext of the data on the client side. The data is first decrypted and then encrypted. This operation can also be used to change the encryption context of a ciphertext.

Unlike other actions, ReEncrypt is authorized twice - once as ReEncryptFrom on the source key and once as ReEncryptTo on the destination key. We therefore recommend that you include the "action": "kms:ReEncrypt*" statement in your key policies to permit re-encryption from or to the key. The statement is included automatically when you authorize use of the key through the console but must be included manually when you set a policy by using the PutKeyPolicy function.

Request Syntax

```
response = client.re_encrypt(
    CiphertextBlob=b'bytes',
    SourceEncryptionContext={
        'string': 'string'
    },
    DestinationKeyId='string',
    DestinationEncryptionContext={
        'string': 'string'
    },
    GrantTokens=[
        'string',
    ]
)
```

Parameters

- **CiphertextBlob** (*bytes*) – [REQUIRED]

Ciphertext of the data to re-encrypt.

- **SourceEncryptionContext** (*dict*) – Encryption context used to encrypt and decrypt the data specified in the CiphertextBlob parameter.

- (*string*) –
- * (*string*) –

- **DestinationKeyId** (*string*) – [REQUIRED]

A unique identifier for the customer master key used to re-encrypt the data. This value can be a globally unique identifier, a fully specified ARN to either an alias or a key, or an alias name prefixed by “alias”.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Alias ARN Example - arn:aws:kms:us-east-1:123456789012:alias/MyAliasName
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012
- Alias Name Example - alias/MyAliasName

- **DestinationEncryptionContext** (*dict*) – Encryption context to be used when the data is re-encrypted.

- (*string*) –
- * (*string*) –

- **GrantTokens** (*list*) – For more information, see [Grant Tokens](#).

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'CiphertextBlob': b'bytes',
    'SourceKeyId': 'string',
    'KeyId': 'string'
}
```

Response Structure

- (*dict*) –
 - **CiphertextBlob** (*bytes*) –
The re-encrypted data. If you are using the CLI, the value is Base64 encoded. Otherwise, it is not encoded.
 - **SourceKeyId** (*string*) –
Unique identifier of the key used to originally encrypt the data.
 - **KeyId** (*string*) –
Unique identifier of the key used to re-encrypt the data.

retire_grant (***kwargs*)

Retires a grant. You can retire a grant when you're done using it to clean up. You should revoke a grant when you intend to actively deny operations that depend on it. The following are permitted to call this API:

- The account that created the grant
- The `RetiringPrincipal`, if present
- The `GranteePrincipal`, if `RetireGrant` is a grantee operation

The grant to retire must be identified by its grant token or by a combination of the key ARN and the grant ID. A grant token is a unique variable-length base64-encoded string. A grant ID is a 64 character unique identifier of a grant. Both are returned by the `CreateGrant` function.

Request Syntax

```
response = client.retire_grant(
    GrantToken='string',
    KeyId='string',
    GrantId='string'
)
```

Parameters

- **GrantToken** (*string*) – Token that identifies the grant to be retired.
- **KeyId** (*string*) – A unique identifier for the customer master key associated with the grant. This value can be a globally unique identifier or a fully specified ARN of the key.
 - Key ARN Example - `arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012`
 - Globally Unique Key ID Example - `12345678-1234-1234-1234-123456789012`
- **GrantId** (*string*) – Unique identifier of the grant to be retired. The grant ID is returned by the `CreateGrant` function.
 - Grant ID Example - `012345678901234567890123456789012345678901234567890123456789012`

Returns None

revoke_grant (**kwargs)

Revokes a grant. You can revoke a grant to actively deny operations that depend on it.

Request Syntax

```
response = client.revoke_grant(
    KeyId='string',
    GrantId='string'
)
```

Parameters

- **KeyId** (*string*) – [REQUIRED]

A unique identifier for the customer master key associated with the grant. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

- **GrantId** (*string*) – [REQUIRED]

Identifier of the grant to be revoked.

Returns None

update_alias (**kwargs)

Updates an alias to associate it with a different key.

An alias name can contain only alphanumeric characters, forward slashes (/), underscores (_), and dashes (-). An alias must start with the word “alias” followed by a forward slash (alias/). An alias that begins with “aws” after the forward slash (alias/aws...) is reserved by Amazon Web Services (AWS).

An alias is not a property of a key. Therefore, an alias can be associated with and disassociated from an existing key without changing the properties of the key.

Note that you cannot create or update an alias that represents a key in another account.

Request Syntax

```
response = client.update_alias(
    AliasName='string',
    TargetKeyId='string'
)
```

Parameters

- **AliasName** (*string*) – [REQUIRED] String that contains the name of the alias to be modified. The name must start with the word “alias” followed by a forward slash (alias/). Aliases that begin with “alias/AWS” are reserved.
- **TargetKeyId** (*string*) – [REQUIRED]

Unique identifier of the customer master key to be associated with the alias. This value can be a globally unique identifier or the fully specified ARN of a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Returns None

update_key_description (**kwargs)

Updates the description of a key.

Request Syntax

```
response = client.update_key_description(  
    KeyId='string',  
    Description='string'  
)
```

Parameters

- **KeyId** (*string*) – [REQUIRED]

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

- **Description** (*string*) – [REQUIRED]

New description for the key.

Returns None

Lambda

Table of Contents

- [Lambda](#)
 - [Client](#)
 - [Paginators](#)

Client

`class Lambda.Client`

A low-level client representing AWS Lambda:

```
import boto3
```

```
client = boto3.client('lambda')
```

These are the available methods:

- `add_permission()`
- `can_paginate()`
- `create_event_source_mapping()`
- `create_function()`
- `delete_event_source_mapping()`
- `delete_function()`
- `generate_presigned_url()`
- `get_event_source_mapping()`
- `get_function()`
- `get_function_configuration()`
- `get_paginator()`
- `get_policy()`
- `get_waiter()`

- `invoke()`
- `invoke_async()`
- `list_event_source_mappings()`
- `list_functions()`
- `remove_permission()`
- `update_event_source_mapping()`
- `update_function_code()`
- `update_function_configuration()`

add_permission (***kwargs*)

Adds a permission to the access policy associated with the specified AWS Lambda function. In a “push event” model, the access policy attached to the Lambda function grants Amazon S3 or a user application permission for the Lambda `lambda:Invoke` action. For information about the push model, see [AWS Lambda: How it Works](#). Each Lambda function has one access policy associated with it. You can use the `AddPermission` API to add a permission to the policy. You have one access policy but it can have multiple permission statements.

This operation requires permission for the `lambda:AddPermission` action.

Request Syntax

```
response = client.add_permission(
    FunctionName='string',
    StatementId='string',
    Action='string',
    Principal='string',
    SourceArn='string',
    SourceAccount='string'
)
```

Parameters

- **FunctionName** (*string*) – [REQUIRED]

Name of the Lambda function whose access policy you are updating by adding a new permission.

You can specify an unqualified function name (for example, “Thumbnail”) or you can specify Amazon Resource Name (ARN) of the function (for example, “arn:aws:lambda:us-west-2:account-id:function:ThumbNail”). AWS Lambda also allows you to specify only the account ID qualifier (for example, “account-id:Thumbnail”). Note that the length constraint applies only to the ARN. If you specify only the function name, it is limited to 64 character in length.

- **StatementId** (*string*) – [REQUIRED]

A unique statement identifier.

- **Action** (*string*) – [REQUIRED]

The AWS Lambda action you want to allow in this statement. Each Lambda action is a string starting with “lambda:” followed by the API name (see [Operations](#)). For example, “lambda:CreateFunction”. You can use wildcard (“lambda:*)” to grant permission for all AWS Lambda actions.

- **Principal** (*string*) – [REQUIRED]

The principal who is getting this permission. It can be Amazon S3 service Principal (“s3.amazonaws.com”) if you want Amazon S3 to invoke the function, an AWS account ID if you are granting cross-account permission, or any valid AWS service principal such as “sns.amazonaws.com”. For example, you might want to allow a custom application in another AWS account to push events to AWS Lambda by invoking your function.

- **SourceArn** (*string*) – This is optional; however, when granting Amazon S3 permission to invoke your function, you should specify this field with the bucket Amazon Resource Name (ARN) as its value. This ensures that only events generated from the specified bucket can invoke the function.

Warning: If you add a permission for the Amazon S3 principal without providing the source ARN, any AWS account that creates a mapping to your function ARN can send events to invoke your Lambda function from Amazon S3.

- **SourceAccount** (*string*) – The AWS account ID (without a hyphen) of the source owner. For example, if the `SourceArn` identifies a bucket, then this is the bucket owner's account ID. You can use this additional condition to ensure the bucket you specify is owned by a specific account (it is possible the bucket owner deleted the bucket and some other AWS account created the bucket). You can also use this condition to specify all sources (that is, you don't specify the `SourceArn`) owned by a specific account.

Return type dict

Returns

Response Syntax

```
{
    'Statement': 'string'
}
```

Response Structure

- (*dict*) –
 - **Statement** (*string*) –
The permission statement you specified in the request. The response returns the same as a string using `""` as an escape character in the JSON.

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

create_event_source_mapping (***kwargs*)

Identifies a stream as an event source for a Lambda function. It can be either an Amazon Kinesis stream or an Amazon DynamoDB stream. AWS Lambda invokes the specified function when records are posted to the stream.

This is the pull model, where AWS Lambda invokes the function. For more information, go to [AWS Lambda: How it Works](#) in the *AWS Lambda Developer Guide*.

This association between an Amazon Kinesis stream and a Lambda function is called the event source mapping. You provide the configuration information (for example, which stream to read from and which Lambda function to invoke) for the event source mapping in the request body.

Each event source, such as an Amazon Kinesis or a DynamoDB stream, can be associated with multiple AWS Lambda function. A given Lambda function can be associated with multiple AWS event sources.

This operation requires permission for the `lambda:CreateEventSourceMapping` action.

Request Syntax

```
response = client.create_event_source_mapping(
    EventSourceArn='string',
    FunctionName='string',
    Enabled=True|False,
    BatchSize=123,
    StartingPosition='TRIM_HORIZON'|'LATEST'
)
```

Parameters

- **EventSourceArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the Amazon Kinesis or the Amazon DynamoDB stream that is the event source. Any record added to this stream could cause AWS Lambda to invoke your Lambda function, it depends on the `BatchSize`. AWS Lambda POSTs the Amazon Kinesis event, containing records, to your Lambda function as JSON.

- **FunctionName** (*string*) – [REQUIRED]

The Lambda function to invoke when AWS Lambda detects an event on the stream.

You can specify an unqualified function name (for example, “Thumbnail”) or you can specify Amazon Resource Name (ARN) of the function (for example, “arn:aws:lambda:us-west-2:account-id:function:ThumbNail”). AWS Lambda also allows you to specify only the account ID qualifier (for example, “account-id:Thumbnail”). Note that the length constraint applies only to the ARN. If you specify only the function name, it is limited to 64 character in length.

- **Enabled** (*boolean*) – Indicates whether AWS Lambda should begin polling the event source, the default is not enabled.
- **BatchSize** (*integer*) – The largest number of records that AWS Lambda will retrieve from your event source at the time of invoking your function. Your function receives an event with all the retrieved records. The default is 100 records.
- **StartingPosition** (*string*) – [REQUIRED]

The position in the stream where AWS Lambda should start reading. For more information, go to [ShardIteratorType](#) in the *Amazon Kinesis API Reference*.

Return type dict

Returns

Response Syntax

```
{
    'UUID': 'string',
    'BatchSize': 123,
    'EventSourceArn': 'string',
    'FunctionArn': 'string',
    'LastModified': datetime(2015, 1, 1),
    'LastProcessingResult': 'string',
    'State': 'string',
    'StateTransitionReason': 'string'
}
```

Response Structure

- (*dict*) –

Describes mapping between an Amazon Kinesis stream and a Lambda function.

- **UUID** (*string*) –
The AWS Lambda assigned opaque identifier for the mapping.
- **BatchSize** (*integer*) –
The largest number of records that AWS Lambda will retrieve from your event source at the time of invoking your function. Your function receives an event with all the retrieved records.
- **EventSourceArn** (*string*) –
The Amazon Resource Name (ARN) of the Amazon Kinesis stream that is the source of events.
- **FunctionArn** (*string*) –
The Lambda function to invoke when AWS Lambda detects an event on the stream.
- **LastModified** (*datetime*) –
The UTC time string indicating the last time the event mapping was updated.
- **LastProcessingResult** (*string*) –
The result of the last AWS Lambda invocation of your Lambda function.
- **State** (*string*) –
The state of the event source mapping. It can be “Creating”, “Enabled”, “Disabled”, “Enabling”, “Disabling”, “Updating”, or “Deleting”.
- **StateTransitionReason** (*string*) –
The reason the event source mapping is in its current state. It is either user-requested or an AWS Lambda-initiated state transition.

create_function (***kwargs*)

Creates a new Lambda function. The function metadata is created from the request parameters, and the code for the function is provided by a .zip file in the request body. If the function name already exists, the operation will fail. Note that the function name is case-sensitive.

This operation requires permission for the `lambda:CreateFunction` action.

Request Syntax

```
response = client.create_function(  
    FunctionName='string',  
    Runtime='nodejs'|'java8',  
    Role='string',  
    Handler='string',  
    Description='string',  
    Timeout=123,  
    MemorySize=123,  
    Code={  
        'ZipFile': b'bytes',  
        'S3Bucket': 'string',  
        'S3Key': 'string',  
        'S3ObjectVersion': 'string'  
    }  
)
```

Parameters

- **FunctionName** (*string*) – [REQUIRED]

The name you want to assign to the function you are uploading. You can specify an unqualified function name (for example, “Thumbnail”) or you can specify Amazon Resource Name (ARN) of the function (for example, “arn:aws:lambda:us-west-2:account-id:function:ThumbNail”). AWS Lambda also allows you to specify only the account ID qualifier (for example, “account-id:Thumbnail”). Note that the length constraint applies only to the ARN. If you specify only the function name, it is limited to 64 character in length. The function names appear in the console and are returned in the ListFunctions API. Function names are used to specify functions to other AWS Lambda APIs, such as Invoke .

- **Runtime** (*string*) – [REQUIRED]

The runtime environment for the Lambda function you are uploading. Currently, Lambda supports “java” and “nodejs” as the runtime.

- **Role** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the IAM role that Lambda assumes when it executes your function to access any other Amazon Web Services (AWS) resources. For more information, see [AWS Lambda: How it Works](#)

- **Handler** (*string*) – [REQUIRED]

The function within your code that Lambda calls to begin execution. For Node.js, it is the *module-name* *.*export** value in your function. For Java, it can be *package.class-name:handler* or *package.class-name* . For more information, see [Lambda Function Handler \(Java\)](#) .

- **Description** (*string*) – A short, user-defined function description. Lambda does not use this value. Assign a meaningful description as you see fit.
- **Timeout** (*integer*) – The function execution time at which Lambda should terminate the function. Because the execution time has cost implications, we recommend you set this value based on your expected execution time. The default is 3 seconds.
- **MemorySize** (*integer*) – The amount of memory, in MB, your Lambda function is given. Lambda uses this memory size to infer the amount of CPU and memory allocated to your function. Your function use-case determines your CPU and memory requirements. For example, a database operation might need less memory compared to an image processing function. The default value is 128 MB. The value must be a multiple of 64 MB.
- **Code** (*dict*) – [REQUIRED]

The code for the Lambda function.

- **ZipFile** (*bytes*) –

A base64-encoded .zip file containing your deployment package. For more information about creating a .zip file, go to [Execution Permissions](#) in the *AWS Lambda Developer Guide* .

- **S3Bucket** (*string*) –

Amazon S3 bucket name where the .zip file containing your deployment package is stored. This bucket must reside in the same AWS region where you are creating the Lambda function.

- **S3Key** (*string*) –

The Amazon S3 object (the deployment package) key name you want to upload.

- **S3ObjectVersion** (*string*) –

The Amazon S3 object (the deployment package) version you want to upload.

Return type dict

Returns

Response Syntax

```
{
    'FunctionName': 'string',
    'FunctionArn': 'string',
    'Runtime': 'nodejs'|'java8',
    'Role': 'string',
    'Handler': 'string',
    'CodeSize': 123,
    'Description': 'string',
    'Timeout': 123,
    'MemorySize': 123,
    'LastModified': 'string'
}
```

Response Structure

- (*dict*) –

A complex type that describes function metadata.

- **FunctionName** (*string*) –

The name of the function.

- **FunctionArn** (*string*) –

The Amazon Resource Name (ARN) assigned to the function.

- **Runtime** (*string*) –

The runtime environment for the Lambda function.

- **Role** (*string*) –

The Amazon Resource Name (ARN) of the IAM role that Lambda assumes when it executes your function to access any other Amazon Web Services (AWS) resources.

- **Handler** (*string*) –

The function Lambda calls to begin executing your function.

- **CodeSize** (*integer*) –

The size, in bytes, of the function .zip file you uploaded.

- **Description** (*string*) –

The user-provided description.

- **Timeout** (*integer*) –

The function execution time at which Lambda should terminate the function. Because the execution time has cost implications, we recommend you set this value based on your expected execution time. The default is 3 seconds.

- **MemorySize** (*integer*) –

The memory size, in MB, you configured for the function. Must be a multiple of 64 MB.

- **LastModified** (*string*) –

The timestamp of the last time you updated the function.

delete_event_source_mapping (**kwargs)

Removes an event source mapping. This means AWS Lambda will no longer invoke the function for events in the associated source.

This operation requires permission for the `lambda:DeleteEventSourceMapping` action.

Request Syntax

```
response = client.delete_event_source_mapping(
    UUID='string'
)
```

Parameters **UUID** (*string*) – [REQUIRED]

The event source mapping ID.

Return type dict

Returns**Response Syntax**

```
{
    'UUID': 'string',
    'BatchSize': 123,
    'EventSourceArn': 'string',
    'FunctionArn': 'string',
    'LastModified': datetime(2015, 1, 1),
    'LastProcessingResult': 'string',
    'State': 'string',
    'StateTransitionReason': 'string'
}
```

Response Structure

- (*dict*) –

Describes mapping between an Amazon Kinesis stream and a Lambda function.

- **UUID** (*string*) –

The AWS Lambda assigned opaque identifier for the mapping.

- **BatchSize** (*integer*) –

The largest number of records that AWS Lambda will retrieve from your event source at the time of invoking your function. Your function receives an event with all the retrieved records.

- **EventSourceArn** (*string*) –

The Amazon Resource Name (ARN) of the Amazon Kinesis stream that is the source of events.

- **FunctionArn** (*string*) –

The Lambda function to invoke when AWS Lambda detects an event on the stream.

- **LastModified** (*datetime*) –

The UTC time string indicating the last time the event mapping was updated.

- **LastProcessingResult** (*string*) –

The result of the last AWS Lambda invocation of your Lambda function.

- **State** (*string*) –

The state of the event source mapping. It can be “Creating”, “Enabled”, “Disabled”, “Enabling”, “Disabling”, “Updating”, or “Deleting”.

– **StateTransitionReason** (*string*) –

The reason the event source mapping is in its current state. It is either user-requested or an AWS Lambda-initiated state transition.

delete_function (***kwargs*)

Deletes the specified Lambda function code and configuration.

When you delete a function the associated access policy is also deleted. You will need to delete the event source mappings explicitly.

This operation requires permission for the `lambda:DeleteFunction` action.

Request Syntax

```
response = client.delete_function(  
    FunctionName='string'  
)
```

Parameters **FunctionName** (*string*) – [REQUIRED]

The Lambda function to delete.

You can specify an unqualified function name (for example, “Thumbnail”) or you can specify Amazon Resource Name (ARN) of the function (for example, “arn:aws:lambda:us-west-2:account-id:function:ThumbNail”). AWS Lambda also allows you to specify only the account ID qualifier (for example, “account-id:Thumbnail”). Note that the length constraint applies only to the ARN. If you specify only the function name, it is limited to 64 character in length.

Returns None

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method’s model.

Returns The presigned url

get_event_source_mapping (***kwargs*)

Returns configuration information for the specified event source mapping (see `CreateEventSourceMapping`).

This operation requires permission for the `lambda:GetEventSourceMapping` action.

Request Syntax

```
response = client.get_event_source_mapping(  
    UUID='string'  
)
```

Parameters **UUID** (*string*) – [REQUIRED]

The AWS Lambda assigned ID of the event source mapping.

Return type dict

Returns

Response Syntax

```
{
  'UUID': 'string',
  'BatchSize': 123,
  'EventSourceArn': 'string',
  'FunctionArn': 'string',
  'LastModified': datetime(2015, 1, 1),
  'LastProcessingResult': 'string',
  'State': 'string',
  'StateTransitionReason': 'string'
}
```

Response Structure

- (*dict*) –

Describes mapping between an Amazon Kinesis stream and a Lambda function.

- **UUID** (*string*) –

The AWS Lambda assigned opaque identifier for the mapping.

- **BatchSize** (*integer*) –

The largest number of records that AWS Lambda will retrieve from your event source at the time of invoking your function. Your function receives an event with all the retrieved records.

- **EventSourceArn** (*string*) –

The Amazon Resource Name (ARN) of the Amazon Kinesis stream that is the source of events.

- **FunctionArn** (*string*) –

The Lambda function to invoke when AWS Lambda detects an event on the stream.

- **LastModified** (*datetime*) –

The UTC time string indicating the last time the event mapping was updated.

- **LastProcessingResult** (*string*) –

The result of the last AWS Lambda invocation of your Lambda function.

- **State** (*string*) –

The state of the event source mapping. It can be “Creating”, “Enabled”, “Disabled”, “Enabling”, “Disabling”, “Updating”, or “Deleting”.

- **StateTransitionReason** (*string*) –

The reason the event source mapping is in its current state. It is either user-requested or an AWS Lambda-initiated state transition.

get_function (***kwargs*)

Returns the configuration information of the Lambda function and a presigned URL link to the .zip file you uploaded with `CreateFunction` so you can download the .zip file. Note that the URL is valid for up to 10 minutes. The configuration information is the same information you provided as parameters when uploading the function.

This operation requires permission for the `lambda:GetFunction` action.

Request Syntax

```
response = client.get_function(  
    FunctionName='string'  
)
```

Parameters **FunctionName** (*string*) – [REQUIRED]

The Lambda function name.

You can specify an unqualified function name (for example, “Thumbnail”) or you can specify Amazon Resource Name (ARN) of the function (for example, “arn:aws:lambda:us-west-2:account-id:function:ThumbNail”). AWS Lambda also allows you to specify only the account ID qualifier (for example, “account-id:Thumbnail”). Note that the length constraint applies only to the ARN. If you specify only the function name, it is limited to 64 character in length.

Return type dict**Returns****Response Syntax**

```
{  
    'Configuration': {  
        'FunctionName': 'string',  
        'FunctionArn': 'string',  
        'Runtime': 'nodejs'|'java8',  
        'Role': 'string',  
        'Handler': 'string',  
        'CodeSize': 123,  
        'Description': 'string',  
        'Timeout': 123,  
        'MemorySize': 123,  
        'LastModified': 'string'  
    },  
    'Code': {  
        'RepositoryType': 'string',  
        'Location': 'string'  
    }  
}
```

Response Structure

- (*dict*) –

This response contains the object for the Lambda function location (see `API_FunctionCodeLocation`

- **Configuration** (*dict*) –

A complex type that describes function metadata.

- * **FunctionName** (*string*) –

The name of the function.

- * **FunctionArn** (*string*) –

The Amazon Resource Name (ARN) assigned to the function.

- * **Runtime** (*string*) –

The runtime environment for the Lambda function.

- * **Role** (*string*) –

The Amazon Resource Name (ARN) of the IAM role that Lambda assumes when it executes your function to access any other Amazon Web Services (AWS) resources.

- * **Handler** (*string*) –
The function Lambda calls to begin executing your function.
- * **CodeSize** (*integer*) –
The size, in bytes, of the function .zip file you uploaded.
- * **Description** (*string*) –
The user-provided description.
- * **Timeout** (*integer*) –
The function execution time at which Lambda should terminate the function. Because the execution time has cost implications, we recommend you set this value based on your expected execution time. The default is 3 seconds.
- * **MemorySize** (*integer*) –
The memory size, in MB, you configured for the function. Must be a multiple of 64 MB.
- * **LastModified** (*string*) –
The timestamp of the last time you updated the function.
- **Code** (*dict*) –
The object for the Lambda function location.
 - * **RepositoryType** (*string*) –
The repository from which you can download the function.
 - * **Location** (*string*) –
The presigned URL you can use to download the function's .zip file that you previously uploaded. The URL is valid for up to 10 minutes.

get_function_configuration (***kwargs*)

Returns the configuration information of the Lambda function. This the same information you provided as parameters when uploading the function by using `CreateFunction` .

This operation requires permission for the `lambda:GetFunctionConfiguration` operation.

Request Syntax

```
response = client.get_function_configuration(
    FunctionName='string'
)
```

Parameters **FunctionName** (*string*) – [REQUIRED]

The name of the Lambda function for which you want to retrieve the configuration information.

You can specify an unqualified function name (for example, “Thumbnail”) or you can specify Amazon Resource Name (ARN) of the function (for example, “arn:aws:lambda:us-west-2:account-id:function:ThumbNail”). AWS Lambda also allows you to specify only the account ID qualifier (for example, “account-id:Thumbnail”). Note that the length constraint applies only to the ARN. If you specify only the function name, it is limited to 64 character in length.

Return type dict

Returns

Response Syntax

```
{
    'FunctionName': 'string',
    'FunctionArn': 'string',
    'Runtime': 'nodejs'|'java8',
    'Role': 'string',
    'Handler': 'string',
    'CodeSize': 123,
    'Description': 'string',
    'Timeout': 123,
    'MemorySize': 123,
    'LastModified': 'string'
}
```

Response Structure

- (*dict*) –

A complex type that describes function metadata.

- **FunctionName** (*string*) –

The name of the function.

- **FunctionArn** (*string*) –

The Amazon Resource Name (ARN) assigned to the function.

- **Runtime** (*string*) –

The runtime environment for the Lambda function.

- **Role** (*string*) –

The Amazon Resource Name (ARN) of the IAM role that Lambda assumes when it executes your function to access any other Amazon Web Services (AWS) resources.

- **Handler** (*string*) –

The function Lambda calls to begin executing your function.

- **CodeSize** (*integer*) –

The size, in bytes, of the function .zip file you uploaded.

- **Description** (*string*) –

The user-provided description.

- **Timeout** (*integer*) –

The function execution time at which Lambda should terminate the function. Because the execution time has cost implications, we recommend you set this value based on your expected execution time. The default is 3 seconds.

- **MemorySize** (*integer*) –

The memory size, in MB, you configured for the function. Must be a multiple of 64 MB.

- **LastModified** (*string*) –

The timestamp of the last time you updated the function.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`,

if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises `OperationNotPageableError` Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_policy (***kwargs*)

Returns the access policy, containing a list of permissions granted via the `AddPermission` API, associated with the specified bucket.

You need permission for the `lambda:GetPolicy` action.

Request Syntax

```
response = client.get_policy(
    FunctionName='string'
)
```

Parameters `FunctionName` (*string*) – **[REQUIRED]**

Function name whose access policy you want to retrieve.

You can specify an unqualified function name (for example, “Thumbnail”) or you can specify Amazon Resource Name (ARN) of the function (for example, “arn:aws:lambda:us-west-2:account-id:function:ThumbNail”). AWS Lambda also allows you to specify only the account ID qualifier (for example, “account-id:Thumbnail”). Note that the length constraint applies only to the ARN. If you specify only the function name, it is limited to 64 character in length.

Return type `dict`

Returns

Response Syntax

```
{
    'Policy': 'string'
}
```

Response Structure

- (*dict*) –
 - **Policy** (*string*) –

The access policy associated with the specified function. The response returns the same as a string using “” as an escape character in the JSON.

get_waiter (*waiter_name*)

invoke (***kwargs*)

Invokes a specified Lambda function.

This operation requires permission for the `lambda:InvokeFunction` action.

Request Syntax

```
response = client.invoke(
    FunctionName='string',
    InvocationType='Event'|'RequestResponse'|'DryRun',
    LogType='None'|'Tail',
    ClientContext='string',
```

```
Payload=b'bytes'  
)
```

Parameters

- **FunctionName** (*string*) – [REQUIRED]

The Lambda function name.

You can specify an unqualified function name (for example, “Thumbnail”) or you can specify Amazon Resource Name (ARN) of the function (for example, “arn:aws:lambda:us-west-2:account-id:function:ThumbNail”). AWS Lambda also allows you to specify only the account ID qualifier (for example, “account-id:Thumbnail”). Note that the length constraint applies only to the ARN. If you specify only the function name, it is limited to 64 character in length.

- **InvocationType** (*string*) – By default, the `Invoke` API assumes “RequestResponse” invocation type. You can optionally request asynchronous execution by specifying “Event” as the `InvocationType`. You can also use this parameter to request AWS Lambda to not execute the function but do some verification, such as if the caller is authorized to invoke the function and if the inputs are valid. You request this by specifying “DryRun” as the `InvocationType`. This is useful in a cross-account scenario when you want to verify access to a function without running it.
- **LogType** (*string*) – You can set this optional parameter to “Tail” in the request only if you specify the `InvocationType` parameter with value “RequestResponse”. In this case, AWS Lambda returns the base64-encoded last 4 KB of log data produced by your Lambda function in the `x-amz-log-results` header.
- **ClientContext** (*string*) – Using the `ClientContext` you can pass client-specific information to the Lambda function you are invoking. You can then process the client information in your Lambda function as you choose through the context variable. For an example of a `ClientContext` JSON, go to [PutEvents](#) in the *Amazon Mobile Analytics API Reference and User Guide*.

The `ClientContext` JSON must be base64-encoded.

- **Payload** (*bytes*) – JSON that you want to provide to your Lambda function as input.

Return type dict

Returns

Response Syntax

```
{  
    'StatusCode': 123,  
    'FunctionError': 'string',  
    'LogResult': 'string',  
    'Payload': b'bytes'  
}
```

Response Structure

- (*dict*) –

Upon success, returns an empty response. Otherwise, throws an exception.

- **StatusCode** (*integer*) –

The HTTP status code will be in the 200 range for successful request. For the “RequestResponse” invocation type this status code will be 200. For the “Event” invocation type this status code will be 202. For the “DryRun” invocation type the status code will be 204.

– **FunctionError** (*string*) –

Indicates whether an error occurred while executing the Lambda function. If an error occurred this field will have one of two values; `Handled` or `Unhandled`. `Handled` errors are errors that are reported by the function while the `Unhandled` errors are those detected and reported by AWS Lambda. `Unhandled` errors include out of memory errors and function timeouts. For information about how to report an `Handled` error, see [Programming Model](#).

– **LogResult** (*string*) –

It is the base64-encoded logs for the Lambda function invocation. This is present only if the invocation type is “RequestResponse” and the logs were requested.

– **Payload** (*bytes*) –

It is the JSON representation of the object returned by the Lambda function. In This is present only if the invocation type is “RequestResponse”.

In the event of a function error this field contains a message describing the error. For the `Handled` errors the Lambda function will report this message. For `Unhandled` errors AWS Lambda reports the message.

`invoke_async(**kwargs)`

Warning: This API is deprecated. We recommend you use `Invoke` API (see `Invoke`).

Submits an invocation request to AWS Lambda. Upon receiving the request, Lambda executes the specified function asynchronously. To see the logs generated by the Lambda function execution, see the CloudWatch logs console.

This operation requires permission for the `lambda:InvokeFunction` action.

Request Syntax

```
response = client.invoke_async(
    FunctionName='string',
    InvokeArgs=b'bytes'
)
```

Parameters

- **FunctionName** (*string*) – [REQUIRED]

The Lambda function name.

- **InvokeArgs** (*bytes*) – [REQUIRED]

JSON that you want to provide to your Lambda function as input.

Return type dict

Returns

Response Syntax

```
{
    'Status': 123
}
```

Response Structure

- (*dict*) –

Upon success, it returns empty response. Otherwise, throws an exception.

– **Status** (*integer*) –

It will be 202 upon success.

list_event_source_mappings (***kwargs*)

Returns a list of event source mappings you created using the `CreateEventSourceMapping` (see `CreateEventSourceMapping`), where you identify a stream as an event source. This list does not include Amazon S3 event sources.

For each mapping, the API returns configuration information. You can optionally specify filters to retrieve specific event source mappings.

This operation requires permission for the `lambda:ListEventSourceMappings` action.

Request Syntax

```
response = client.list_event_source_mappings(
    EventSourceArn='string',
    FunctionName='string',
    Marker='string',
    MaxItems=123
)
```

Parameters

- **EventSourceArn** (*string*) – The Amazon Resource Name (ARN) of the Amazon Kinesis stream.
- **FunctionName** (*string*) – The name of the Lambda function.

You can specify an unqualified function name (for example, “Thumbnail”) or you can specify Amazon Resource Name (ARN) of the function (for example, “arn:aws:lambda:us-west-2:account-id:function:ThumbNail”). AWS Lambda also allows you to specify only the account ID qualifier (for example, “account-id:Thumbnail”). Note that the length constraint applies only to the ARN. If you specify only the function name, it is limited to 64 character in length.

- **Marker** (*string*) – Optional string. An opaque pagination token returned from a previous `ListEventSourceMappings` operation. If present, specifies to continue the list from where the returning call left off.
- **MaxItems** (*integer*) – Optional integer. Specifies the maximum number of event sources to return in response. This value must be greater than 0.

Return type dict

Returns

Response Syntax

```
{
  'NextMarker': 'string',
  'EventSourceMappings': [
    {
      'UUID': 'string',
      'BatchSize': 123,
      'EventSourceArn': 'string',
      'FunctionArn': 'string',
      'LastModified': datetime(2015, 1, 1),
      'LastProcessingResult': 'string',
      'State': 'string',
      'StateTransitionReason': 'string'
    },
  ],
}
```


Response Structure

- (*dict*) –

Contains a list of event sources (see `API_EventSourceMappingConfiguration`)

- **NextMarker** (*string*) –

A string, present if there are more event source mappings.

- **EventSourceMappings** (*list*) –

An array of `EventSourceMappingConfiguration` objects.

- * (*dict*) –

Describes mapping between an Amazon Kinesis stream and a Lambda function.

- **UUID** (*string*) –

The AWS Lambda assigned opaque identifier for the mapping.

- **BatchSize** (*integer*) –

The largest number of records that AWS Lambda will retrieve from your event source at the time of invoking your function. Your function receives an event with all the retrieved records.

- **EventSourceArn** (*string*) –

The Amazon Resource Name (ARN) of the Amazon Kinesis stream that is the source of events.

- **FunctionArn** (*string*) –

The Lambda function to invoke when AWS Lambda detects an event on the stream.

- **LastModified** (*datetime*) –

The UTC time string indicating the last time the event mapping was updated.

- **LastProcessingResult** (*string*) –

The result of the last AWS Lambda invocation of your Lambda function.

- **State** (*string*) –

The state of the event source mapping. It can be “Creating”, “Enabled”, “Disabled”, “Enabling”, “Disabling”, “Updating”, or “Deleting”.

- **StateTransitionReason** (*string*) –

The reason the event source mapping is in its current state. It is either user-requested or an AWS Lambda-initiated state transition.

list_functions (***kwargs*)

Returns a list of your Lambda functions. For each function, the response includes the function configuration information. You must use `GetFunction` to retrieve the code for your function.

This operation requires permission for the `lambda:ListFunctions` action.

Request Syntax

```
response = client.list_functions(
    Marker='string',
```

```
MaxItems=123
)
```

Parameters

- **Marker** (*string*) – Optional string. An opaque pagination token returned from a previous `ListFunctions` operation. If present, indicates where to continue the listing.
- **MaxItems** (*integer*) – Optional integer. Specifies the maximum number of AWS Lambda functions to return in response. This parameter value must be greater than 0.

Return type dict**Returns****Response Syntax**

```
{
  'NextMarker': 'string',
  'Functions': [
    {
      'FunctionName': 'string',
      'FunctionArn': 'string',
      'Runtime': 'nodejs'|'java8',
      'Role': 'string',
      'Handler': 'string',
      'CodeSize': 123,
      'Description': 'string',
      'Timeout': 123,
      'MemorySize': 123,
      'LastModified': 'string'
    },
  ]
}
```

Response Structure

- (*dict*) –

Contains a list of AWS Lambda function configurations (see `FunctionConfiguration`).

- **NextMarker** (*string*) –

A string, present if there are more functions.

- **Functions** (*list*) –

A list of Lambda functions.

- * (*dict*) –

A complex type that describes function metadata.

- **FunctionName** (*string*) –

The name of the function.

- **FunctionArn** (*string*) –

The Amazon Resource Name (ARN) assigned to the function.

- **Runtime** (*string*) –

The runtime environment for the Lambda function.

- **Role** (*string*) –

The Amazon Resource Name (ARN) of the IAM role that Lambda assumes when it executes your function to access any

other Amazon Web Services (AWS) resources.

- **Handler** (*string*) –

The function Lambda calls to begin executing your function.

- **CodeSize** (*integer*) –

The size, in bytes, of the function .zip file you uploaded.

- **Description** (*string*) –

The user-provided description.

- **Timeout** (*integer*) –

The function execution time at which Lambda should terminate the function. Because the execution time has cost implications, we recommend you set this value based on your expected execution time. The default is 3 seconds.

- **MemorySize** (*integer*) –

The memory size, in MB, you configured for the function. Must be a multiple of 64 MB.

- **LastModified** (*string*) –

The timestamp of the last time you updated the function.

remove_permission (***kwargs*)

You can remove individual permissions from an access policy associated with a Lambda function by providing a Statement ID.

Note that removal of a permission will cause an active event source to lose permission to the function.

You need permission for the `lambda:RemovePermission` action.

Request Syntax

```
response = client.remove_permission(
    FunctionName='string',
    StatementId='string'
)
```

Parameters

- **FunctionName** (*string*) – **[REQUIRED]**

Lambda function whose access policy you want to remove a permission from.

You can specify an unqualified function name (for example, “Thumbnail”) or you can specify Amazon Resource Name (ARN) of the function (for example, “arn:aws:lambda:us-west-2:account-id:function:ThumbNail”). AWS Lambda also allows you to specify only the account ID qualifier (for example, “account-id:Thumbnail”). Note that the length constraint applies only to the ARN. If you specify only the function name, it is limited to 64 character in length.

- **StatementId** (*string*) – **[REQUIRED]**

Statement ID of the permission to remove.

Returns None

update_event_source_mapping (***kwargs*)

You can update an event source mapping. This is useful if you want to change the parameters of the existing mapping without losing your position in the stream. You can change which function will receive the stream records, but to change the stream itself, you must create a new mapping.

This operation requires permission for the `lambda:UpdateEventSourceMapping` action.

Request Syntax

```
response = client.update_event_source_mapping(
    UUID='string',
    FunctionName='string',
    Enabled=True|False,
    BatchSize=123
)
```

Parameters

- **UUID** (*string*) – [REQUIRED]

The event source mapping identifier.

- **FunctionName** (*string*) – The Lambda function to which you want the stream records sent.

You can specify an unqualified function name (for example, “Thumbnail”) or you can specify Amazon Resource Name (ARN) of the function (for example, “arn:aws:lambda:us-west-2:account-id:function:ThumbNail”). AWS Lambda also allows you to specify only the account ID qualifier (for example, “account-id:Thumbnail”). Note that the length constraint applies only to the ARN. If you specify only the function name, it is limited to 64 character in length.

- **Enabled** (*boolean*) – Specifies whether AWS Lambda should actively poll the stream or not. If disabled, AWS Lambda will not poll the stream.
- **BatchSize** (*integer*) – The maximum number of stream records that can be sent to your Lambda function for a single invocation.

Return type dict

Returns

Response Syntax

```
{
    'UUID': 'string',
    'BatchSize': 123,
    'EventSourceArn': 'string',
    'FunctionArn': 'string',
    'LastModified': datetime(2015, 1, 1),
    'LastProcessingResult': 'string',
    'State': 'string',
    'StateTransitionReason': 'string'
}
```

Response Structure

- (*dict*) –

Describes mapping between an Amazon Kinesis stream and a Lambda function.

- **UUID** (*string*) –

The AWS Lambda assigned opaque identifier for the mapping.

- **BatchSize** (*integer*) –

The largest number of records that AWS Lambda will retrieve from your event source at the time of invoking your function. Your function receives an event with all the retrieved records.

- **EventSourceArn** (*string*) –

The Amazon Resource Name (ARN) of the Amazon Kinesis stream that is the source of events.

- **FunctionArn** (*string*) –
The Lambda function to invoke when AWS Lambda detects an event on the stream.
- **LastModified** (*datetime*) –
The UTC time string indicating the last time the event mapping was updated.
- **LastProcessingResult** (*string*) –
The result of the last AWS Lambda invocation of your Lambda function.
- **State** (*string*) –
The state of the event source mapping. It can be “Creating”, “Enabled”, “Disabled”, “Enabling”, “Disabling”, “Updating”, or “Deleting”.
- **StateTransitionReason** (*string*) –
The reason the event source mapping is in its current state. It is either user-requested or an AWS Lambda-initiated state transition.

update_function_code (***kwargs*)

Updates the code for the specified Lambda function. This operation must only be used on an existing Lambda function and cannot be used to update the function configuration.

This operation requires permission for the `lambda:UpdateFunctionCode` action.

Request Syntax

```
response = client.update_function_code(
    FunctionName='string',
    ZipFile=b'bytes',
    S3Bucket='string',
    S3Key='string',
    S3ObjectVersion='string'
)
```

Parameters

- **FunctionName** (*string*) – **[REQUIRED]**
The existing Lambda function name whose code you want to replace.

You can specify an unqualified function name (for example, “Thumbnail”) or you can specify Amazon Resource Name (ARN) of the function (for example, “arn:aws:lambda:us-west-2:account-id:function:ThumbNail”). AWS Lambda also allows you to specify only the account ID qualifier (for example, “account-id:Thumbnail”). Note that the length constraint applies only to the ARN. If you specify only the function name, it is limited to 64 character in length.
- **ZipFile** (*bytes*) – Based64-encoded .zip file containing your packaged source code.
- **S3Bucket** (*string*) – Amazon S3 bucket name where the .zip file containing your deployment package is stored. This bucket must reside in the same AWS region where you are creating the Lambda function.
- **S3Key** (*string*) – The Amazon S3 object (the deployment package) key name you want to upload.
- **S3ObjectVersion** (*string*) – The Amazon S3 object (the deployment package) version you want to upload.

Return type dict

Returns

Response Syntax

```
{
    'FunctionName': 'string',
    'FunctionArn': 'string',
    'Runtime': 'nodejs'|'java8',
    'Role': 'string',
    'Handler': 'string',
    'CodeSize': 123,
    'Description': 'string',
    'Timeout': 123,
    'MemorySize': 123,
    'LastModified': 'string'
}
```

Response Structure

- (*dict*) –

A complex type that describes function metadata.

- **FunctionName** (*string*) –

The name of the function.

- **FunctionArn** (*string*) –

The Amazon Resource Name (ARN) assigned to the function.

- **Runtime** (*string*) –

The runtime environment for the Lambda function.

- **Role** (*string*) –

The Amazon Resource Name (ARN) of the IAM role that Lambda assumes when it executes your function to access any other Amazon Web Services (AWS) resources.

- **Handler** (*string*) –

The function Lambda calls to begin executing your function.

- **CodeSize** (*integer*) –

The size, in bytes, of the function .zip file you uploaded.

- **Description** (*string*) –

The user-provided description.

- **Timeout** (*integer*) –

The function execution time at which Lambda should terminate the function. Because the execution time has cost implications, we recommend you set this value based on your expected execution time. The default is 3 seconds.

- **MemorySize** (*integer*) –

The memory size, in MB, you configured for the function. Must be a multiple of 64 MB.

- **LastModified** (*string*) –

The timestamp of the last time you updated the function.

update_function_configuration (***kwargs*)

Updates the configuration parameters for the specified Lambda function by using the values provided in the request. You provide only the parameters you want to change. This operation must only be used on an existing Lambda function and cannot be used to update the function's code.

This operation requires permission for the `lambda:UpdateFunctionConfiguration` action.

Request Syntax

```
response = client.update_function_configuration(
    FunctionName='string',
    Role='string',
    Handler='string',
    Description='string',
    Timeout=123,
    MemorySize=123
)
```

Parameters

- **FunctionName** (*string*) – [REQUIRED]

The name of the Lambda function.

You can specify an unqualified function name (for example, “Thumbnail”) or you can specify Amazon Resource Name (ARN) of the function (for example, “arn:aws:lambda:us-west-2:account-id:function:ThumbNail”). AWS Lambda also allows you to specify only the account ID qualifier (for example, “account-id:Thumbnail”). Note that the length constraint applies only to the ARN. If you specify only the function name, it is limited to 64 character in length.

- **Role** (*string*) – The Amazon Resource Name (ARN) of the IAM role that Lambda will assume when it executes your function.
- **Handler** (*string*) – The function that Lambda calls to begin executing your function. For Node.js, it is the *module-name.export* value in your function.
- **Description** (*string*) – A short user-defined function description. AWS Lambda does not use this value. Assign a meaningful description as you see fit.
- **Timeout** (*integer*) – The function execution time at which AWS Lambda should terminate the function. Because the execution time has cost implications, we recommend you set this value based on your expected execution time. The default is 3 seconds.
- **MemorySize** (*integer*) – The amount of memory, in MB, your Lambda function is given. AWS Lambda uses this memory size to infer the amount of CPU allocated to your function. Your function use-case determines your CPU and memory requirements. For example, a database operation might need less memory compared to an image processing function. The default value is 128 MB. The value must be a multiple of 64 MB.

Return type dict

Returns

Response Syntax

```
{
    'FunctionName': 'string',
    'FunctionArn': 'string',
    'Runtime': 'nodejs'|'java8',
    'Role': 'string',
    'Handler': 'string',
    'CodeSize': 123,
    'Description': 'string',
    'Timeout': 123,
    'MemorySize': 123,
    'LastModified': 'string'
}
```

Response Structure

- (*dict*) –

A complex type that describes function metadata.

- **FunctionName** (*string*) –

The name of the function.

- **FunctionArn** (*string*) –

The Amazon Resource Name (ARN) assigned to the function.

- **Runtime** (*string*) –

The runtime environment for the Lambda function.

- **Role** (*string*) –

The Amazon Resource Name (ARN) of the IAM role that Lambda assumes when it executes your function to access any other Amazon Web Services (AWS) resources.

- **Handler** (*string*) –

The function Lambda calls to begin executing your function.

- **CodeSize** (*integer*) –

The size, in bytes, of the function .zip file you uploaded.

- **Description** (*string*) –

The user-provided description.

- **Timeout** (*integer*) –

The function execution time at which Lambda should terminate the function. Because the execution time has cost implications, we recommend you set this value based on your expected execution time. The default is 3 seconds.

- **MemorySize** (*integer*) –

The memory size, in MB, you configured for the function. Must be a multiple of 64 MB.

- **LastModified** (*string*) –

The timestamp of the last time you updated the function.

Paginators

The available paginators are:

- `Lambda.Paginator.ListEventSourceMappings`
- `Lambda.Paginator.ListFunctions`

`class Lambda.Paginator.ListEventSourceMappings`

```
paginator = client.get_paginator('list_event_source_mappings')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Lambda.Client.list_event_source_mappings()`.

Request Syntax


```

response_iterator = paginator.paginate(
    EventSourceArn='string',
    FunctionName='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)

```

Parameters

- **EventSourceArn** (*string*) – The Amazon Resource Name (ARN) of the Amazon Kinesis stream.
- **FunctionName** (*string*) – The name of the Lambda function.

You can specify an unqualified function name (for example, “Thumbnail”) or you can specify Amazon Resource Name (ARN) of the function (for example, “arn:aws:lambda:us-west-2:account-id:function:ThumbNail”). AWS Lambda also allows you to specify only the account ID qualifier (for example, “account-id:Thumbnail”). Note that the length constraint applies only to the ARN. If you specify only the function name, it is limited to 64 character in length.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```

{
    'EventSourceMappings': [
        {
            'UUID': 'string',
            'BatchSize': 123,
            'EventSourceArn': 'string',
            'FunctionArn': 'string',
            'LastModified': datetime(2015, 1, 1),
            'LastProcessingResult': 'string',
            'State': 'string',
            'StateTransitionReason': 'string'
        },
    ],
    'NextToken': 'string'
}

```

Response Structure

- (*dict*) –

Contains a list of event sources (see `API_EventSourceMappingConfiguration`)

– **EventSourceMappings** (*list*) –

An array of `EventSourceMappingConfiguration` objects.

* (*dict*) –

Describes mapping between an Amazon Kinesis stream and a Lambda function.

· **UUID** (*string*) –

The AWS Lambda assigned opaque identifier for the mapping.

· **BatchSize** (*integer*) –

The largest number of records that AWS Lambda will retrieve from your event source at the time of invoking your function. Your function receives an event with all the retrieved records.

· **EventSourceArn** (*string*) –

The Amazon Resource Name (ARN) of the Amazon Kinesis stream that is the source of events.

· **FunctionArn** (*string*) –

The Lambda function to invoke when AWS Lambda detects an event on the stream.

· **LastModified** (*datetime*) –

The UTC time string indicating the last time the event mapping was updated.

· **LastProcessingResult** (*string*) –

The result of the last AWS Lambda invocation of your Lambda function.

· **State** (*string*) –

The state of the event source mapping. It can be “Creating”, “Enabled”, “Disabled”, “Enabling”, “Disabling”, “Updating”, or “Deleting”.

· **StateTransitionReason** (*string*) –

The reason the event source mapping is in its current state. It is either user-requested or an AWS Lambda-initiated state transition.

– **NextToken** (*string*) –

A token to resume pagination.

class `Lambda.Paginator.ListFunctions`

```
paginator = client.get_paginator('list_functions')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Lambda.Client.list_functions()`.

Request Syntax

```
response_iterator = paginator.paginate(
    PaginationConfig={
```

```

        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)

```

Parameters **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```

{
    'Functions': [
        {
            'FunctionName': 'string',
            'FunctionArn': 'string',
            'Runtime': 'nodejs'|'java8',
            'Role': 'string',
            'Handler': 'string',
            'CodeSize': 123,
            'Description': 'string',
            'Timeout': 123,
            'MemorySize': 123,
            'LastModified': 'string'
        },
    ],
    'NextToken': 'string'
}

```

Response Structure

- (*dict*) –

Contains a list of AWS Lambda function configurations (see `FunctionConfiguration`).

- **Functions** (*list*) –

A list of Lambda functions.

- * (*dict*) –

A complex type that describes function metadata.

- **FunctionName** (*string*) –

The name of the function.

- **FunctionArn** (*string*) –

The Amazon Resource Name (ARN) assigned to the function.

- **Runtime** (*string*) –
The runtime environment for the Lambda function.
- **Role** (*string*) –
The Amazon Resource Name (ARN) of the IAM role that Lambda assumes when it executes your function to access any other Amazon Web Services (AWS) resources.
- **Handler** (*string*) –
The function Lambda calls to begin executing your function.
- **CodeSize** (*integer*) –
The size, in bytes, of the function .zip file you uploaded.
- **Description** (*string*) –
The user-provided description.
- **Timeout** (*integer*) –
The function execution time at which Lambda should terminate the function. Because the execution time has cost implications, we recommend you set this value based on your expected execution time. The default is 3 seconds.
- **MemorySize** (*integer*) –
The memory size, in MB, you configured for the function. Must be a multiple of 64 MB.
- **LastModified** (*string*) –
The timestamp of the last time you updated the function.
- **NextToken** (*string*) –
A token to resume pagination.

CloudWatchLogs

Table of Contents

- CloudWatchLogs
 - Client
 - Paginators

Client

`class CloudWatchLogs.Client`

A low-level client representing Amazon CloudWatch Logs:

```
import boto3

client = boto3.client('logs')
```

These are the available methods:

- `can_paginate()`
- `cancel_export_task()`
- `create_export_task()`

- `create_log_group()`
- `create_log_stream()`
- `delete_destination()`
- `delete_log_group()`
- `delete_log_stream()`
- `delete_metric_filter()`
- `delete_retention_policy()`
- `delete_subscription_filter()`
- `describe_destinations()`
- `describe_export_tasks()`
- `describe_log_groups()`
- `describe_log_streams()`
- `describe_metric_filters()`
- `describe_subscription_filters()`
- `filter_log_events()`
- `generate_presigned_url()`
- `get_log_events()`
- `get_paginator()`
- `get_waiter()`
- `put_destination()`
- `put_destination_policy()`
- `put_log_events()`
- `put_metric_filter()`
- `put_retention_policy()`
- `put_subscription_filter()`
- `test_metric_filter()`

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

cancel_export_task (***kwargs*)

Cancels an export task if it is in PENDING or RUNNING state.

Request Syntax

```
response = client.cancel_export_task(
    taskId='string'
)
```

Parameters *taskId* (*string*) – [REQUIRED]

Id of the export task to cancel.

Returns None

create_export_task (***kwargs*)

Creates an `ExportTask` which allows you to efficiently export data from a Log Group to your Amazon S3 bucket.

This is an asynchronous call. If all the required information is provided, this API will initiate an export task and respond with the task Id. Once started, `DescribeExportTasks` can be used to get the status of an export task.

Request Syntax

```
response = client.create_export_task(
    taskName='string',
    logGroupName='string',
    logStreamNamePrefix='string',
    from=123,
    to=123,
    destination='string',
    destinationPrefix='string'
)
```

Parameters

- **taskName** (*string*) – The name of the export task.
- **logGroupName** (*string*) – **[REQUIRED]**
The name of the log group to export.
- **logStreamNamePrefix** (*string*) – Will only export log streams that match the provided logStreamNamePrefix. If you don't specify a value, no prefix filter is applied.
- **from** (*integer*) – **[REQUIRED]**
A unix timestamp indicating the start time of the range for the request. Events with a timestamp prior to this time will not be exported.
- **to** (*integer*) – **[REQUIRED]**
A unix timestamp indicating the end time of the range for the request. Events with a timestamp later than this time will not be exported.
- **destination** (*string*) – **[REQUIRED]**
Name of Amazon S3 bucket to which the log data will be exported. **NOTE: Only buckets in the same AWS region are supported**
- **destinationPrefix** (*string*) – Prefix that will be used as the start of Amazon S3 key for every object exported. If not specified, this defaults to 'exportedlogs'.

Return type dict

Returns

Response Syntax

```
{
    'taskId': 'string'
}
```

Response Structure

- (*dict*) –
 - **taskId** (*string*) –
Id of the export task that got created.

create_log_group (***kwargs*)

Creates a new log group with the specified name. The name of the log group must be unique within a region for an AWS account. You can create up to 500 log groups per account.

You must use the following guidelines when naming a log group:

- Log group names can be between 1 and 512 characters long.
- Allowed characters are a-z, A-Z, 0-9, '_' (underscore), '-' (hyphen), '/' (forward slash), and '.' (period).

Request Syntax

```
response = client.create_log_group(
    logGroupName='string'
)
```

Parameters `logGroupName` (*string*) – [REQUIRED]

The name of the log group to create.

Returns None

create_log_stream (***kwargs*)

Creates a new log stream in the specified log group. The name of the log stream must be unique within the log group. There is no limit on the number of log streams that can exist in a log group.

You must use the following guidelines when naming a log stream:

- Log stream names can be between 1 and 512 characters long.
- The ':' colon character is not allowed.

Request Syntax

```
response = client.create_log_stream(
    logGroupName='string',
    logStreamName='string'
)
```

Parameters

- **logGroupName** (*string*) – [REQUIRED]

The name of the log group under which the log stream is to be created.

- **logStreamName** (*string*) – [REQUIRED]

The name of the log stream to create.

Returns None

delete_destination (***kwargs*)

Deletes the destination with the specified name and eventually disables all the subscription filters that publish to it. This will not delete the physical resource encapsulated by the destination.

Request Syntax

```
response = client.delete_destination(
    destinationName='string'
)
```

Parameters `destinationName` (*string*) – [REQUIRED]

The name of destination to delete.

Returns None

delete_log_group (***kwargs*)

Deletes the log group with the specified name and permanently deletes all the archived log events associated with it.

Request Syntax

```
response = client.delete_log_group(
    logGroupName='string'
)
```

Parameters `logGroupName` (*string*) – [REQUIRED]

The name of the log group to delete.

Returns None

delete_log_stream (**kwargs)

Deletes a log stream and permanently deletes all the archived log events associated with it.

Request Syntax

```
response = client.delete_log_stream(
    logGroupName='string',
    logStreamName='string'
)
```

Parameters

- **logGroupName** (*string*) – [REQUIRED]
The name of the log group under which the log stream to delete belongs.
- **logStreamName** (*string*) – [REQUIRED]
The name of the log stream to delete.

Returns None

delete_metric_filter (**kwargs)

Deletes a metric filter associated with the specified log group.

Request Syntax

```
response = client.delete_metric_filter(
    logGroupName='string',
    filterName='string'
)
```

Parameters

- **logGroupName** (*string*) – [REQUIRED]
The name of the log group that is associated with the metric filter to delete.
- **filterName** (*string*) – [REQUIRED]
The name of the metric filter to delete.

Returns None

delete_retention_policy (**kwargs)

Deletes the retention policy of the specified log group. Log events would not expire if they belong to log groups without a retention policy.

Request Syntax

```
response = client.delete_retention_policy(
    logGroupName='string'
)
```

Parameters **logGroupName** (*string*) – [REQUIRED]

The name of the log group that is associated with the retention policy to delete.

Returns None

delete_subscription_filter (**kwargs)

Deletes a subscription filter associated with the specified log group.

Request Syntax


```
response = client.delete_subscription_filter(
    logGroupName='string',
    filterName='string'
)
```

Parameters

- **logGroupName** (*string*) – [REQUIRED]
The name of the log group that is associated with the subscription filter to delete.
- **filterName** (*string*) – [REQUIRED]
The name of the subscription filter to delete.

Returns None

describe_destinations (***kwargs*)

Returns all the destinations that are associated with the AWS account making the request. The list returned in the response is ASCII-sorted by destination name.

By default, this operation returns up to 50 destinations. If there are more destinations to list, the response would contain a `nextToken` value in the response body. You can also limit the number of destinations returned in the response by specifying the `limit` parameter in the request.

Request Syntax

```
response = client.describe_destinations(
    DestinationNamePrefix='string',
    nextToken='string',
    limit=123
)
```

Parameters

- **DestinationNamePrefix** (*string*) – Will only return destinations that match the provided `destinationNamePrefix`. If you don't specify a value, no prefix is applied.
- **nextToken** (*string*) – A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous request. The token expires after 24 hours.
- **limit** (*integer*) – The maximum number of results to return.

Return type dict

Returns

Response Syntax

```
{
    'destinations': [
        {
            'destinationName': 'string',
            'targetArn': 'string',
            'roleArn': 'string',
            'accessPolicy': 'string',
            'arn': 'string',
            'creationTime': 123
        },
    ],
    'nextToken': 'string'
}
```

Response Structure

- (*dict*) –

- **destinations** (*list*) –
 - * (*dict*) –

A cross account destination that is the recipient of subscription log events.

 - **destinationName** (*string*) –

Name of the destination.
 - **targetArn** (*string*) –

ARN of the physical target where the log events will be delivered (eg. ARN of a Kinesis stream).
 - **roleArn** (*string*) –

A role for impersonation for delivering log events to the target.
 - **accessPolicy** (*string*) –

An IAM policy document that governs which AWS accounts can create subscription filters against this destination.
 - **arn** (*string*) –

ARN of this destination.
 - **creationTime** (*integer*) –

A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC specifying when this destination was created.
- **nextToken** (*string*) –

A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous request. The token expires after 24 hours.

describe_export_tasks (***kwargs*)

Returns all the export tasks that are associated with the AWS account making the request. The export tasks can be filtered based on `taskId` or `taskStatus`.

By default, this operation returns up to 50 export tasks that satisfy the specified filters. If there are more export tasks to list, the response would contain a `nextToken` value in the response body. You can also limit the number of export tasks returned in the response by specifying the `limit` parameter in the request.

Request Syntax

```
response = client.describe_export_tasks(  
    taskId='string',  
    statusCode='CANCELLED' | 'COMPLETED' | 'FAILED' | 'PENDING' | 'PENDING_CANCEL' | 'RUNNING',  
    nextToken='string',  
    limit=123  
)
```

Parameters

- **taskId** (*string*) – Export task that matches the specified task Id will be returned. This can result in zero or one export task.
- **statusCode** (*string*) – All export tasks that matches the specified status code will be returned. This can return zero or more export tasks.
- **nextToken** (*string*) – A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous `DescribeExportTasks` request.

- **limit** (*integer*) – The maximum number of items returned in the response. If you don't specify a value, the request would return up to 50 items.

Return type dict

Returns

Response Syntax

```
{
  'exportTasks': [
    {
      'taskId': 'string',
      'taskName': 'string',
      'logGroupName': 'string',
      'from': 123,
      'to': 123,
      'destination': 'string',
      'destinationPrefix': 'string',
      'status': {
        'code': 'CANCELLED' | 'COMPLETED' | 'FAILED' | 'PENDING' | 'PENDING_CANCELLED',
        'message': 'string'
      },
      'executionInfo': {
        'creationTime': 123,
        'completionTime': 123
      }
    },
  ],
  'nextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **exportTasks** (*list*) –

A list of export tasks.

 - * (*dict*) –

Represents an export task.

 - **taskId** (*string*) –

Id of the export task.
 - **taskName** (*string*) –

The name of the export task.
 - **logGroupName** (*string*) –

The name of the log group from which logs data was exported.
 - **from** (*integer*) –

A unix timestamp indicating the start time of the range for the request. Events with a timestamp prior to this time were not exported.
 - **to** (*integer*) –

A unix timestamp indicating the end time of the range for the request. Events with a timestamp later than this time were not exported.
 - **destination** (*string*) –

Name of Amazon S3 bucket to which the log data was exported.

- **destinationPrefix** (*string*) –

Prefix that was used as the start of Amazon S3 key for every object exported.

- **status** (*dict*) –

Status of the export task.

- **code** (*string*) –

Status code of the export task.

- **message** (*string*) –

Status message related to the `code`.

- **executionInfo** (*dict*) –

Execution info about the export task.

- **creationTime** (*integer*) –

A point in time when the export task got created.

- **completionTime** (*integer*) –

A point in time when the export task got completed.

- **nextToken** (*string*) –

A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous request. The token expires after 24 hours.

describe_log_groups (***kwargs*)

Returns all the log groups that are associated with the AWS account making the request. The list returned in the response is ASCII-sorted by log group name.

By default, this operation returns up to 50 log groups. If there are more log groups to list, the response would contain a `nextToken` value in the response body. You can also limit the number of log groups returned in the response by specifying the `limit` parameter in the request.

Request Syntax

```
response = client.describe_log_groups(
    logGroupNamePrefix='string',
    nextToken='string',
    limit=123
)
```

Parameters

- **logGroupNamePrefix** (*string*) – Will only return log groups that match the provided `logGroupNamePrefix`. If you don't specify a value, no prefix filter is applied.
- **nextToken** (*string*) – A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous `DescribeLogGroups` request.
- **limit** (*integer*) – The maximum number of items returned in the response. If you don't specify a value, the request would return up to 50 items.

Return type dict

Returns

Response Syntax

```
{
  'logGroups': [
    {
      'logGroupName': 'string',
      'creationTime': 123,
      'retentionInDays': 123,
      'metricFilterCount': 123,
      'arn': 'string',
      'storedBytes': 123
    },
  ],
  'nextToken': 'string'
}
```

Response Structure

- *(dict)* –
 - **logGroups** (*list*) –

A list of log groups.

 - * *(dict)* –
 - **logGroupName** (*string*) –
 - **creationTime** (*integer*) –

A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.
 - **retentionInDays** (*integer*) –

Specifies the number of days you want to retain log events in the specified log group. Possible values are: 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 3653.
 - **metricFilterCount** (*integer*) –

The number of metric filters associated with the log group.
 - **arn** (*string*) –
 - **storedBytes** (*integer*) –
 - **nextToken** (*string*) –

A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous request. The token expires after 24 hours.

describe_log_streams (***kwargs*)

Returns all the log streams that are associated with the specified log group. The list returned in the response is ASCII-sorted by log stream name.

By default, this operation returns up to 50 log streams. If there are more log streams to list, the response would contain a `nextToken` value in the response body. You can also limit the number of log streams returned in the response by specifying the `limit` parameter in the request. This operation has a limit of five transactions per second, after which transactions are throttled.

Request Syntax

```
response = client.describe_log_streams(
    logGroupName='string',
    logStreamNamePrefix='string',
    orderBy='LogStreamName'|'LastEventTime',
    descending=True|False,
    nextToken='string',
```

```
limit=123
)
```

Parameters

- **logGroupName** (*string*) – [REQUIRED]
The log group name for which log streams are to be listed.
- **logStreamNamePrefix** (*string*) – Will only return log streams that match the provided logStreamNamePrefix. If you don't specify a value, no prefix filter is applied.
- **orderBy** (*string*) – Specifies what to order the returned log streams by. Valid arguments are 'LogStreamName' or 'LastEventTime'. If you don't specify a value, results are ordered by LogStreamName. If 'LastEventTime' is chosen, the request cannot also contain a logStreamNamePrefix.
- **descending** (*boolean*) – If set to true, results are returned in descending order. If you don't specify a value or set it to false, results are returned in ascending order.
- **nextToken** (*string*) – A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous DescribeLogStreams request.
- **limit** (*integer*) – The maximum number of items returned in the response. If you don't specify a value, the request would return up to 50 items.

Return type dict

Returns

Response Syntax

```
{
  'logStreams': [
    {
      'logStreamName': 'string',
      'creationTime': 123,
      'firstEventTimestamp': 123,
      'lastEventTimestamp': 123,
      'lastIngestionTime': 123,
      'uploadSequenceToken': 'string',
      'arn': 'string',
      'storedBytes': 123
    },
  ],
  'nextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **logStreams** (*list*) –
A list of log streams.
 - * (*dict*) –
A log stream is sequence of log events from a single emitter of logs.
 - **logStreamName** (*string*) –
 - **creationTime** (*integer*) –
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.
 - **firstEventTimestamp** (*integer*) –

A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

- **lastEventTimestamp** (*integer*) –

A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

- **lastIngestionTime** (*integer*) –

A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

- **uploadSequenceToken** (*string*) –

A string token used for making PutLogEvents requests. A sequenceToken can only be used once, and PutLogEvents requests must include the sequenceToken obtained from the response of the previous request.

- **arn** (*string*) –
- **storedBytes** (*integer*) –

- **nextToken** (*string*) –

A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous request. The token expires after 24 hours.

describe_metric_filters (***kwargs*)

Returns all the metrics filters associated with the specified log group. The list returned in the response is ASCII-sorted by filter name.

By default, this operation returns up to 50 metric filters. If there are more metric filters to list, the response would contain a nextToken value in the response body. You can also limit the number of metric filters returned in the response by specifying the limit parameter in the request.

Request Syntax

```
response = client.describe_metric_filters(
    logGroupName='string',
    filterNamePrefix='string',
    nextToken='string',
    limit=123
)
```

Parameters

- **logGroupName** (*string*) – [REQUIRED]
The log group name for which metric filters are to be listed.
- **filterNamePrefix** (*string*) – Will only return metric filters that match the provided filterNamePrefix. If you don't specify a value, no prefix filter is applied.
- **nextToken** (*string*) – A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous DescribeMetricFilters request.
- **limit** (*integer*) – The maximum number of items returned in the response. If you don't specify a value, the request would return up to 50 items.

Return type dict

Returns

Response Syntax

```
{
    'metricFilters': [
```

```
{
    'filterName': 'string',
    'filterPattern': 'string',
    'metricTransformations': [
        {
            'metricName': 'string',
            'metricNamespace': 'string',
            'metricValue': 'string'
        },
    ],
    'creationTime': 123
},
'nextToken': 'string'
}
```

Response Structure

- *(dict)* –
 - **metricFilters** (*list*) –
 - * *(dict)* –

Metric filters can be used to express how Amazon CloudWatch Logs would extract metric observations from ingested log events and transform them to metric data in a CloudWatch metric.

- **filterName** (*string*) –

A name for a metric or subscription filter.

- **filterPattern** (*string*) –

A symbolic description of how Amazon CloudWatch Logs should interpret the data in each log event. For example, a log event may contain timestamps, IP addresses, strings, and so on. You use the filter pattern to specify what to look for in the log event message.

- **metricTransformations** (*list*) –

- *(dict)* –

- **metricName** (*string*) –

The name of the CloudWatch metric to which the monitored log information should be published. For example, you may publish to a metric called ErrorCount.

- **metricNamespace** (*string*) –

The destination namespace of the new CloudWatch metric.

- **metricValue** (*string*) –

What to publish to the metric. For example, if you're counting the occurrences of a particular term like "Error", the value will be "1" for each occurrence. If you're counting the bytes transferred the published value will be the value in the log event.

- **creationTime** (*integer*) –

A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

- **nextToken** (*string*) –

A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous request. The

token expires after 24 hours.

describe_subscription_filters (**kwargs)

Returns all the subscription filters associated with the specified log group. The list returned in the response is ASCII-sorted by filter name.

By default, this operation returns up to 50 subscription filters. If there are more subscription filters to list, the response would contain a `nextToken` value in the response body. You can also limit the number of subscription filters returned in the response by specifying the `limit` parameter in the request.

Request Syntax

```
response = client.describe_subscription_filters(
    logGroupName='string',
    filterNamePrefix='string',
    nextToken='string',
    limit=123
)
```

Parameters

- **logGroupName** (*string*) – [REQUIRED]
The log group name for which subscription filters are to be listed.
- **filterNamePrefix** (*string*) – Will only return subscription filters that match the provided filterNamePrefix. If you don't specify a value, no prefix filter is applied.
- **nextToken** (*string*) – A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous request. The token expires after 24 hours.
- **limit** (*integer*) – The maximum number of results to return.

Return type dict

Returns

Response Syntax

```
{
  'subscriptionFilters': [
    {
      'filterName': 'string',
      'logGroupName': 'string',
      'filterPattern': 'string',
      'destinationArn': 'string',
      'roleArn': 'string',
      'creationTime': 123
    },
  ],
  'nextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **subscriptionFilters** (*list*) –
 - * (*dict*) –
 - **filterName** (*string*) –
A name for a metric or subscription filter.
 - **logGroupName** (*string*) –
 - **filterPattern** (*string*) –

A symbolic description of how Amazon CloudWatch Logs should interpret the data in each log event. For example, a log event may contain timestamps, IP addresses, strings, and so on. You use the filter pattern to specify what to look for in the log event message.

- **destinationArn** (*string*) –
- **roleArn** (*string*) –
- **creationTime** (*integer*) –

A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

- **nextToken** (*string*) –

A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous request. The token expires after 24 hours.

filter_log_events (***kwargs*)

Retrieves log events, optionally filtered by a filter pattern from the specified log group. You can provide an optional time range to filter the results on the event `timestamp`. You can limit the streams searched to an explicit list of `logStreamNames`.

By default, this operation returns as much matching log events as can fit in a response size of 1MB, up to 10,000 log events, or all the events found within a time-bounded scan window. If the response includes a `nextToken`, then there is more data to search, and the search can be resumed with a new request providing the `nextToken`. The response will contain a list of `searchedLogStreams` that contains information about which streams were searched in the request and whether they have been searched completely or require further pagination. The `limit` parameter in the request. can be used to specify the maximum number of events to return in a page.

Request Syntax

```
response = client.filter_log_events(  
    logGroupName='string',  
    logStreamNames=[  
        'string',  
    ],  
    startTime=123,  
    endTime=123,  
    filterPattern='string',  
    nextToken='string',  
    limit=123,  
    interleaved=True|False  
)
```

Parameters

- **logGroupName** (*string*) – [REQUIRED]
The name of the log group to query.
- **logStreamNames** (*list*) – Optional list of log stream names within the specified log group to search. Defaults to all the log streams in the log group.
 - (*string*) –
- **startTime** (*integer*) – A unix timestamp indicating the start time of the range for the request. If provided, events with a timestamp prior to this time will not be returned.
- **endTime** (*integer*) – A unix timestamp indicating the end time of the range for the request. If provided, events with a timestamp later than this time will not be returned.

- **filterPattern** (*string*) – A valid CloudWatch Logs filter pattern to use for filtering the response. If not provided, all the events are matched.
- **nextToken** (*string*) – A pagination token obtained from a `FilterLogEvents` response to continue paginating the `FilterLogEvents` results.
- **limit** (*integer*) – The maximum number of events to return in a page of results. Default is 10,000 events.
- **interleaved** (*boolean*) – If provided, the API will make a best effort to provide responses that contain events from multiple log streams within the log group interleaved in a single response. If not provided, all the matched log events in the first log stream will be searched first, then those in the next log stream, etc.

Return type dict

Returns

Response Syntax

```
{
  'events': [
    {
      'logStreamName': 'string',
      'timestamp': 123,
      'message': 'string',
      'ingestionTime': 123,
      'eventId': 'string'
    },
  ],
  'searchedLogStreams': [
    {
      'logStreamName': 'string',
      'searchedCompletely': True|False
    },
  ],
  'nextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **events** (*list*) –

A list of `FilteredLogEvent` objects representing the matched events from the request.

- * (*dict*) –

Represents a matched event from a `FilterLogEvents` request.

- **logStreamName** (*string*) –
The name of the log stream this event belongs to.
- **timestamp** (*integer*) –
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.
- **message** (*string*) –
The data contained in the log event.
- **ingestionTime** (*integer*) –
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.
- **eventId** (*string*) –

A unique identifier for this event.

– **searchedLogStreams** (*list*) –

A list of `SearchedLogStream` objects indicating which log streams have been searched in this request and whether each has been searched completely or still has more to be paginated.

* (*dict*) –

An object indicating the search status of a log stream in a `FilterLogEvents` request.

· **logStreamName** (*string*) –

The name of the log stream.

· **searchedCompletely** (*boolean*) –

Indicates whether all the events in this log stream were searched or more data exists to search by paginating further.

– **nextToken** (*string*) –

A pagination token obtained from a `FilterLogEvents` response to continue paginating the `FilterLogEvents` results.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_log_events (***kwargs*)

Retrieves log events from the specified log stream. You can provide an optional time range to filter the results on the event `timestamp`.

By default, this operation returns as much log events as can fit in a response size of 1MB, up to 10,000 log events. The response will always include a `nextForwardToken` and a `nextBackwardToken` in the response body. You can use any of these tokens in subsequent `GetLogEvents` requests to paginate through events in either forward or backward direction. You can also limit the number of log events returned in the response by specifying the `limit` parameter in the request.

Request Syntax

```
response = client.get_log_events(
    logGroupName='string',
    logStreamName='string',
    startTime=123,
    endTime=123,
    nextToken='string',
    limit=123,
    startFromHead=True|False
)
```

Parameters

- **logGroupName** (*string*) – [REQUIRED]

The name of the log group to query.

- **logStreamName** (*string*) – [REQUIRED]

The name of the log stream to query.

- **startTime** (*integer*) – A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.
- **endTime** (*integer*) – A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.
- **nextToken** (*string*) – A string token used for pagination that points to the next page of results. It must be a value obtained from the `nextForwardToken` or `nextBackwardToken` fields in the response of the previous `GetLogEvents` request.
- **limit** (*integer*) – The maximum number of log events returned in the response. If you don't specify a value, the request would return as many log events as can fit in a response size of 1MB, up to 10,000 log events.
- **startFromHead** (*boolean*) – If set to true, the earliest log events would be returned first. The default is false (the latest log events are returned first).

Return type dict

Returns

Response Syntax

```
{
  'events': [
    {
      'timestamp': 123,
      'message': 'string',
      'ingestionTime': 123
    },
  ],
  'nextForwardToken': 'string',
  'nextBackwardToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **events** (*list*) –
 - * (*dict*) –
 - **timestamp** (*integer*) –

A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.
 - **message** (*string*) –
 - **ingestionTime** (*integer*) –

A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.
 - **nextForwardToken** (*string*) –

A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous request. The token expires after 24 hours.
 - **nextBackwardToken** (*string*) –

A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous request. The token expires after 24 hours.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type L{botocore.paginate.Paginator}

Returns A paginator object.

get_waiter (*waiter_name*)

put_destination (***kwargs*)

Creates or updates a `Destination`. A destination encapsulates a physical resource (such as a Kinesis stream) and allows you to subscribe to a real-time stream of log events of a different account, ingested through `PutLogEvents` requests. Currently, the only supported physical resource is a Amazon Kinesis stream belonging to the same account as the destination.

A destination controls what is written to its Amazon Kinesis stream through an access policy. By default, `PutDestination` does not set any access policy with the destination, which means a cross-account user will not be able to call `PutSubscriptionFilter` against this destination. To enable that, the destination owner must call `PutDestinationPolicy` after `PutDestination`.

Request Syntax

```
response = client.put_destination(
    destinationName='string',
    targetArn='string',
    roleArn='string'
)
```

Parameters

- **destinationName** (*string*) – [REQUIRED]
A name for the destination.
- **targetArn** (*string*) – [REQUIRED]
The ARN of an Amazon Kinesis stream to deliver matching log events to.
- **roleArn** (*string*) – [REQUIRED]
The ARN of an IAM role that grants Amazon CloudWatch Logs permissions to do Amazon Kinesis `PutRecord` requests on the destination stream.

Return type dict

Returns

Response Syntax

```
{
  'destination': {
    'destinationName': 'string',
    'targetArn': 'string',
    'roleArn': 'string',
    'accessPolicy': 'string',
    'arn': 'string',
    'creationTime': 123
  }
}
```

Response Structure

- *(dict)* –
 - **destination** *(dict)* –
 - A cross account destination that is the recipient of subscription log events.
 - * **destinationName** *(string)* –
 - Name of the destination.
 - * **targetArn** *(string)* –
 - ARN of the physical target where the log events will be delivered (eg. ARN of a Kinesis stream).
 - * **roleArn** *(string)* –
 - A role for impersonation for delivering log events to the target.
 - * **accessPolicy** *(string)* –
 - An IAM policy document that governs which AWS accounts can create subscription filters against this destination.
 - * **arn** *(string)* –
 - ARN of this destination.
 - * **creationTime** *(integer)* –
 - A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC specifying when this destination was created.

put_destination_policy (***kwargs*)

Creates or updates an access policy associated with an existing `Destination`. An access policy is an [IAM policy document](#) that is used to authorize claims to register a subscription filter against a given destination.

Request Syntax

```
response = client.put_destination_policy(
    destinationName='string',
    accessPolicy='string'
)
```

Parameters

- **destinationName** *(string)* – **[REQUIRED]**
 - A name for an existing destination.
- **accessPolicy** *(string)* – **[REQUIRED]**
 - An IAM policy document that authorizes cross-account users to deliver their log events to associated destination.

Returns None**put_log_events** (***kwargs*)

Uploads a batch of log events to the specified log stream.

Every `PutLogEvents` request must include the `sequenceToken` obtained from the response of the previous request. An upload in a newly created log stream does not require a `sequenceToken`.

The batch of events must satisfy the following constraints:

- The maximum batch size is 1,048,576 bytes, and this size is calculated as the sum of all event messages in UTF-8, plus 26 bytes for each log event.
- None of the log events in the batch can be more than 2 hours in the future.
- None of the log events in the batch can be older than 14 days or the retention period of the log group.

- The log events in the batch must be in chronological ordered by their `timestamp`.
- The maximum number of log events in a batch is 10,000.

Request Syntax

```
response = client.put_log_events(
    logGroupName='string',
    logStreamName='string',
    logEvents=[
        {
            'timestamp': 123,
            'message': 'string'
        },
    ],
    sequenceToken='string'
)
```

Parameters

- **logGroupName** (*string*) – [REQUIRED]
The name of the log group to put log events to.
- **logStreamName** (*string*) – [REQUIRED]
The name of the log stream to put log events to.
- **logEvents** (*list*) – [REQUIRED]
A list of log events belonging to a log stream.
 - (*dict*) –
A log event is a record of some activity that was recorded by the application or resource being monitored. The log event record that Amazon CloudWatch Logs understands contains two properties: the timestamp of when the event occurred, and the raw event message.
 - * **timestamp** (*integer*) – [REQUIRED]
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.
 - * **message** (*string*) – [REQUIRED]
- **sequenceToken** (*string*) – A string token that must be obtained from the response of the previous `PutLogEvents` request.

Return type dict

Returns

Response Syntax

```
{
    'nextSequenceToken': 'string',
    'rejectedLogEventsInfo': {
        'tooNewLogEventStartIndex': 123,
        'tooOldLogEventEndIndex': 123,
        'expiredLogEventEndIndex': 123
    }
}
```

Response Structure

- (*dict*) –
 - **nextSequenceToken** (*string*) –
A string token used for making `PutLogEvents` requests. A `sequenceToken` can only be used once, and `PutLogEvents` requests

must include the `sequenceToken` obtained from the response of the previous request.

- **rejectedLogEventsInfo** (*dict*) –
 - * **tooNewLogEventStartIndex** (*integer*) –
 - * **tooOldLogEventEndIndex** (*integer*) –
 - * **expiredLogEventEndIndex** (*integer*) –

put_metric_filter (***kwargs*)

Creates or updates a metric filter and associates it with the specified log group. Metric filters allow you to configure rules to extract metric data from log events ingested through `PutLogEvents` requests.

The maximum number of metric filters that can be associated with a log group is 100.

Request Syntax

```
response = client.put_metric_filter(
    logGroupName='string',
    filterName='string',
    filterPattern='string',
    metricTransformations=[
        {
            'metricName': 'string',
            'metricNamespace': 'string',
            'metricValue': 'string'
        },
    ]
)
```

Parameters

- **logGroupName** (*string*) – [REQUIRED]
The name of the log group to associate the metric filter with.
- **filterName** (*string*) – [REQUIRED]
A name for the metric filter.
- **filterPattern** (*string*) – [REQUIRED]
A valid CloudWatch Logs filter pattern for extracting metric data out of ingested log events.
- **metricTransformations** (*list*) – [REQUIRED]
A collection of information needed to define how metric data gets emitted.
 - (*dict*) –
 - * **metricName** (*string*) – [REQUIRED]
The name of the CloudWatch metric to which the monitored log information should be published. For example, you may publish to a metric called `ErrorCount`.
 - * **metricNamespace** (*string*) – [REQUIRED]
The destination namespace of the new CloudWatch metric.
 - * **metricValue** (*string*) – [REQUIRED]
What to publish to the metric. For example, if you’re counting the occurrences of a particular term like “Error”, the value will be “1” for each occurrence. If you’re counting the bytes transferred the published value will be the value in the log event.

Returns None

put_retention_policy (***kwargs*)

Sets the retention of the specified log group. A retention policy allows you to configure the number of days you want to retain log events in the specified log group.

Request Syntax

```
response = client.put_retention_policy(
    logGroupName='string',
    retentionInDays=123
)
```

Parameters

- **logGroupName** (*string*) – [REQUIRED]
The name of the log group to associate the retention policy with.
- **retentionInDays** (*integer*) – [REQUIRED]
Specifies the number of days you want to retain log events in the specified log group. Possible values are: 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 3653.

Returns None

put_subscription_filter (***kwargs*)

Creates or updates a subscription filter and associates it with the specified log group. Subscription filters allow you to subscribe to a real-time stream of log events ingested through `PutLogEvents` requests and have them delivered to a specific destination. Currently, the supported destinations are:

- A Amazon Kinesis stream belonging to the same account as the subscription filter, for same-account delivery.
- A logical destination (used via an ARN of `Destination`) belonging to a different account, for cross-account delivery.

Currently there can only be one subscription filter associated with a log group.

Request Syntax

```
response = client.put_subscription_filter(
    logGroupName='string',
    filterName='string',
    filterPattern='string',
    destinationArn='string',
    roleArn='string'
)
```

Parameters

- **logGroupName** (*string*) – [REQUIRED]
The name of the log group to associate the subscription filter with.
- **filterName** (*string*) – [REQUIRED]
A name for the subscription filter.
- **filterPattern** (*string*) – [REQUIRED]
A valid CloudWatch Logs filter pattern for subscribing to a filtered stream of log events.
- **destinationArn** (*string*) – [REQUIRED]
The ARN of the destination to deliver matching log events to. Currently, the supported destinations are:
 - A Amazon Kinesis stream belonging to the same account as the subscription filter, for same-account delivery.

- A logical destination (used via an ARN of `Destination`) belonging to a different account, for cross-account delivery.
- **roleArn** (*string*) – The ARN of an IAM role that grants Amazon CloudWatch Logs permissions to deliver ingested log events to the destination stream. You don't need to provide the ARN when you are working with a logical destination (used via an ARN of `Destination`) for cross-account delivery.

Returns None

test_metric_filter (***kwargs*)

Tests the filter pattern of a metric filter against a sample of log event messages. You can use this operation to validate the correctness of a metric filter pattern.

Request Syntax

```
response = client.test_metric_filter(
    filterPattern='string',
    logEventMessages=[
        'string',
    ]
)
```

Parameters

- **filterPattern** (*string*) – [REQUIRED]

A symbolic description of how Amazon CloudWatch Logs should interpret the data in each log event. For example, a log event may contain timestamps, IP addresses, strings, and so on. You use the filter pattern to specify what to look for in the log event message.

- **logEventMessages** (*list*) – [REQUIRED]

A list of log event messages to test.

– (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'matches': [
    {
      'eventNumber': 123,
      'eventMessage': 'string',
      'extractedValues': {
        'string': 'string'
      }
    },
  ],
}
```

Response Structure

- (*dict*) –
 - **matches** (*list*) –
 - * (*dict*) –
 - **eventNumber** (*integer*) –
 - **eventMessage** (*string*) –
 - **extractedValues** (*dict*) –
 - (*string*) –
 - (*string*) –

Paginators

The available paginators are:

- `CloudWatchLogs.Paginator.DescribeDestinations`
- `CloudWatchLogs.Paginator.DescribeLogGroups`
- `CloudWatchLogs.Paginator.DescribeLogStreams`
- `CloudWatchLogs.Paginator.DescribeMetricFilters`
- `CloudWatchLogs.Paginator.DescribeSubscriptionFilters`
- `CloudWatchLogs.Paginator.FilterLogEvents`

class `CloudWatchLogs.Paginator.DescribeDestinations`

```
paginator = client.get_paginator('describe_destinations')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `CloudWatchLogs.Client.describe_destinations()`.

Request Syntax

```
response_iterator = paginator.paginate(
    DestinationNamePrefix='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **DestinationNamePrefix** (*string*) – Will only return destinations that match the provided destinationNamePrefix. If you don't specify a value, no prefix is applied.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'destinations': [
    {
      'destinationName': 'string',
      'targetArn': 'string',
      'roleArn': 'string',
      'accessPolicy': 'string',
      'arn': 'string',
      'creationTime': 123
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- *(dict)* –
 - **destinations** *(list)* –
 - * *(dict)* –

A cross account destination that is the recipient of subscription log events.

 - **destinationName** *(string)* –

Name of the destination.
 - **targetArn** *(string)* –

ARN of the physical target where the log events will be delivered (eg. ARN of a Kinesis stream).
 - **roleArn** *(string)* –

A role for impersonation for delivering log events to the target.
 - **accessPolicy** *(string)* –

An IAM policy document that governs which AWS accounts can create subscription filters against this destination.
 - **arn** *(string)* –

ARN of this destination.
 - **creationTime** *(integer)* –

A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC specifying when this destination was created.
 - **NextToken** *(string)* –

A token to resume pagination.

class CloudWatchLogs.Paginator.**DescribeLogGroups**

```
paginator = client.get_paginator('describe_log_groups')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `CloudWatchLogs.Client.describe_log_groups()`.

Request Syntax

```
response_iterator = paginator.paginate(
    logGroupNamePrefix='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **logGroupNamePrefix** (*string*) – Will only return log groups that match the provided logGroupNamePrefix. If you don't specify a value, no prefix filter is applied.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'logGroups': [
    {
      'logGroupName': 'string',
      'creationTime': 123,
      'retentionInDays': 123,
      'metricFilterCount': 123,
      'arn': 'string',
      'storedBytes': 123
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **logGroups** (*list*) –
A list of log groups.
 - * (*dict*) –
 - **logGroupName** (*string*) –
 - **creationTime** (*integer*) –
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

- **retentionInDays** (*integer*) –
Specifies the number of days you want to retain log events in the specified log group. Possible values are: 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 3653.
- **metricFilterCount** (*integer*) –
The number of metric filters associated with the log group.
- **arn** (*string*) –
- **storedBytes** (*integer*) –
- **NextToken** (*string*) –
A token to resume pagination.

class CloudWatchLogs.Paginator.**DescribeLogStreams**

```
paginator = client.get_paginator('describe_log_streams')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `CloudWatchLogs.Client.describe_log_streams()`.

Request Syntax

```
response_iterator = paginator.paginate(
    logGroupName='string',
    logStreamNamePrefix='string',
    orderBy='LogStreamName'|'LastEventTime',
    descending=True|False,
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **logGroupName** (*string*) – [REQUIRED]
The log group name for which log streams are to be listed.
- **logStreamNamePrefix** (*string*) – Will only return log streams that match the provided logStreamNamePrefix. If you don't specify a value, no prefix filter is applied.
- **orderBy** (*string*) – Specifies what to order the returned log streams by. Valid arguments are 'LogStreamName' or 'LastEventTime'. If you don't specify a value, results are ordered by LogStreamName. If 'LastEventTime' is chosen, the request cannot also contain a logStreamNamePrefix.
- **descending** (*boolean*) – If set to true, results are returned in descending order. If you don't specify a value or set it to false, results are returned in ascending order.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a NextToken will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –
The size of each page.
- **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'logStreams': [
    {
      'logStreamName': 'string',
      'creationTime': 123,
      'firstEventTimestamp': 123,
      'lastEventTimestamp': 123,
      'lastIngestionTime': 123,
      'uploadSequenceToken': 'string',
      'arn': 'string',
      'storedBytes': 123
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **logStreams** (*list*) –
A list of log streams.
 - * (*dict*) –
A log stream is sequence of log events from a single emitter of logs.
 - **logStreamName** (*string*) –
 - **creationTime** (*integer*) –
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.
 - **firstEventTimestamp** (*integer*) –
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.
 - **lastEventTimestamp** (*integer*) –
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.
 - **lastIngestionTime** (*integer*) –
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.
 - **uploadSequenceToken** (*string*) –
A string token used for making `PutLogEvents` requests. A `sequenceToken` can only be used once, and `PutLogEvents` requests must include the `sequenceToken` obtained from the response of the previous request.
 - **arn** (*string*) –

- **storedBytes** (*integer*) –
- **NextToken** (*string*) –
- A token to resume pagination.

class CloudWatchLogs.Paginator.**DescribeMetricFilters**

```
paginator = client.get_paginator('describe_metric_filters')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `CloudWatchLogs.Client.describe_metric_filters()`.

Request Syntax

```
response_iterator = paginator.paginate(
    logGroupName='string',
    filterNamePrefix='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **logGroupName** (*string*) – **[REQUIRED]**
The log group name for which metric filters are to be listed.
- **filterNamePrefix** (*string*) – Will only return metric filters that match the provided filterNamePrefix. If you don't specify a value, no prefix filter is applied.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
    'metricFilters': [
        {
            'filterName': 'string',
            'filterPattern': 'string',
            'metricTransformations': [
                {
                    'metricName': 'string',
```

```
        'metricNamespace': 'string',
        'metricValue': 'string'
    },
    ],
    'creationTime': 123
},
],
'NextToken': 'string'
}
```

Response Structure

- *(dict)* –
 - **metricFilters** (*list*) –
 - * *(dict)* –

Metric filters can be used to express how Amazon CloudWatch Logs would extract metric observations from ingested log events and transform them to metric data in a CloudWatch metric.

- **filterName** (*string*) –

A name for a metric or subscription filter.

- **filterPattern** (*string*) –

A symbolic description of how Amazon CloudWatch Logs should interpret the data in each log event. For example, a log event may contain timestamps, IP addresses, strings, and so on. You use the filter pattern to specify what to look for in the log event message.

- **metricTransformations** (*list*) –
- *(dict)* –
- **metricName** (*string*) –

The name of the CloudWatch metric to which the monitored log information should be published. For example, you may publish to a metric called ErrorCount.

- **metricNamespace** (*string*) –

The destination namespace of the new CloudWatch metric.

- **metricValue** (*string*) –

What to publish to the metric. For example, if you’re counting the occurrences of a particular term like “Error”, the value will be “1” for each occurrence. If you’re counting the bytes transferred the published value will be the value in the log event.

- **creationTime** (*integer*) –

A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

- **NextToken** (*string*) –

A token to resume pagination.

class CloudWatchLogs.Paginator.**DescribeSubscriptionFilters**

```
paginator = client.get_paginator('describe_subscription_filters')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `CloudWatchLogs.Client.describe_subscription_filters()`.

Request Syntax

```
response_iterator = paginator.paginate(
    logGroupName='string',
    filterNamePrefix='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **logGroupName** (*string*) – [REQUIRED]

The log group name for which subscription filters are to be listed.

- **filterNamePrefix** (*string*) – Will only return subscription filters that match the provided filterNamePrefix. If you don't specify a value, no prefix filter is applied.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns**Response Syntax**

```
{
    'subscriptionFilters': [
        {
            'filterName': 'string',
            'logGroupName': 'string',
            'filterPattern': 'string',
            'destinationArn': 'string',
            'roleArn': 'string',
            'creationTime': 123
        },
    ],
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **subscriptionFilters** (*list*) –
 - * (*dict*) –

- **filterName** (*string*) –

A name for a metric or subscription filter.

- **logGroupName** (*string*) –

- **filterPattern** (*string*) –

A symbolic description of how Amazon CloudWatch Logs should interpret the data in each log event. For example, a log event may contain timestamps, IP addresses, strings, and so on. You use the filter pattern to specify what to look for in the log event message.

- **destinationArn** (*string*) –

- **roleArn** (*string*) –

- **creationTime** (*integer*) –

A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

- **NextToken** (*string*) –

A token to resume pagination.

class CloudWatchLogs.Paginator.**FilterLogEvents**

```
paginator = client.get_paginator('filter_log_events')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `CloudWatchLogs.Client.filter_log_events()`.

Request Syntax

```
response_iterator = paginator.paginate(
    logGroupName='string',
    logStreamNames=[
        'string',
    ],
    startTime=123,
    endTime=123,
    filterPattern='string',
    interleaved=True|False,
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **logGroupName** (*string*) – [REQUIRED]

The name of the log group to query.

- **logStreamNames** (*list*) – Optional list of log stream names within the specified log group to search. Defaults to all the log streams in the log group.
 - (*string*) –

- **startTime** (*integer*) – A unix timestamp indicating the start time of the range for the request. If provided, events with a timestamp prior to this time will not be returned.

- **endTime** (*integer*) – A unix timestamp indicating the end time of the range for the request. If provided, events with a timestamp later than this time will not be returned.
- **filterPattern** (*string*) – A valid CloudWatch Logs filter pattern to use for filtering the response. If not provided, all the events are matched.
- **interleaved** (*boolean*) – If provided, the API will make a best effort to provide responses that contain events from multiple log streams within the log group interleaved in a single response. If not provided, all the matched log events in the first log stream will be searched first, then those in the next log stream, etc.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'events': [
    {
      'logStreamName': 'string',
      'timestamp': 123,
      'message': 'string',
      'ingestionTime': 123,
      'eventId': 'string'
    },
  ],
  'searchedLogStreams': [
    {
      'logStreamName': 'string',
      'searchedCompletely': True|False
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **events** (*list*) –
A list of `FilteredLogEvent` objects representing the matched events from the request.
 - * (*dict*) –
Represents a matched event from a `FilterLogEvents` request.
 - **logStreamName** (*string*) –
The name of the log stream this event belongs to.

- **timestamp** (*integer*) –
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.
- **message** (*string*) –
The data contained in the log event.
- **ingestionTime** (*integer*) –
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.
- **eventId** (*string*) –
A unique identifier for this event.
- **searchedLogStreams** (*list*) –
A list of `SearchedLogStream` objects indicating which log streams have been searched in this request and whether each has been searched completely or still has more to be paginated.
 - * (*dict*) –
An object indicating the search status of a log stream in a `FilterLogEvents` request.
 - **logStreamName** (*string*) –
The name of the log stream.
 - **searchedCompletely** (*boolean*) –
Indicates whether all the events in this log stream were searched or more data exists to search by paginating further.
- **NextToken** (*string*) –
A token to resume pagination.

MachineLearning

Table of Contents

- MachineLearning
 - Client
 - Paginators
 - Waiters

Client

`class MachineLearning.Client`

A low-level client representing Amazon Machine Learning:

```
import boto3

client = boto3.client('machinelearning')
```

These are the available methods:

- `can_paginate()`
- `create_batch_prediction()`
- `create_data_source_from_rds()`

- `create_data_source_from_redshift()`
- `create_data_source_from_s3()`
- `create_evaluation()`
- `create_ml_model()`
- `create_realtime_endpoint()`
- `delete_batch_prediction()`
- `delete_data_source()`
- `delete_evaluation()`
- `delete_ml_model()`
- `delete_realtime_endpoint()`
- `describe_batch_predictions()`
- `describe_data_sources()`
- `describe_evaluations()`
- `describe_ml_models()`
- `generate_presigned_url()`
- `get_batch_prediction()`
- `get_data_source()`
- `get_evaluation()`
- `get_ml_model()`
- `get_paginator()`
- `get_waiter()`
- `predict()`
- `update_batch_prediction()`
- `update_data_source()`
- `update_evaluation()`
- `update_ml_model()`

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

create_batch_prediction (***kwargs*)

Generates predictions for a group of observations. The observations to process exist in one or more data files referenced by a `DataSource`. This operation creates a new `BatchPrediction`, and uses an `MLModel` and the data files referenced by the `DataSource` as information sources.

`CreateBatchPrediction` is an asynchronous operation. In response to `CreateBatchPrediction`, Amazon Machine Learning (Amazon ML) immediately returns and sets the `BatchPrediction` status to `PENDING`. After the `BatchPrediction` completes, Amazon ML sets the status to `COMPLETED`.

You can poll for status updates by using the `GetBatchPrediction` operation and checking the `Status` parameter of the result. After the `COMPLETED` status appears, the results are available in the location specified by the `OutputUri` parameter.

Request Syntax

```
response = client.create_batch_prediction(
    BatchPredictionId='string',
    BatchPredictionName='string',
    MLModelId='string',
    BatchPredictionDataSourceId='string',
```

```
    OutputUri='string'  
)
```

Parameters

- **BatchPredictionId** (*string*) – [REQUIRED]

A user-supplied ID that uniquely identifies the `BatchPrediction`.

- **BatchPredictionName** (*string*) – A user-supplied name or description of the `BatchPrediction`. `BatchPredictionName` can only use the UTF-8 character set.
- **MLModelId** (*string*) – [REQUIRED]

The ID of the `MLModel` that will generate predictions for the group of observations.

- **BatchPredictionDataSourceId** (*string*) – [REQUIRED]

The ID of the `DataSource` that points to the group of observations to predict.

- **OutputUri** (*string*) – [REQUIRED]

The location of an Amazon Simple Storage Service (Amazon S3) bucket or directory to store the batch prediction results. The following substrings are not allowed in the s3 key portion of the “outputURI” field: ‘:’, ‘//’, ‘/.’, ‘./.’.

Amazon ML needs permissions to store and retrieve the logs on your behalf. For information about how to set permissions, see the [Amazon Machine Learning Developer Guide](#).

Return type dict

Returns

Response Syntax

```
{  
    'BatchPredictionId': 'string'  
}
```

Response Structure

- (*dict*) –

Represents the output of a `CreateBatchPrediction` operation, and is an acknowledgement that Amazon ML received the request.

The `CreateBatchPrediction` operation is asynchronous. You can poll for status updates by using the `GetBatchPrediction` operation and checking the `Status` parameter of the result.

- **BatchPredictionId** (*string*) –

A user-supplied ID that uniquely identifies the `BatchPrediction`. This value is identical to the value of the `BatchPredictionId` in the request.

`create_data_source_from_rds` (**kwargs)

Creates a `DataSource` object from an [Amazon Relational Database Service](#) (Amazon RDS). A `DataSource` references data that can be used to perform `CreateMLModel`, `CreateEvaluation`, or `CreateBatchPrediction` operations.

`CreateDataSourceFromRDS` is an asynchronous operation. In response to `CreateDataSourceFromRDS`, Amazon Machine Learning (Amazon ML) immediately returns and sets the `DataSource` status to `PENDING`. After the `DataSource` is created and ready for use, Amazon ML sets the `Status` parameter to `COMPLETED`. `DataSource` in `COMPLETED` or `PENDING`

status can only be used to perform `CreateMLModel` , `CreateEvaluation` , or `CreateBatchPrediction` operations.

If Amazon ML cannot accept the input source, it sets the `Status` parameter to `FAILED` and includes an error message in the `Message` attribute of the `GetDataSource` operation response.

Request Syntax

```
response = client.create_data_source_from_rds(
    DataSourceId='string',
    DataSourceName='string',
    RDSData={
        'DatabaseInformation': {
            'InstanceIdentifier': 'string',
            'DatabaseName': 'string'
        },
        'SelectSqlQuery': 'string',
        'DatabaseCredentials': {
            'Username': 'string',
            'Password': 'string'
        },
        'S3StagingLocation': 'string',
        'DataRearrangement': 'string',
        'DataSchema': 'string',
        'DataSchemaUri': 'string',
        'ResourceRole': 'string',
        'ServiceRole': 'string',
        'SubnetId': 'string',
        'SecurityGroupIds': [
            'string',
        ]
    },
    RoleARN='string',
    ComputeStatistics=True|False
)
```

Parameters

- **DataSourceId** (*string*) – [REQUIRED]

A user-supplied ID that uniquely identifies the `DataSource` . Typically, an Amazon Resource Number (ARN) becomes the ID for a `DataSource` .

- **DataSourceName** (*string*) – A user-supplied name or description of the `DataSource` .
- **RDSData** (*dict*) – [REQUIRED]

The data specification of an Amazon RDS `DataSource` :

- **DatabaseInformation** - * **DatabaseName** - Name of the Amazon RDS database.
- **InstanceIdentifier** - Unique identifier for the Amazon RDS database instance.
- **DatabaseCredentials** - AWS Identity and Access Management (IAM) credentials that are used to connect to the Amazon RDS database.
- **ResourceRole** - Role (`DataPipelineDefaultResourceRole`) assumed by an Amazon Elastic Compute Cloud (EC2) instance to carry out the copy task from Amazon RDS to Amazon S3. For more information, see [Role templates](#) for data pipelines.
- **ServiceRole** - Role (`DataPipelineDefaultRole`) assumed by the AWS Data Pipeline service to monitor the progress of the copy task from Amazon

RDS to Amazon Simple Storage Service (S3). For more information, see [Role templates](#) for data pipelines.

- **SecurityInfo** - Security information to use to access an Amazon RDS instance. You need to set up appropriate ingress rules for the security entity IDs provided to allow access to the Amazon RDS instance. Specify a [SubnetId , SecurityGroupIds] pair for a VPC-based Amazon RDS instance.
- **SelectSqlQuery** - Query that is used to retrieve the observation data for the DataSource .
- **S3StagingLocation** - Amazon S3 location for staging RDS data. The data retrieved from Amazon RDS using SelectSqlQuery is stored in this location.
- **DataSchemaUri** - Amazon S3 location of the DataSchema .
- **DataSchema** - A JSON string representing the schema. This is not required if DataSchemaUri is specified.
- **DataRearrangement** - A JSON string representing the splitting requirement of a DataSource .
Sample -
{"\"splitting\": {\"percentBegin\":10, \"percentEnd\":60}}

- **DatabaseInformation** (*dict*) – [REQUIRED]

Describes the DatabaseName and InstanceIdentifier of an Amazon RDS database.

* **InstanceIdentifier** (*string*) – [REQUIRED]

The ID of an RDS DB instance.

* **DatabaseName** (*string*) – [REQUIRED]

The name of a database hosted on an RDS DB instance.

- **SelectSqlQuery** (*string*) – [REQUIRED]

The query that is used to retrieve the observation data for the DataSource .

- **DatabaseCredentials** (*dict*) – [REQUIRED]

The AWS Identity and Access Management (IAM) credentials that are used connect to the Amazon RDS database.

* **Username** (*string*) – [REQUIRED]

The username to be used by Amazon ML to connect to database on an Amazon RDS instance. The username should have sufficient permissions to execute an RDSSelectSqlQuery query.

* **Password** (*string*) – [REQUIRED]

The password to be used by Amazon ML to connect to a database on an RDS DB instance. The password should have sufficient permissions to execute the RDSSelectQuery query.

- **S3StagingLocation** (*string*) – [REQUIRED]

The Amazon S3 location for staging Amazon RDS data. The data retrieved from Amazon RDS using SelectSqlQuery is stored in this location.

- **DataRearrangement** (*string*) –

DataRearrangement - A JSON string that represents the splitting requirement of a DataSource .

Sample - {"\"splitting\": {\"percentBegin\":10, \"percentEnd\":60}}

- **DataSchema** (*string*) –

A JSON string that represents the schema for an Amazon RDS `DataSource`. The `DataSchema` defines the structure of the observation data in the data file(s) referenced in the `DataSource`.

A `DataSchema` is not required if you specify a `DataSchemaUri`

Define your `DataSchema` as a series of key-value pairs. `attributes` and `excludedVariableNames` have an array of key-value pairs for their value. Use the following format to define your `DataSchema`.

```
{ "version": "1.0",
  "recordAnnotationFieldName": "F1",
  "recordWeightFieldName": "F2",
  "targetFieldName": "F3",
  "dataFormat": "CSV",
  "dataFileContainsHeader": true,
  "attributes": [
    { "fieldName": "F1", "fieldType": "TEXT" }, { "fieldName": "F2",
    "fieldType": "NUMERIC" }, { "fieldName": "F3", "fieldType": "CAT-
    EGORICAL" }, { "fieldName": "F4", "fieldType": "NUMERIC" },
    { "fieldName": "F5", "fieldType": "CATEGORICAL" }, { "field-
    Name": "F6", "fieldType": "TEXT" }, { "fieldName": "F7", "fieldType":
    "WEIGHTED_INT_SEQUENCE" }, { "fieldName": "F8", "fieldType":
    "WEIGHTED_STRING_SEQUENCE" } ],
  "excludedVariableNames": [ "F6" ] }
```

– **DataSchemaUri** (*string*) –

The Amazon S3 location of the `DataSchema`.

– **ResourceRole** (*string*) – **[REQUIRED]**

The role (`DataPipelineDefaultResourceRole`) assumed by an Amazon Elastic Compute Cloud (Amazon EC2) instance to carry out the copy operation from Amazon RDS to an Amazon S3 task. For more information, see [Role templates](#) for data pipelines.

– **ServiceRole** (*string*) – **[REQUIRED]**

The role (`DataPipelineDefaultRole`) assumed by AWS Data Pipeline service to monitor the progress of the copy task from Amazon RDS to Amazon S3. For more information, see [Role templates](#) for data pipelines.

– **SubnetId** (*string*) – **[REQUIRED]**

The subnet ID to be used to access a VPC-based RDS DB instance. This attribute is used by Data Pipeline to carry out the copy task from Amazon RDS to Amazon S3.

– **SecurityGroupIds** (*list*) – **[REQUIRED]**

The security group IDs to be used to access a VPC-based RDS DB instance. Ensure that there are appropriate ingress rules set up to allow access to the RDS DB instance. This attribute is used by Data Pipeline to carry out the copy operation from Amazon RDS to an Amazon S3 task.

* (*string*) –

• **RoleARN** (*string*) – **[REQUIRED]**

The role that Amazon ML assumes on behalf of the user to create and activate a data pipeline in the user's account and copy data (using the `SelectSqlQuery`) query from Amazon RDS to Amazon S3.

- **ComputeStatistics** (*boolean*) – The compute statistics for a `DataSource` . The statistics are generated from the observation data referenced by a `DataSource` . Amazon ML uses the statistics internally during an `MLModel` training. This parameter must be set to `true` if the ““ DataSource “” needs to be used for `MLModel` training.

Return type dict

Returns

Response Syntax

```
{
    'DataSourceId': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a `CreateDataSourceFromRDS` operation, and is an acknowledgement that Amazon ML received the request.

The `CreateDataSourceFromRDS` operation is asynchronous. You can poll for updates by using the `GetBatchPrediction` operation and checking the `Status` parameter. You can inspect the `Message` when `Status` shows up as `FAILED` . You can also check the progress of the copy operation by going to the `DataPipeline` console and looking up the pipeline using the `pipelineId` from the `describe` call.

- **DataSourceId** (*string*) –

A user-supplied ID that uniquely identifies the `datasource`. This value should be identical to the value of the `DataSourceId` in the request.

`create_data_source_from_redshift` (***kwargs*)

Creates a `DataSource` from [Amazon Redshift](#) . A `DataSource` references data that can be used to perform either `CreateMLModel` , `CreateEvaluation` or `CreateBatchPrediction` operations.

`CreateDataSourceFromRedshift` is an asynchronous operation. In response to `CreateDataSourceFromRedshift` , Amazon Machine Learning (Amazon ML) immediately returns and sets the `DataSource` status to `PENDING` . After the `DataSource` is created and ready for use, Amazon ML sets the `Status` parameter to `COMPLETED` . `DataSource` in `COMPLETED` or `PENDING` status can only be used to perform `CreateMLModel` , `CreateEvaluation` , or `CreateBatchPrediction` operations.

If Amazon ML cannot accept the input source, it sets the `Status` parameter to `FAILED` and includes an error message in the `Message` attribute of the `GetDataSource` operation response.

The observations should exist in the database hosted on an Amazon Redshift cluster and should be specified by a `SelectSqlQuery` . Amazon ML executes `Unload` command in Amazon Redshift to transfer the result set of `SelectSqlQuery` to `S3StagingLocation` .

After the `DataSource` is created, it's ready for use in evaluations and batch predictions. If you plan to use the `DataSource` to train an `MLModel` , the `DataSource` requires another item – a recipe. A recipe describes the observation variables that participate in training an `MLModel` . A recipe describes how each input variable will be used in training. Will the variable be included or excluded from training? Will the variable be manipulated, for example, combined with another variable or split apart into word combinations? The recipe provides answers to these questions. For more information, see the Amazon Machine Learning Developer Guide.

Request Syntax

```
response = client.create_data_source_from_redshift(
    DataSourceId='string',
    DataSourceName='string',
    DataSpec={
        'DatabaseInformation': {
            'DatabaseName': 'string',
            'ClusterIdentifier': 'string'
        },
        'SelectSqlQuery': 'string',
        'DatabaseCredentials': {
            'Username': 'string',
            'Password': 'string'
        },
        'S3StagingLocation': 'string',
        'DataRearrangement': 'string',
        'DataSchema': 'string',
        'DataSchemaUri': 'string'
    },
    RoleARN='string',
    ComputeStatistics=True|False
)
```

Parameters

- **DataSourceId** (*string*) – [REQUIRED]

A user-supplied ID that uniquely identifies the `DataSource`.

- **DataSourceName** (*string*) – A user-supplied name or description of the `DataSource`.

- **DataSpec** (*dict*) – [REQUIRED]

The data specification of an Amazon Redshift `DataSource`:

- **DatabaseInformation** - * **DatabaseName** - Name of the Amazon Redshift database.
- **ClusterIdentifier** - Unique ID for the Amazon Redshift cluster.
- **DatabaseCredentials** - AWS Identity and Access Management (IAM) credentials that are used to connect to the Amazon Redshift database.
- **SelectSqlQuery** - Query that is used to retrieve the observation data for the `Datasource`.
- **S3StagingLocation** - Amazon Simple Storage Service (Amazon S3) location for staging Amazon Redshift data. The data retrieved from Amazon Relational Database Service (Amazon RDS) using `SelectSqlQuery` is stored in this location.
- **DataSchemaUri** - Amazon S3 location of the `DataSchema`.
- **DataSchema** - A JSON string representing the schema. This is not required if `DataSchemaUri` is specified.
- **DataRearrangement** - A JSON string representing the splitting requirement of a `Datasource`. Sample - `"{\"splitting\":{\"percentBegin\":10,\"percentEnd\":60}}"`

- **DatabaseInformation** (*dict*) – [REQUIRED]

Describes the `DatabaseName` and `ClusterIdentifier` for an Amazon Redshift `DataSource`.

- * **DatabaseName** (*string*) – [REQUIRED]

The name of a database hosted on an Amazon Redshift cluster.

- * **ClusterIdentifier** (*string*) – [REQUIRED]

The ID of an Amazon Redshift cluster.

- **SelectSqlQuery** (*string*) – [REQUIRED]

Describes the SQL Query to execute on an Amazon Redshift database for an Amazon Redshift `DataSource`.

- **DatabaseCredentials** (*dict*) – [REQUIRED]

Describes AWS Identity and Access Management (IAM) credentials that are used connect to the Amazon Redshift database.

- * **Username** (*string*) – [REQUIRED]

A username to be used by Amazon Machine Learning (Amazon ML) to connect to a database on an Amazon Redshift cluster. The username should have sufficient permissions to execute the `RedshiftSelectSqlQuery` query. The username should be valid for an Amazon Redshift [USER](#).

- * **Password** (*string*) – [REQUIRED]

A password to be used by Amazon ML to connect to a database on an Amazon Redshift cluster. The password should have sufficient permissions to execute a `RedshiftSelectSqlQuery` query. The password should be valid for an Amazon Redshift [USER](#).

- **S3StagingLocation** (*string*) – [REQUIRED]

Describes an Amazon S3 location to store the result set of the `SelectSqlQuery` query.

- **DataRearrangement** (*string*) –

Describes the splitting specifications for a `DataSource`.

- **DataSchema** (*string*) –

A JSON string that represents the schema for an Amazon Redshift `DataSource`. The `DataSchema` defines the structure of the observation data in the data file(s) referenced in the `DataSource`.

A `DataSchema` is not required if you specify a `DataSchemaUri`.

Define your `DataSchema` as a series of key-value pairs. `attributes` and `excludedVariableNames` have an array of key-value pairs for their value. Use the following format to define your `DataSchema`.

```
{ "version": "1.0",
  "recordAnnotationFieldName": "F1",
  "recordWeightFieldName": "F2",
  "targetFieldName": "F3",
  "dataFormat": "CSV",
  "dataFileContainsHeader": true,
  "attributes": [
    { "fieldName": "F1", "fieldType": "TEXT" }, { "fieldName": "F2",
    "fieldType": "NUMERIC" }, { "fieldName": "F3", "fieldType": "CAT-
    EGORICAL" }, { "fieldName": "F4", "fieldType": "NUMERIC" },
    { "fieldName": "F5", "fieldType": "CATEGORICAL" }, { "field-
    Name": "F6", "fieldType": "TEXT" }, { "fieldName": "F7", "fieldType":
```

```
“WEIGHTED_INT_SEQUENCE” }, { “fieldName”: “F8”, “fieldType”:
“WEIGHTED_STRING_SEQUENCE” } ],
```

```
“excludedVariableNames”: [ “F6” ] }
```

– **DataSchemaUri** (*string*) –

Describes the schema location for an Amazon Redshift `DataSource`.

• **RoleARN** (*string*) – **[REQUIRED]**

A fully specified role Amazon Resource Name (ARN). Amazon ML assumes the role on behalf of the user to create the following:

- A security group to allow Amazon ML to execute the `SelectSqlQuery` query on an Amazon Redshift cluster
- An Amazon S3 bucket policy to grant Amazon ML read/write permissions on the `S3StagingLocation`

• **ComputeStatistics** (*boolean*) – The compute statistics for a `DataSource`. The statistics are generated from the observation data referenced by a `DataSource`. Amazon ML uses the statistics internally during `MLModel` training. This parameter must be set to `true` if the `““ DataSource ““` needs to be used for `MLModel` training

Return type dict

Returns

Response Syntax

```
{
    'DataSourceId': 'string'
}
```

Response Structure

• (*dict*) –

Represents the output of a `CreateDataSourceFromRedshift` operation, and is an acknowledgement that Amazon ML received the request.

The `CreateDataSourceFromRedshift` operation is asynchronous. You can poll for updates by using the `GetBatchPrediction` operation and checking the `Status` parameter.

– **DataSourceId** (*string*) –

A user-supplied ID that uniquely identifies the `datasource`. This value should be identical to the value of the `DataSourceId` in the request.

`create_data_source_from_s3` (**kwargs)

Creates a `DataSource` object. A `DataSource` references data that can be used to perform `CreateMLModel`, `CreateEvaluation`, or `CreateBatchPrediction` operations.

`CreateDataSourceFromS3` is an asynchronous operation. In response to `CreateDataSourceFromS3`, Amazon Machine Learning (Amazon ML) immediately returns and sets the `DataSource` status to `PENDING`. After the `DataSource` is created and ready for use, Amazon ML sets the `Status` parameter to `COMPLETED`. `DataSource` in `COMPLETED` or `PENDING` status can only be used to perform `CreateMLModel`, `CreateEvaluation` or `CreateBatchPrediction` operations.

If Amazon ML cannot accept the input source, it sets the `Status` parameter to `FAILED` and includes an error message in the `Message` attribute of the `GetDataSource` operation response.

The observation data used in a `DataSource` should be ready to use; that is, it should have a consistent structure, and missing data values should be kept to a minimum. The observation data must reside in one or more CSV files in an Amazon Simple Storage Service (Amazon S3) bucket, along with a schema

that describes the data items by name and type. The same schema must be used for all of the data files referenced by the `DataSource`.

After the `DataSource` has been created, it's ready to use in evaluations and batch predictions. If you plan to use the `DataSource` to train an `MLModel`, the `DataSource` requires another item: a recipe. A recipe describes the observation variables that participate in training an `MLModel`. A recipe describes how each input variable will be used in training. Will the variable be included or excluded from training? Will the variable be manipulated, for example, combined with another variable, or split apart into word combinations? The recipe provides answers to these questions. For more information, see the [Amazon Machine Learning Developer Guide](#).

Request Syntax

```
response = client.create_data_source_from_s3(
    DataSourceId='string',
    DataSourceName='string',
    DataSpec={
        'DataLocationS3': 'string',
        'DataRearrangement': 'string',
        'DataSchema': 'string',
        'DataSchemaLocationS3': 'string'
    },
    ComputeStatistics=True|False
)
```

Parameters

- **DataSourceId** (*string*) – [REQUIRED]

A user-supplied identifier that uniquely identifies the `DataSource`.

- **DataSourceName** (*string*) – A user-supplied name or description of the `DataSource`.

- **DataSpec** (*dict*) – [REQUIRED]

The data specification of a `DataSource`:

- **DataLocationS3** - Amazon Simple Storage Service (Amazon S3) location of the observation data.
- **DataSchemaLocationS3** - Amazon S3 location of the `DataSchema`.
- **DataSchema** - A JSON string representing the schema. This is not required if `DataSchemaUri` is specified.
- **DataRearrangement** - A JSON string representing the splitting requirement of a `Datasource`. Sample - `"{\"splitting\":{\"percentBegin\":10,\"percentEnd\":60}}"`

- **DataLocationS3** (*string*) – [REQUIRED]

The location of the data file(s) used by a `DataSource`. The URI specifies a data file or an Amazon Simple Storage Service (Amazon S3) directory or bucket containing data files.

- **DataRearrangement** (*string*) –

Describes the splitting requirement of a `Datasource`.

- **DataSchema** (*string*) –

A JSON string that represents the schema for an Amazon S3 `DataSource`. The `DataSchema` defines the structure of the observation data in the data file(s) referenced in the `DataSource`.

Define your `DataSchema` as a series of key-value pairs. `attributes` and `excludedVariableNames` have an array of key-value pairs for

their value. Use the following format to define your `DataSchema`.

```
{ "version": "1.0",
  "recordAnnotationFieldName": "F1",
  "recordWeightFieldName": "F2",
  "targetFieldName": "F3",
  "dataFormat": "CSV",
  "dataFileContainsHeader": true,
  "attributes": [
    { "fieldName": "F1", "fieldType": "TEXT" }, { "fieldName": "F2",
    "fieldType": "NUMERIC" }, { "fieldName": "F3", "fieldType": "CAT-
    EGORICAL" }, { "fieldName": "F4", "fieldType": "NUMERIC" },
    { "fieldName": "F5", "fieldType": "CATEGORICAL" }, { "field-
    Name": "F6", "fieldType": "TEXT" }, { "fieldName": "F7", "fieldType":
    "WEIGHTED_INT_SEQUENCE" }, { "fieldName": "F8", "fieldType":
    "WEIGHTED_STRING_SEQUENCE" } ],
  "excludedVariableNames": [ "F6" ] }
```

– **DataSchemaLocationS3** (*string*) –

Describes the schema Location in Amazon S3.

- **ComputeStatistics** (*boolean*) – The compute statistics for a `DataSource`. The statistics are generated from the observation data referenced by a `DataSource`. Amazon ML uses the statistics internally during an `MLModel` training. This parameter must be set to `true` if the ““ DataSource “” needs to be used for `MLModel` training

Return type dict

Returns

Response Syntax

```
{
  'DataSourceId': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a `CreateDataSourceFromS3` operation, and is an acknowledgement that Amazon ML received the request.

The `CreateDataSourceFromS3` operation is asynchronous. You can poll for updates by using the `GetBatchPrediction` operation and checking the `Status` parameter.

- **DataSourceId** (*string*) –

A user-supplied ID that uniquely identifies the `datasource`. This value should be identical to the value of the `DataSourceId` in the request.

create_evaluation (***kwargs*)

Creates a new `Evaluation` of an `MLModel`. An `MLModel` is evaluated on a set of observations associated to a `DataSource`. Like a `DataSource` for an `MLModel`, the `DataSource` for an `Evaluation` contains values for the Target Variable. The `Evaluation` compares the predicted result for each observation to the actual outcome and provides a summary so that you know how effective the `MLModel` functions on the test data. Evaluation generates a relevant performance metric such as

BinaryAUC, RegressionRMSE or MulticlassAvgFScore based on the corresponding MLModelType : BINARY , REGRESSION or MULTICLASS .

CreateEvaluation is an asynchronous operation. In response to CreateEvaluation , Amazon Machine Learning (Amazon ML) immediately returns and sets the evaluation status to PENDING . After the Evaluation is created and ready for use, Amazon ML sets the status to COMPLETED .

You can use the GetEvaluation operation to check progress of the evaluation during the creation operation.

Request Syntax

```
response = client.create_evaluation(
    EvaluationId='string',
    EvaluationName='string',
    MLModelId='string',
    EvaluationDataSourceId='string'
)
```

Parameters

- **EvaluationId** (*string*) – [REQUIRED]
A user-supplied ID that uniquely identifies the Evaluation .
- **EvaluationName** (*string*) – A user-supplied name or description of the Evaluation .
- **MLModelId** (*string*) – [REQUIRED]
The ID of the MLModel to evaluate.
The schema used in creating the MLModel must match the schema of the DataSource used in the Evaluation .
- **EvaluationDataSourceId** (*string*) – [REQUIRED]
The ID of the DataSource for the evaluation. The schema of the DataSource must match the schema used to create the MLModel .

Return type dict

Returns

Response Syntax

```
{
    'EvaluationId': 'string'
}
```

Response Structure

- (*dict*) –
Represents the output of a CreateEvaluation operation, and is an acknowledgement that Amazon ML received the request.
CreateEvaluation operation is asynchronous. You can poll for status updates by using the GetEvaluation operation and checking the Status parameter.
- **EvaluationId** (*string*) –
The user-supplied ID that uniquely identifies the Evaluation . This value should be identical to the value of the EvaluationId in the request.

create_ml_model (***kwargs*)

Creates a new MLModel using the data files and the recipe as information sources.

An `MLModel` is nearly immutable. Users can only update the `MLModelName` and the `ScoreThreshold` in an `MLModel` without creating a new `MLModel`.

`CreateMLModel` is an asynchronous operation. In response to `CreateMLModel`, Amazon Machine Learning (Amazon ML) immediately returns and sets the `MLModel` status to `PENDING`. After the `MLModel` is created and ready for use, Amazon ML sets the status to `COMPLETED`.

You can use the `GetMLModel` operation to check progress of the `MLModel` during the creation operation.

`CreateMLModel` requires a `DataSource` with computed statistics, which can be created by setting `ComputeStatistics` to `true` in `CreateDataSourceFromRDS`, `CreateDataSourceFromS3`, or `CreateDataSourceFromRedshift` operations.

Request Syntax

```
response = client.create_ml_model(
    MLModelId='string',
    MLModelName='string',
    MLModelType='REGRESSION'|'BINARY'|'MULTICLASS',
    Parameters={
        'string': 'string'
    },
    TrainingDataSourceId='string',
    Recipe='string',
    RecipeUri='string'
)
```

Parameters

- **MLModelId** (*string*) – [REQUIRED]

A user-supplied ID that uniquely identifies the `MLModel`.

- **MLModelName** (*string*) – A user-supplied name or description of the `MLModel`.

- **MLModelType** (*string*) – [REQUIRED]

The category of supervised learning that this `MLModel` will address. Choose from the following types:

- Choose `REGRESSION` if the `MLModel` will be used to predict a numeric value.
- Choose `BINARY` if the `MLModel` result has two possible values.
- Choose `MULTICLASS` if the `MLModel` result has a limited number of values.

For more information, see the [Amazon Machine Learning Developer Guide](#).

- **Parameters** (*dict*) – A list of the training parameters in the `MLModel`. The list is implemented as a map of key/value pairs.

The following is the current set of training parameters:

- `sgd.l1RegularizationAmount` - Coefficient regularization L1 norm. It controls overfitting the data by penalizing large coefficients. This tends to drive coefficients to zero, resulting in sparse feature set. If you use this parameter, start by specifying a small value such as 1.0E-08. The value is a double that ranges from 0 to `MAX_DOUBLE`. The default is not to use L1 normalization. The parameter cannot be used when L2 is specified. Use this parameter sparingly.
- `sgd.l2RegularizationAmount` - Coefficient regularization L2 norm. It controls overfitting the data by penalizing large coefficients. This tends to drive coefficients to small, nonzero values. If you use this parameter, start by specifying a small value such as 1.0E-08. The value is a double that ranges from 0 to `MAX_DOUBLE`. The default is not to use

L2 normalization. This cannot be used when L1 is specified. Use this parameter sparingly.

- `sgd.maxPasses` - Number of times that the training process traverses the observations to build the `MLModel`. The value is an integer that ranges from 1 to 10000. The default value is 10.
- `sgd.maxMLModelSizeInBytes` - Maximum allowed size of the model. Depending on the input data, the size of the model might affect its performance. The value is an integer that ranges from 100000 to 2147483648. The default value is 33554432.
- *(string)* -
String type.
* *(string)* -
String type.

- **TrainingDataSourceId** *(string)* - [REQUIRED]

The `DataSource` that points to the training data.

- **Recipe** *(string)* - The data recipe for creating `MLModel`. You must specify either the recipe or its URI. If you don't specify a recipe or its URI, Amazon ML creates a default.
- **RecipeUri** *(string)* - The Amazon Simple Storage Service (Amazon S3) location and file name that contains the `MLModel` recipe. You must specify either the recipe or its URI. If you don't specify a recipe or its URI, Amazon ML creates a default.

Return type dict

Returns

Response Syntax

```
{
    'MLModelId': 'string'
}
```

Response Structure

- *(dict)* -

Represents the output of a `CreateMLModel` operation, and is an acknowledgment that Amazon ML received the request.

The `CreateMLModel` operation is asynchronous. You can poll for status updates by using the `GetMLModel` operation and checking the `Status` parameter.

- **MLModelId** *(string)* -

A user-supplied ID that uniquely identifies the `MLModel`. This value should be identical to the value of the `MLModelId` in the request.

create_realtime_endpoint (***kwargs*)

Creates a real-time endpoint for the `MLModel`. The endpoint contains the URI of the `MLModel`; that is, the location to send real-time prediction requests for the specified `MLModel`.

Request Syntax

```
response = client.create_realtime_endpoint(
    MLModelId='string'
)
```

Parameters `MLModelId` (*string*) – [REQUIRED]

The ID assigned to the `MLModel` during creation.

Return type dict

Returns**Response Syntax**

```
{
  'MLModelId': 'string',
  'RealtimeEndpointInfo': {
    'PeakRequestsPerSecond': 123,
    'CreatedAt': datetime(2015, 1, 1),
    'EndpointUrl': 'string',
    'EndpointStatus': 'NONE' | 'READY' | 'UPDATING' | 'FAILED'
  }
}
```

Response Structure

- (*dict*) –

Represents the output of an `CreateRealtimeEndpoint` operation.

The result contains the `MLModelId` and the endpoint information for the `MLModel`.

Note: The endpoint information includes the URI of the `MLModel`; that is, the location to send online prediction requests for the specified `MLModel`.

- **MLModelId** (*string*) –

A user-supplied ID that uniquely identifies the `MLModel`. This value should be identical to the value of the `MLModelId` in the request.

- **RealtimeEndpointInfo** (*dict*) –

The endpoint information of the `MLModel`

- * **PeakRequestsPerSecond** (*integer*) –

The maximum processing rate for the real-time endpoint for `MLModel`, measured in incoming requests per second.

- * **CreatedAt** (*datetime*) –

The time that the request to create the real-time endpoint for the `MLModel` was received. The time is expressed in epoch time.

- * **EndpointUrl** (*string*) –

The URI that specifies where to send real-time prediction requests for the `MLModel`.

Note: Note

The application must wait until the real-time endpoint is ready before using this URI.

- * **EndpointStatus** (*string*) –

The current status of the real-time endpoint for the `MLModel`. This element can have one of the following values:

- NONE - Endpoint does not exist or was previously deleted.

- READY - Endpoint is ready to be used for real-time predictions.
- UPDATING - Updating/creating the endpoint.

delete_batch_prediction (***kwargs*)

Assigns the DELETED status to a BatchPrediction, rendering it unusable.

After using the DeleteBatchPrediction operation, you can use the GetBatchPrediction operation to verify that the status of the BatchPrediction changed to DELETED.

Caution: The result of the DeleteBatchPrediction operation is irreversible.

Request Syntax

```
response = client.delete_batch_prediction(  
    BatchPredictionId='string'  
)
```

Parameters BatchPredictionId (*string*) – **[REQUIRED]**

A user-supplied ID that uniquely identifies the BatchPrediction.

Return type dict

Returns**Response Syntax**

```
{  
    'BatchPredictionId': 'string'  
}
```

Response Structure

- (*dict*) –

Represents the output of a DeleteBatchPrediction operation.

You can use the GetBatchPrediction operation and check the value of the Status parameter to see whether a BatchPrediction is marked as DELETED.

- **BatchPredictionId** (*string*) –

A user-supplied ID that uniquely identifies the BatchPrediction. This value should be identical to the value of the BatchPredictionID in the request.

delete_data_source (***kwargs*)

Assigns the DELETED status to a DataSource, rendering it unusable.

After using the DeleteDataSource operation, you can use the GetDataSource operation to verify that the status of the DataSource changed to DELETED.

Caution: The results of the DeleteDataSource operation are irreversible.

Request Syntax

```
response = client.delete_data_source(  
    DataSourceId='string'  
)
```

Parameters DataSourceId (*string*) – **[REQUIRED]**

A user-supplied ID that uniquely identifies the DataSource.

Return type dict

Returns**Response Syntax**

```
{
    'DataSourceId': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a DeleteDataSource operation.

- **DataSourceId** (*string*) –

A user-supplied ID that uniquely identifies the DataSource . This value should be identical to the value of the DataSourceId in the request.

delete_evaluation (***kwargs*)

Assigns the DELETED status to an Evaluation , rendering it unusable.

After invoking the DeleteEvaluation operation, you can use the GetEvaluation operation to verify that the status of the Evaluation changed to DELETED .

Caution: The results of the DeleteEvaluation operation are irreversible.

Request Syntax

```
response = client.delete_evaluation(
    EvaluationId='string'
)
```

Parameters **EvaluationId** (*string*) – **[REQUIRED]**

A user-supplied ID that uniquely identifies the Evaluation to delete.

Return type dict

Returns**Response Syntax**

```
{
    'EvaluationId': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a DeleteEvaluation operation. The output indicates that Amazon Machine Learning (Amazon ML) received the request.

You can use the GetEvaluation operation and check the value of the Status parameter to see whether an Evaluation is marked as DELETED .

- **EvaluationId** (*string*) –

A user-supplied ID that uniquely identifies the Evaluation . This value should be identical to the value of the EvaluationId in the request.

delete_ml_model (***kwargs*)

Assigns the DELETED status to an MLModel , rendering it unusable.

After using the DeleteMLModel operation, you can use the GetMLModel operation to verify that the status of the MLModel changed to DELETED.

Caution: The result of the `DeleteMLModel` operation is irreversible.

Request Syntax

```
response = client.delete_ml_model(
    MLModelId='string'
)
```

Parameters `MLModelId` (*string*) – [REQUIRED]

A user-supplied ID that uniquely identifies the `MLModel` .

Return type dict

Returns

Response Syntax

```
{
    'MLModelId': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a `DeleteMLModel` operation.

You can use the `GetMLModel` operation and check the value of the `Status` parameter to see whether an `MLModel` is marked as `DELETED` .

– `MLModelId` (*string*) –

A user-supplied ID that uniquely identifies the `MLModel` . This value should be identical to the value of the `MLModelID` in the request.

`delete_realtime_endpoint` (***kwargs*)

Deletes a real time endpoint of an `MLModel` .

Request Syntax

```
response = client.delete_realtime_endpoint(
    MLModelId='string'
)
```

Parameters `MLModelId` (*string*) – [REQUIRED]

The ID assigned to the `MLModel` during creation.

Return type dict

Returns

Response Syntax

```
{
    'MLModelId': 'string',
    'RealtimeEndpointInfo': {
        'PeakRequestsPerSecond': 123,
        'CreatedAt': datetime(2015, 1, 1),
        'EndpointUrl': 'string',
        'EndpointStatus': 'NONE' | 'READY' | 'UPDATING' | 'FAILED'
    }
}
```

Response Structure

- (*dict*) –

Represents the output of an DeleteRealtimeEndpoint operation.

The result contains the `MLModelId` and the endpoint information for the `MLModel`.

- **MLModelId** (*string*) –

A user-supplied ID that uniquely identifies the `MLModel`. This value should be identical to the value of the `MLModelId` in the request.

- **RealtimeEndpointInfo** (*dict*) –

The endpoint information of the `MLModel`

- * **PeakRequestsPerSecond** (*integer*) –

The maximum processing rate for the real-time endpoint for `MLModel`, measured in incoming requests per second.

- * **CreatedAt** (*datetime*) –

The time that the request to create the real-time endpoint for the `MLModel` was received. The time is expressed in epoch time.

- * **EndpointUrl** (*string*) –

The URI that specifies where to send real-time prediction requests for the `MLModel`.

Note: Note

The application must wait until the real-time endpoint is ready before using this URI.

-
- * **EndpointStatus** (*string*) –

The current status of the real-time endpoint for the `MLModel`. This element can have one of the following values:

- NONE - Endpoint does not exist or was previously deleted.
- READY - Endpoint is ready to be used for real-time predictions.
- UPDATING - Updating/creating the endpoint.

describe_batch_predictions (***kwargs*)

Returns a list of BatchPrediction operations that match the search criteria in the request.

Request Syntax

```
response = client.describe_batch_predictions(
    FilterVariable='CreatedAt'|'LastUpdatedAt'|'Status'|'Name'|'IAMUser'|'MLModelId'|'DataS
    EQ='string',
    GT='string',
    LT='string',
    GE='string',
    LE='string',
    NE='string',
    Prefix='string',
    SortOrder='asc'|'dsc',
    NextToken='string',
    Limit=123
)
```

Parameters

- **FilterVariable** (*string*) – Use one of the following variables to filter a list of BatchPrediction:
 - CreatedAt - Sets the search criteria to the BatchPrediction creation date.
 - Status - Sets the search criteria to the BatchPrediction status.
 - Name - Sets the search criteria to the contents of the BatchPrediction ******* Name.
 - IAMUser - Sets the search criteria to the user account that invoked the BatchPrediction creation.
 - MLModelId - Sets the search criteria to the MLModel used in the BatchPrediction.
 - DataSourceId - Sets the search criteria to the DataSource used in the BatchPrediction.
 - DataURI - Sets the search criteria to the data file(s) used in the BatchPrediction. The URL can identify either a file or an Amazon Simple Storage Solution (Amazon S3) bucket or directory.
- **EQ** (*string*) – The equal to operator. The BatchPrediction results will have FilterVariable values that exactly match the value specified with EQ.
- **GT** (*string*) – The greater than operator. The BatchPrediction results will have FilterVariable values that are greater than the value specified with GT.
- **LT** (*string*) – The less than operator. The BatchPrediction results will have FilterVariable values that are less than the value specified with LT.
- **GE** (*string*) – The greater than or equal to operator. The BatchPrediction results will have FilterVariable values that are greater than or equal to the value specified with GE.
- **LE** (*string*) – The less than or equal to operator. The BatchPrediction results will have FilterVariable values that are less than or equal to the value specified with LE.
- **NE** (*string*) – The not equal to operator. The BatchPrediction results will have FilterVariable values not equal to the value specified with NE.
- **Prefix** (*string*) – A string that is found at the beginning of a variable, such as Name or Id.

For example, a Batch Prediction operation could have the Name 2014-09-09-HolidayGiftMailer. To search for this BatchPrediction, select Name for the FilterVariable and any of the following strings for the Prefix:

- 2014-09
 - 2014-09-09
 - 2014-09-09-Holiday
- **SortOrder** (*string*) – A two-value parameter that determines the sequence of the resulting list of MLModel s.
 - asc - Arranges the list in ascending order (A-Z, 0-9).
 - dsc - Arranges the list in descending order (Z-A, 9-0).Results are sorted by FilterVariable.
 - **NextToken** (*string*) – An ID of the page in the paginated results.
 - **Limit** (*integer*) – The number of pages of information to include in the result. The range of acceptable values is 1 through 100. The default value is 100.

Return type dict

Returns

Response Syntax

```

{
  'Results': [
    {
      'BatchPredictionId': 'string',
      'MLModelId': 'string',
      'BatchPredictionDataSourceId': 'string',
      'InputDataLocationS3': 'string',
      'CreatedByIamUser': 'string',
      'CreatedAt': datetime(2015, 1, 1),
      'LastUpdatedAt': datetime(2015, 1, 1),
      'Name': 'string',
      'Status': 'PENDING' | 'INPROGRESS' | 'FAILED' | 'COMPLETED' | 'DELETED',
      'OutputUri': 'string',
      'Message': 'string'
    },
  ],
  'NextToken': 'string'
}

```

Response Structure

- (*dict*) –

Represents the output of a `DescribeBatchPredictions` operation. The content is essentially a list of `BatchPredictions`.

- **Results** (*list*) –

A list of `BatchPrediction` objects that meet the search criteria.

- * (*dict*) –

Represents the output of `GetBatchPrediction` operation.

The content consists of the detailed metadata, the status, and the data file information of a *Batch Prediction*.

- **BatchPredictionId** (*string*) –

The ID assigned to the `BatchPrediction` at creation. This value should be identical to the value of the `BatchPredictionID` in the request.

- **MLModelId** (*string*) –

The ID of the `MLModel` that generated predictions for the `BatchPrediction` request.

- **BatchPredictionDataSourceId** (*string*) –

The ID of the `DataSource` that points to the group of observations to predict.

- **InputDataLocationS3** (*string*) –

The location of the data file or directory in Amazon Simple Storage Service (Amazon S3).

- **CreatedByIamUser** (*string*) –

The AWS user account that invoked the `BatchPrediction`. The account type can be either an AWS root account or an AWS Identity and Access Management (IAM) user account.

- **CreatedAt** (*datetime*) –

The time that the `BatchPrediction` was created. The time is expressed in epoch time.

- **LastUpdatedAt** (*datetime*) –

The time of the most recent edit to the `BatchPrediction`. The time is expressed in epoch time.

- **Name** (*string*) –

A user-supplied name or description of the `BatchPrediction`.

- **Status** (*string*) –

The status of the `BatchPrediction`. This element can have one of the following values:

- `PENDING` - Amazon Machine Learning (Amazon ML) submitted a request to generate predictions for a batch of observations.
- `INPROGRESS` - The process is underway.
- `FAILED` - The request to perform a batch prediction did not run to completion. It is not usable.
- `COMPLETED` - The batch prediction process completed successfully.
- `DELETED` - The `BatchPrediction` is marked as deleted. It is not usable.
- **OutputUri** (*string*) –

The location of an Amazon S3 bucket or directory to receive the operation results. The following substrings are not allowed in the s3 key portion of the “outputURI” field: ‘:’, ‘//’, ‘/.’, ‘/./’.

- **Message** (*string*) –

A description of the most recent details about processing the batch prediction request.

- **NextToken** (*string*) –

The ID of the next page in the paginated results that indicates at least one more page follows.

describe_data_sources (**kwargs)

Returns a list of `DataSource` that match the search criteria in the request.

Request Syntax

```
response = client.describe_data_sources(
    FilterVariable='CreatedAt'|'LastUpdatedAt'|'Status'|'Name'|'DataLocationS3'|'IAMUser',
    EQ='string',
    GT='string',
    LT='string',
    GE='string',
    LE='string',
    NE='string',
    Prefix='string',
    SortOrder='asc'|'dsc',
    NextToken='string',
    Limit=123
)
```

Parameters

- **FilterVariable** (*string*) – Use one of the following variables to filter a list of `DataSource`:

- **CreatedAt** - Sets the search criteria to `DataSource` creation dates.
- **Status** - Sets the search criteria to `DataSource` statuses.
- **Name** - Sets the search criteria to the contents of `DataSource` `Name`.
- **DataUri** - Sets the search criteria to the URI of data files used to create the `DataSource`. The URI can identify either a file or an Amazon Simple Storage Service (Amazon S3) bucket or directory.
- **IAMUser** - Sets the search criteria to the user account that invoked the `DataSource` creation.
- **EQ** (*string*) - The equal to operator. The `DataSource` results will have `FilterVariable` values that exactly match the value specified with `EQ`.
- **GT** (*string*) - The greater than operator. The `DataSource` results will have `FilterVariable` values that are greater than the value specified with `GT`.
- **LT** (*string*) - The less than operator. The `DataSource` results will have `FilterVariable` values that are less than the value specified with `LT`.
- **GE** (*string*) - The greater than or equal to operator. The `DataSource` results will have `FilterVariable` values that are greater than or equal to the value specified with `GE`.
- **LE** (*string*) - The less than or equal to operator. The `DataSource` results will have `FilterVariable` values that are less than or equal to the value specified with `LE`.
- **NE** (*string*) - The not equal to operator. The `DataSource` results will have `FilterVariable` values not equal to the value specified with `NE`.
- **Prefix** (*string*) - A string that is found at the beginning of a variable, such as `Name` or `Id`.

For example, a `DataSource` could have the `Name` `2014-09-09-HolidayGiftMailer`. To search for this `DataSource`, select `Name` for the `FilterVariable` and any of the following strings for the `Prefix`:

- `2014-09`
- `2014-09-09`
- `2014-09-09-Holiday`
- **SortOrder** (*string*) - A two-value parameter that determines the sequence of the resulting list of `DataSource`.
 - `asc` - Arranges the list in ascending order (A-Z, 0-9).
 - `dsc` - Arranges the list in descending order (Z-A, 9-0).

Results are sorted by `FilterVariable`.

- **NextToken** (*string*) - The ID of the page in the paginated results.
- **Limit** (*integer*) - The maximum number of `DataSource` to include in the result.

Return type dict

Returns

Response Syntax

```
{
  'Results': [
    {
      'DataSourceId': 'string',
      'DataLocationS3': 'string',
      'DataRearrangement': 'string',
      'CreatedByIamUser': 'string',
      'CreatedAt': datetime(2015, 1, 1),
      'LastUpdatedAt': datetime(2015, 1, 1),
    }
  ]
}
```

```
    'DataSizeInBytes': 123,
    'NumberOfFiles': 123,
    'Name': 'string',
    'Status': 'PENDING'|'INPROGRESS'|'FAILED'|'COMPLETED'|'DELETED',
    'Message': 'string',
    'RedshiftMetadata': {
        'RedshiftDatabase': {
            'DatabaseName': 'string',
            'ClusterIdentifier': 'string'
        },
        'DatabaseUserName': 'string',
        'SelectSqlQuery': 'string'
    },
    'RDSMetadata': {
        'Database': {
            'InstanceIdentifier': 'string',
            'DatabaseName': 'string'
        },
        'DatabaseUserName': 'string',
        'SelectSqlQuery': 'string',
        'ResourceRole': 'string',
        'ServiceRole': 'string',
        'DataPipelineId': 'string'
    },
    'RoleARN': 'string',
    'ComputeStatistics': True|False
},
],
'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents the query results from a `DescribeDataSources` operation. The content is essentially a list of `DataSource`.

- **Results** (*list*) –

A list of `DataSource` that meet the search criteria.

- * (*dict*) –

Represents the output of the `GetDataSource` operation.

The content consists of the detailed metadata and data file information and the current status of the `DataSource`.

- **DataSourceId** (*string*) –

The ID that is assigned to the `DataSource` during creation.

- **DataLocationS3** (*string*) –

The location and name of the data in Amazon Simple Storage Service (Amazon S3) that is used by a `DataSource`.

- **DataRearrangement** (*string*) –

A JSON string that represents the splitting requirement of a `Datasource`.

- **CreatedByIamUser** (*string*) –

The AWS user account from which the `DataSource` was created. The account type can be either an AWS root account

or an AWS Identity and Access Management (IAM) user account.

- **CreatedAt** (*datetime*) –

The time that the `DataSource` was created. The time is expressed in epoch time.

- **LastUpdatedAt** (*datetime*) –

The time of the most recent edit to the `BatchPrediction`. The time is expressed in epoch time.

- **DataSizeInBytes** (*integer*) –

The total number of observations contained in the data files that the `DataSource` references.

- **NumberOfFiles** (*integer*) –

The number of data files referenced by the `DataSource`.

- **Name** (*string*) –

A user-supplied name or description of the `DataSource`.

- **Status** (*string*) –

The current status of the `DataSource`. This element can have one of the following values:

- **PENDING** - Amazon Machine Learning (Amazon ML) submitted a request to create a `DataSource`.
- **INPROGRESS** - The creation process is underway.
- **FAILED** - The request to create a `DataSource` did not run to completion. It is not usable.
- **COMPLETED** - The creation process completed successfully.
- **DELETED** - The `DataSource` is marked as deleted. It is not usable.

- **Message** (*string*) –

A description of the most recent details about creating the `DataSource`.

- **RedshiftMetadata** (*dict*) –

Describes the `DataSource` details specific to Amazon Redshift.

- **RedshiftDatabase** (*dict*) –

Describes the database details required to connect to an Amazon Redshift database.

- **DatabaseName** (*string*) –

The name of a database hosted on an Amazon Redshift cluster.

- **ClusterIdentifier** (*string*) –

The ID of an Amazon Redshift cluster.

- **DatabaseUserName** (*string*) –

A username to be used by Amazon Machine Learning (Amazon ML) to connect to a database on an Amazon Redshift cluster. The username should have sufficient permissions to execute the `RedshiftSelectSqlQuery` query. The username should be valid for an Amazon Redshift [USER](#).

- **SelectSqlQuery** (*string*) –

The SQL query that is specified during `CreateDataSourceFromRedshift`. Returns only if `Verbose` is true in `GetDataSourceInput`.

- **RDSMetadata** (*dict*) –

The datasource details that are specific to Amazon RDS.

- **Database** (*dict*) –

The database details required to connect to an Amazon RDS.

- **InstanceIdentifier** (*string*) –

The ID of an RDS DB instance.

- **DatabaseName** (*string*) –

The name of a database hosted on an RDS DB instance.

- **DatabaseUserName** (*string*) –

The username to be used by Amazon ML to connect to database on an Amazon RDS instance. The username should have sufficient permissions to execute an `RDSSelectSqlQuery` query.

- **SelectSqlQuery** (*string*) –

The SQL query that is supplied during `CreateDataSourceFromRDS`. Returns only if `Verbose` is true in `GetDataSourceInput`.

- **ResourceRole** (*string*) –

The role (`DataPipelineDefaultResourceRole`) assumed by an Amazon EC2 instance to carry out the copy task from Amazon RDS to Amazon S3. For more information, see [Role templates](#) for data pipelines.

- **ServiceRole** (*string*) –

The role (`DataPipelineDefaultRole`) assumed by the Data Pipeline service to monitor the progress of the copy task from Amazon RDS to Amazon S3. For more information, see [Role templates](#) for data pipelines.

- **DataPipelineId** (*string*) –

The ID of the Data Pipeline instance that is used to carry to copy data from Amazon RDS to Amazon S3. You can use the ID to find details about the instance in the Data Pipeline console.

- **RoleARN** (*string*) –

The Amazon Resource Name (ARN) of an [AWS IAM Role](#) such as the following: `arn:aws:iam::account:role/rolename`.

- **ComputeStatistics** (*boolean*) –

The parameter is `true` if statistics need to be generated from the observation data.

- **NextToken** (*string*) –

An ID of the next page in the paginated results that indicates at least one more page follows.

describe_evaluations (**kwargs)

Returns a list of `DescribeEvaluations` that match the search criteria in the request.

Request Syntax

```
response = client.describe_evaluations(
    FilterVariable='CreatedAt'|'LastUpdatedAt'|'Status'|'Name'|'IAMUser'|'MLModelId'|'DataSourceId'|'DataUri',
    EQ='string',
    GT='string',
    LT='string',
    GE='string',
    LE='string',
    NE='string',
    Prefix='string',
    SortOrder='asc'|'desc',
    NextToken='string',
    Limit=123
)
```

Parameters

- **FilterVariable** (*string*) – Use one of the following variable to filter a list of Evaluation objects:
 - CreatedAt - Sets the search criteria to the Evaluation creation date.
 - Status - Sets the search criteria to the Evaluation status.
 - Name - Sets the search criteria to the contents of Evaluation **Name**.
 - IAMUser - Sets the search criteria to the user account that invoked an Evaluation.
 - MLModelId - Sets the search criteria to the MLModel that was evaluated.
 - DataSourceId - Sets the search criteria to the DataSource used in Evaluation.
 - DataUri - Sets the search criteria to the data file(s) used in Evaluation. The URL can identify either a file or an Amazon Simple Storage Solution (Amazon S3) bucket or directory.
- **EQ** (*string*) – The equal to operator. The Evaluation results will have FilterVariable values that exactly match the value specified with EQ.
- **GT** (*string*) – The greater than operator. The Evaluation results will have FilterVariable values that are greater than the value specified with GT.
- **LT** (*string*) – The less than operator. The Evaluation results will have FilterVariable values that are less than the value specified with LT.
- **GE** (*string*) – The greater than or equal to operator. The Evaluation results will have FilterVariable values that are greater than or equal to the value specified with GE.
- **LE** (*string*) – The less than or equal to operator. The Evaluation results will have FilterVariable values that are less than or equal to the value specified with LE.
- **NE** (*string*) – The not equal to operator. The Evaluation results will have FilterVariable values not equal to the value specified with NE.
- **Prefix** (*string*) – A string that is found at the beginning of a variable, such as Name or Id.

For example, an Evaluation could have the Name 2014-09-09-HolidayGiftMailer. To search for this Evaluation, select Name for the FilterVariable and any of the following strings for the Prefix:

- 2014-09
- 2014-09-09
- 2014-09-09-Holiday

- **SortOrder** (*string*) – A two-value parameter that determines the sequence of the resulting list of `Evaluation`.
 - `asc` - Arranges the list in ascending order (A-Z, 0-9).
 - `dsc` - Arranges the list in descending order (Z-A, 9-0).Results are sorted by `FilterVariable`.
- **NextToken** (*string*) – The ID of the page in the paginated results.
- **Limit** (*integer*) – The maximum number of `Evaluation` to include in the result.

Return type dict

Returns

Response Syntax

```
{
  'Results': [
    {
      'EvaluationId': 'string',
      'MLModelId': 'string',
      'EvaluationDataSourceId': 'string',
      'InputDataLocationS3': 'string',
      'CreatedByIamUser': 'string',
      'CreatedAt': datetime(2015, 1, 1),
      'LastUpdatedAt': datetime(2015, 1, 1),
      'Name': 'string',
      'Status': 'PENDING'|'INPROGRESS'|'FAILED'|'COMPLETED'|'DELETED',
      'PerformanceMetrics': {
        'Properties': {
          'string': 'string'
        }
      },
      'Message': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents the query results from a `DescribeEvaluations` operation. The content is essentially a list of `Evaluation`.

- **Results** (*list*) –

A list of `Evaluation` that meet the search criteria.

- * (*dict*) –

Represents the output of `GetEvaluation` operation.

The content consists of the detailed metadata and data file information and the current status of the `Evaluation`.

- **EvaluationId** (*string*) –

The ID that is assigned to the `Evaluation` at creation.

- **MLModelId** (*string*) –

The ID of the `MLModel` that is the focus of the evaluation.

- **EvaluationDataSourceId** (*string*) –

The ID of the `DataSource` that is used to evaluate the `MLModel`.

- **InputDataLocationS3** (*string*) –

The location and name of the data in Amazon Simple Storage Server (Amazon S3) that is used in the evaluation.

- **CreatedByIamUser** (*string*) –

The AWS user account that invoked the evaluation. The account type can be either an AWS root account or an AWS Identity and Access Management (IAM) user account.

- **CreatedAt** (*datetime*) –

The time that the `Evaluation` was created. The time is expressed in epoch time.

- **LastUpdatedAt** (*datetime*) –

The time of the most recent edit to the `Evaluation`. The time is expressed in epoch time.

- **Name** (*string*) –

A user-supplied name or description of the `Evaluation`.

- **Status** (*string*) –

The status of the evaluation. This element can have one of the following values:

- `PENDING` - Amazon Machine Learning (Amazon ML) submitted a request to evaluate an `MLModel`.
- `INPROGRESS` - The evaluation is underway.
- `FAILED` - The request to evaluate an `MLModel` did not run to completion. It is not usable.
- `COMPLETED` - The evaluation process completed successfully.
- `DELETED` - The `Evaluation` is marked as deleted. It is not usable.

- **PerformanceMetrics** (*dict*) –

Measurements of how well the `MLModel` performed, using observations referenced by the `DataSource`. One of the following metrics is returned, based on the type of the `MLModel`:

- `BinaryAUC`: A binary `MLModel` uses the Area Under the Curve (AUC) technique to measure performance.
- `RegressionRMSE`: A regression `MLModel` uses the Root Mean Square Error (RMSE) technique to measure performance. RMSE measures the difference between predicted and actual values for a single variable.
- `MulticlassAvgFScore`: A multiclass `MLModel` uses the F1 score technique to measure performance.

For more information about performance metrics, please see the [Amazon Machine Learning Developer Guide](#).

- **Properties** (*dict*) –

- (*string*) –

- (*string*) –

- **Message** (*string*) –

A description of the most recent details about evaluating the `MLModel`.

- **NextToken** (*string*) –

The ID of the next page in the paginated results that indicates at least one more page follows.

describe_ml_models (***kwargs*)

Returns a list of `MLModel` that match the search criteria in the request.

Request Syntax

```
response = client.describe_ml_models(
    FilterVariable='CreatedAt'|'LastUpdatedAt'|'Status'|'Name'|'IAMUser'|'TrainingDataSourceId',
    EQ='string',
    GT='string',
    LT='string',
    GE='string',
    LE='string',
    NE='string',
    Prefix='string',
    SortOrder='asc'|'desc',
    NextToken='string',
    Limit=123
)
```

Parameters

- **FilterVariable** (*string*) – Use one of the following variables to filter a list of `MLModel` :
 - `CreatedAt` - Sets the search criteria to `MLModel` creation date.
 - `Status` - Sets the search criteria to `MLModel` status.
 - `Name` - Sets the search criteria to the contents of `MLModel` ******* `Name` .
 - `IAMUser` - Sets the search criteria to the user account that invoked the `MLModel` creation.
 - `TrainingDataSourceId` - Sets the search criteria to the `DataSource` used to train one or more `MLModel` .
 - `RealtimeEndpointStatus` - Sets the search criteria to the `MLModel` real-time endpoint status.
 - `MLModelType` - Sets the search criteria to `MLModel` type: binary, regression, or multi-class.
 - `Algorithm` - Sets the search criteria to the algorithm that the `MLModel` uses.
 - `TrainingDataURI` - Sets the search criteria to the data file(s) used in training a `MLModel` . The URL can identify either a file or an Amazon Simple Storage Service (Amazon S3) bucket or directory.
- **EQ** (*string*) – The equal to operator. The `MLModel` results will have `FilterVariable` values that exactly match the value specified with `EQ` .
- **GT** (*string*) – The greater than operator. The `MLModel` results will have `FilterVariable` values that are greater than the value specified with `GT` .
- **LT** (*string*) – The less than operator. The `MLModel` results will have `FilterVariable` values that are less than the value specified with `LT` .
- **GE** (*string*) – The greater than or equal to operator. The `MLModel` results will have `FilterVariable` values that are greater than or equal to the value specified with `GE` .
- **LE** (*string*) – The less than or equal to operator. The `MLModel` results will have `FilterVariable` values that are less than or equal to the value specified with `LE` .
- **NE** (*string*) – The not equal to operator. The `MLModel` results will have `FilterVariable` values not equal to the value specified with `NE` .

- **Prefix** (*string*) – A string that is found at the beginning of a variable, such as Name or Id.

For example, an `MLModel` could have the Name `2014-09-09-HolidayGiftMailer`. To search for this `MLModel`, select Name for the `FilterVariable` and any of the following strings for the `Prefix`:

- `2014-09`
- `2014-09-09`
- `2014-09-09-Holiday`

- **SortOrder** (*string*) – A two-value parameter that determines the sequence of the resulting list of `MLModel`.

- `asc` - Arranges the list in ascending order (A-Z, 0-9).
- `dsc` - Arranges the list in descending order (Z-A, 9-0).

Results are sorted by `FilterVariable`.

- **NextToken** (*string*) – The ID of the page in the paginated results.
- **Limit** (*integer*) – The number of pages of information to include in the result. The range of acceptable values is 1 through 100. The default value is 100.

Return type dict

Returns

Response Syntax

```
{
  'Results': [
    {
      'MLModelId': 'string',
      'TrainingDataSourceId': 'string',
      'CreatedByIamUser': 'string',
      'CreatedAt': datetime(2015, 1, 1),
      'LastUpdatedAt': datetime(2015, 1, 1),
      'Name': 'string',
      'Status': 'PENDING'|'INPROGRESS'|'FAILED'|'COMPLETED'|'DELETED',
      'SizeInBytes': 123,
      'EndpointInfo': {
        'PeakRequestsPerSecond': 123,
        'CreatedAt': datetime(2015, 1, 1),
        'EndpointUrl': 'string',
        'EndpointStatus': 'NONE'|'READY'|'UPDATING'|'FAILED'
      },
      'TrainingParameters': {
        'string': 'string'
      },
      'InputDataLocationS3': 'string',
      'Algorithm': 'sgd',
      'MLModelType': 'REGRESSION'|'BINARY'|'MULTICLASS',
      'ScoreThreshold': ...,
      'ScoreThresholdLastUpdatedAt': datetime(2015, 1, 1),
      'Message': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a `DescribeMLModels` operation. The content is essen-

tially a list of `MLModel` .

– **Results** (*list*) –

A list of `MLModel` that meet the search criteria.

* (*dict*) –

Represents the output of a `GetMLModel` operation.

The content consists of the detailed metadata and the current status of the `MLModel` .

· **MLModelId** (*string*) –

The ID assigned to the `MLModel` at creation.

· **TrainingDataSourceId** (*string*) –

The ID of the training `DataSource` . The `CreateMLModel` operation uses the `TrainingDataSourceId` .

· **CreatedByIamUser** (*string*) –

The AWS user account from which the `MLModel` was created. The account type can be either an AWS root account or an AWS Identity and Access Management (IAM) user account.

· **CreatedAt** (*datetime*) –

The time that the `MLModel` was created. The time is expressed in epoch time.

· **LastUpdatedAt** (*datetime*) –

The time of the most recent edit to the `MLModel` . The time is expressed in epoch time.

· **Name** (*string*) –

A user-supplied name or description of the `MLModel` .

· **Status** (*string*) –

The current status of an `MLModel` . This element can have one of the following values:

- **PENDING** - Amazon Machine Learning (Amazon ML) submitted a request to create an `MLModel` .
- **INPROGRESS** - The creation process is underway.
- **FAILED** - The request to create an `MLModel` did not run to completion. It is not usable.
- **COMPLETED** - The creation process completed successfully.
- **DELETED** - The `MLModel` is marked as deleted. It is not usable.

· **SizeInBytes** (*integer*) –

Long integer type that is a 64-bit signed number.

· **EndpointInfo** (*dict*) –

The current endpoint of the `MLModel` .

· **PeakRequestsPerSecond** (*integer*) –

The maximum processing rate for the real-time endpoint for `MLModel` , measured in incoming requests per second.

· **CreatedAt** (*datetime*) –

The time that the request to create the real-time endpoint for the `MLModel` was received. The time is expressed in epoch

time.

- **EndpointUrl** (*string*) –

The URI that specifies where to send real-time prediction requests for the `MLModel`.

Note: Note

The application must wait until the real-time endpoint is ready before using this URI.

- **EndpointStatus** (*string*) –

The current status of the real-time endpoint for the `MLModel`. This element can have one of the following values:

- NONE - Endpoint does not exist or was previously deleted.
- READY - Endpoint is ready to be used for real-time predictions.
- UPDATING - Updating/creating the endpoint.
- **TrainingParameters** (*dict*) –

A list of the training parameters in the `MLModel`. The list is implemented as a map of key/value pairs.

The following is the current set of training parameters:

- `sgd.l1RegularizationAmount` - Coefficient regularization L1 norm. It controls overfitting the data by penalizing large coefficients. This tends to drive coefficients to zero, resulting in a sparse feature set. If you use this parameter, specify a small value, such as 1.0E-04 or 1.0E-08. The value is a double that ranges from 0 to MAX_DOUBLE. The default is not to use L1 normalization. The parameter cannot be used when L2 is specified. Use this parameter sparingly.
- `sgd.l2RegularizationAmount` - Coefficient regularization L2 norm. It controls overfitting the data by penalizing large coefficients. This tends to drive coefficients to small, nonzero values. If you use this parameter, specify a small value, such as 1.0E-04 or 1.0E-08. The value is a double that ranges from 0 to MAX_DOUBLE. The default is not to use L2 normalization. This cannot be used when L1 is specified. Use this parameter sparingly.
- `sgd.maxPasses` - Number of times that the training process traverses the observations to build the `MLModel`. The value is an integer that ranges from 1 to 10000. The default value is 10.
- `sgd.maxMLModelSizeInBytes` - Maximum allowed size of the model. Depending on the input data, the model size might affect performance. The value is an integer that ranges from 100000 to 2147483648. The default value is 33554432.
- (*string*) –

String type.

- (*string*) –

String type.

- **InputDataLocationS3** (*string*) –

The location of the data file or directory in Amazon Simple

Storage Service (Amazon S3).

- **Algorithm** (*string*) –

The algorithm used to train the `MLModel`. The following algorithm is supported:

- **SGD** – Stochastic gradient descent. The goal of SGD is to minimize the gradient of the loss function.

- **MLModelType** (*string*) –

Identifies the `MLModel` category. The following are the available types:

- **REGRESSION** - Produces a numeric result. For example, “What listing price should a house have?”.

- **BINARY** - Produces one of two possible results. For example, “Is this a child-friendly web site?”.

- **MULTICLASS** - Produces more than two possible results. For example, “Is this a **HIGH**, **LOW** or **MEDIUM** risk trade?”.

- **ScoreThreshold** (*float*) –

- **ScoreThresholdLastUpdatedAt** (*datetime*) –

The time of the most recent edit to the `ScoreThreshold`. The time is expressed in epoch time.

- **Message** (*string*) –

A description of the most recent details about accessing the `MLModel`.

- **NextToken** (*string*) –

The ID of the next page in the paginated results that indicates at least one more page follows.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method’s model.

Returns The presigned url

get_batch_prediction (***kwargs*)

Returns a `BatchPrediction` that includes detailed metadata, status, and data file information for a Batch Prediction request.

Request Syntax

```
response = client.get_batch_prediction(
    BatchPredictionId='string'
)
```

Parameters **BatchPredictionId** (*string*) – **[REQUIRED]**

An ID assigned to the `BatchPrediction` at creation.

Return type dict

Returns

Response Syntax

```
{
  'BatchPredictionId': 'string',
  'MLModelId': 'string',
  'BatchPredictionDataSourceId': 'string',
  'InputDataLocationS3': 'string',
  'CreatedByIamUser': 'string',
  'CreatedAt': datetime(2015, 1, 1),
  'LastUpdatedAt': datetime(2015, 1, 1),
  'Name': 'string',
  'Status': 'PENDING'|'INPROGRESS'|'FAILED'|'COMPLETED'|'DELETED',
  'OutputUri': 'string',
  'LogUri': 'string',
  'Message': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a `GetBatchPrediction` operation and describes a `BatchPrediction`.

- **BatchPredictionId** (*string*) –

An ID assigned to the `BatchPrediction` at creation. This value should be identical to the value of the `BatchPredictionID` in the request.

- **MLModelId** (*string*) –

The ID of the `MLModel` that generated predictions for the `BatchPrediction` request.

- **BatchPredictionDataSourceId** (*string*) –

The ID of the `DataSource` that was used to create the `BatchPrediction`.

- **InputDataLocationS3** (*string*) –

The location of the data file or directory in Amazon Simple Storage Service (Amazon S3).

- **CreatedByIamUser** (*string*) –

The AWS user account that invoked the `BatchPrediction`. The account type can be either an AWS root account or an AWS Identity and Access Management (IAM) user account.

- **CreatedAt** (*datetime*) –

The time when the `BatchPrediction` was created. The time is expressed in epoch time.

- **LastUpdatedAt** (*datetime*) –

The time of the most recent edit to `BatchPrediction`. The time is expressed in epoch time.

- **Name** (*string*) –

A user-supplied name or description of the `BatchPrediction`.

- **Status** (*string*) –

The status of the `BatchPrediction`, which can be one of the following values:

- * **PENDING** - Amazon Machine Learning (Amazon ML) submitted a request to generate batch predictions.
- * **INPROGRESS** - The batch predictions are in progress.
- * **FAILED** - The request to perform a batch prediction did not run to completion. It is not usable.
- * **COMPLETED** - The batch prediction process completed successfully.
- * **DELETED** - The `BatchPrediction` is marked as deleted. It is not usable.

– **OutputUri** (*string*) –

The location of an Amazon S3 bucket or directory to receive the operation results.

– **LogUri** (*string*) –

A link to the file that contains logs of the `CreateBatchPrediction` operation.

– **Message** (*string*) –

A description of the most recent details about processing the batch prediction request.

get_data_source (***kwargs*)

Returns a `DataSource` that includes metadata and data file information, as well as the current status of the `DataSource`.

`GetDataSource` provides results in normal or verbose format. The verbose format adds the schema description and the list of files pointed to by the `DataSource` to the normal format.

Request Syntax

```
response = client.get_data_source(  
    DataSourceId='string',  
    Verbose=True|False  
)
```

Parameters

• **DataSourceId** (*string*) – **[REQUIRED]**

The ID assigned to the `DataSource` at creation.

• **Verbose** (*boolean*) – Specifies whether the `GetDataSource` operation should return `DataSourceSchema`.

If true, `DataSourceSchema` is returned.

If false, `DataSourceSchema` is not returned.

Return type dict

Returns

Response Syntax

```
{  
    'DataSourceId': 'string',  
    'DataLocationS3': 'string',  
    'DataRearrangement': 'string',  
    'CreatedByIamUser': 'string',  
    'CreatedAt': datetime(2015, 1, 1),  
    'LastUpdatedAt': datetime(2015, 1, 1),  
    'DataSizeInBytes': 123,  
    'NumberOfFiles': 123,  
    'Name': 'string',
```

```

'Status': 'PENDING'|'INPROGRESS'|'FAILED'|'COMPLETED'|'DELETED',
'LogUri': 'string',
'Message': 'string',
'RedshiftMetadata': {
    'RedshiftDatabase': {
        'DatabaseName': 'string',
        'ClusterIdentifier': 'string'
    },
    'DatabaseUserName': 'string',
    'SelectSqlQuery': 'string'
},
'RDSMetadata': {
    'Database': {
        'InstanceIdentifier': 'string',
        'DatabaseName': 'string'
    },
    'DatabaseUserName': 'string',
    'SelectSqlQuery': 'string',
    'ResourceRole': 'string',
    'ServiceRole': 'string',
    'DataPipelineId': 'string'
},
'RoleARN': 'string',
'ComputeStatistics': True|False,
'DataSourceSchema': 'string'
}

```

Response Structure

- (*dict*) –

Represents the output of a `GetDataSource` operation and describes a `DataSource`.

- **DataSourceId** (*string*) –

The ID assigned to the `DataSource` at creation. This value should be identical to the value of the `DataSourceId` in the request.

- **DataLocationS3** (*string*) –

The location of the data file or directory in Amazon Simple Storage Service (Amazon S3).

- **DataRearrangement** (*string*) –

A JSON string that captures the splitting rearrangement requirement of the `DataSource`.

- **CreatedByIamUser** (*string*) –

The AWS user account from which the `DataSource` was created. The account type can be either an AWS root account or an AWS Identity and Access Management (IAM) user account.

- **CreatedAt** (*datetime*) –

The time that the `DataSource` was created. The time is expressed in epoch time.

- **LastUpdatedAt** (*datetime*) –

The time of the most recent edit to the `DataSource`. The time is expressed in epoch time.

- **DataSourceSizeInBytes** (*integer*) –

The total size of observations in the data files.

- **NumberOfFiles** (*integer*) –

The number of data files referenced by the `DataSource`.

- **Name** (*string*) –

A user-supplied name or description of the `DataSource`.

- **Status** (*string*) –

The current status of the `DataSource`. This element can have one of the following values:

- * **PENDING** - Amazon Machine Language (Amazon ML) submitted a request to create a `DataSource`.
- * **INPROGRESS** - The creation process is underway.
- * **FAILED** - The request to create a `DataSource` did not run to completion. It is not usable.
- * **COMPLETED** - The creation process completed successfully.
- * **DELETED** - The `DataSource` is marked as deleted. It is not usable.

- **LogUri** (*string*) –

A link to the file containing logs of either create `DataSource` operation.

- **Message** (*string*) –

The description of the most recent details about creating the `DataSource`.

- **RedshiftMetadata** (*dict*) –

Describes the `DataSource` details specific to Amazon Redshift.

- * **RedshiftDatabase** (*dict*) –

Describes the database details required to connect to an Amazon Redshift database.

- **DatabaseName** (*string*) –

The name of a database hosted on an Amazon Redshift cluster.

- **ClusterIdentifier** (*string*) –

The ID of an Amazon Redshift cluster.

- * **DatabaseUserName** (*string*) –

A username to be used by Amazon Machine Learning (Amazon ML) to connect to a database on an Amazon Redshift cluster. The username should have sufficient permissions to execute the `RedshiftSelectSqlQuery` query. The username should be valid for an Amazon Redshift `USER`.

- * **SelectSqlQuery** (*string*) –

The SQL query that is specified during `CreateDataSourceFromRedshift`. Returns only if `Verbose` is true in `GetDataSourceInput`.

- **RDSMetadata** (*dict*) –

The datasource details that are specific to Amazon RDS.

- * **Database** (*dict*) –

The database details required to connect to an Amazon RDS.

- **InstanceIdentifier** (*string*) –

The ID of an RDS DB instance.

- **DatabaseName** (*string*) –
The name of a database hosted on an RDS DB instance.
- * **DatabaseUserName** (*string*) –
The username to be used by Amazon ML to connect to database on an Amazon RDS instance. The username should have sufficient permissions to execute an `RDSSelectSqlQuery` query.
- * **SelectSqlQuery** (*string*) –
The SQL query that is supplied during `CreateDataSourceFromRDS`. Returns only if `Verbose` is `true` in `GetDataSourceInput`.
- * **ResourceRole** (*string*) –
The role (`DataPipelineDefaultResourceRole`) assumed by an Amazon EC2 instance to carry out the copy task from Amazon RDS to Amazon S3. For more information, see [Role templates](#) for data pipelines.
- * **ServiceRole** (*string*) –
The role (`DataPipelineDefaultRole`) assumed by the Data Pipeline service to monitor the progress of the copy task from Amazon RDS to Amazon S3. For more information, see [Role templates](#) for data pipelines.
- * **DataPipelineId** (*string*) –
The ID of the Data Pipeline instance that is used to carry to copy data from Amazon RDS to Amazon S3. You can use the ID to find details about the instance in the Data Pipeline console.
- **RoleARN** (*string*) –
The Amazon Resource Name (ARN) of an [AWS IAM Role](#) such as the following: `arn:aws:iam::account:role/rolename`.
- **ComputeStatistics** (*boolean*) –
The parameter is `true` if statistics need to be generated from the observation data.
- **DataSourceSchema** (*string*) –
The schema used by all of the data files of this `DataSource`.

Note: Note

This parameter is provided as part of the verbose format.

get_evaluation (***kwargs*)

Returns an `Evaluation` that includes metadata as well as the current status of the `Evaluation`.

Request Syntax

```
response = client.get_evaluation(
    EvaluationId='string'
)
```

Parameters `EvaluationId` (*string*) – [REQUIRED]

The ID of the `Evaluation` to retrieve. The evaluation of each `MLModel` is recorded and cataloged. The ID provides the means to access the information.

Return type dict

Returns

Response Syntax

```
{
    'EvaluationId': 'string',
    'MLModelId': 'string',
    'EvaluationDataSourceId': 'string',
    'InputDataLocationS3': 'string',
    'CreatedByIamUser': 'string',
    'CreatedAt': datetime(2015, 1, 1),
    'LastUpdatedAt': datetime(2015, 1, 1),
    'Name': 'string',
    'Status': 'PENDING'|'INPROGRESS'|'FAILED'|'COMPLETED'|'DELETED',
    'PerformanceMetrics': {
        'Properties': {
            'string': 'string'
        }
    },
    'LogUri': 'string',
    'Message': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a `GetEvaluation` operation and describes an `Evaluation`.

- **EvaluationId** (*string*) –

The evaluation ID which is same as the `EvaluationId` in the request.

- **MLModelId** (*string*) –

The ID of the `MLModel` that was the focus of the evaluation.

- **EvaluationDataSourceId** (*string*) –

The `DataSource` used for this evaluation.

- **InputDataLocationS3** (*string*) –

The location of the data file or directory in Amazon Simple Storage Service (Amazon S3).

- **CreatedByIamUser** (*string*) –

The AWS user account that invoked the evaluation. The account type can be either an AWS root account or an AWS Identity and Access Management (IAM) user account.

- **CreatedAt** (*datetime*) –

The time that the `Evaluation` was created. The time is expressed in epoch time.

- **LastUpdatedAt** (*datetime*) –

The time of the most recent edit to the `BatchPrediction`. The time is expressed in epoch time.

- **Name** (*string*) –

A user-supplied name or description of the `Evaluation`.

- **Status** (*string*) –

The status of the evaluation. This element can have one of the following values:

- * **PENDING** - Amazon Machine Language (Amazon ML) submitted a request to evaluate an `MLModel` .
- * **INPROGRESS** - The evaluation is underway.
- * **FAILED** - The request to evaluate an `MLModel` did not run to completion. It is not usable.
- * **COMPLETED** - The evaluation process completed successfully.
- * **DELETED** - The Evaluation is marked as deleted. It is not usable.

– **PerformanceMetrics** (*dict*) –

Measurements of how well the `MLModel` performed using observations referenced by the `DataSource` . One of the following metric is returned based on the type of the `MLModel` :

- * **BinaryAUC**: A binary `MLModel` uses the Area Under the Curve (AUC) technique to measure performance.
- * **RegressionRMSE**: A regression `MLModel` uses the Root Mean Square Error (RMSE) technique to measure performance. RMSE measures the difference between predicted and actual values for a single variable.
- * **MulticlassAvgFScore**: A multiclass `MLModel` uses the F1 score technique to measure performance.

For more information about performance metrics, please see the [Amazon Machine Learning Developer Guide](#) .

* **Properties** (*dict*) –

- (*string*) –
- (*string*) –

– **LogUri** (*string*) –

A link to the file that contains logs of the `CreateEvaluation` operation.

– **Message** (*string*) –

A description of the most recent details about evaluating the `MLModel` .

get_ml_model (***kwargs*)

Returns an `MLModel` that includes detailed metadata, and data source information as well as the current status of the `MLModel` .

`GetMLModel` provides results in normal or verbose format.

Request Syntax

```
response = client.get_ml_model(
    MLModelId='string',
    Verbose=True|False
)
```

Parameters

- **MLModelId** (*string*) – **[REQUIRED]**

The ID assigned to the `MLModel` at creation.

- **Verbose** (*boolean*) – Specifies whether the `GetMLModel` operation should return Recipe .

If true, Recipe is returned.

If false, Recipe is not returned.

Return type dict

Returns

Response Syntax

```
{
  'MLModelId': 'string',
  'TrainingDataSourceId': 'string',
  'CreatedByIamUser': 'string',
  'CreatedAt': datetime(2015, 1, 1),
  'LastUpdatedAt': datetime(2015, 1, 1),
  'Name': 'string',
  'Status': 'PENDING' | 'INPROGRESS' | 'FAILED' | 'COMPLETED' | 'DELETED',
  'SizeInBytes': 123,
  'EndpointInfo': {
    'PeakRequestsPerSecond': 123,
    'CreatedAt': datetime(2015, 1, 1),
    'EndpointUrl': 'string',
    'EndpointStatus': 'NONE' | 'READY' | 'UPDATING' | 'FAILED'
  },
  'TrainingParameters': {
    'string': 'string'
  },
  'InputDataLocationS3': 'string',
  'MLModelType': 'REGRESSION' | 'BINARY' | 'MULTICLASS',
  'ScoreThreshold': ...,
  'ScoreThresholdLastUpdatedAt': datetime(2015, 1, 1),
  'LogUri': 'string',
  'Message': 'string',
  'Recipe': 'string',
  'Schema': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of a `GetMLModel` operation, and provides detailed information about a `MLModel`.

- **MLModelId** (*string*) –

The MLModel ID which is same as the `MLModelId` in the request.

- **TrainingDataSourceId** (*string*) –

The ID of the training `DataSource`.

- **CreatedByIamUser** (*string*) –

The AWS user account from which the `MLModel` was created. The account type can be either an AWS root account or an AWS Identity and Access Management (IAM) user account.

- **CreatedAt** (*datetime*) –

The time that the `MLModel` was created. The time is expressed in epoch time.

- **LastUpdatedAt** (*datetime*) –

The time of the most recent edit to the `MLModel`. The time is expressed in epoch time.

- **Name** (*string*) –

A user-supplied name or description of the `MLModel`.

- **Status** (*string*) –

The current status of the `MLModel` . This element can have one of the following values:

- * `PENDING` - Amazon Machine Learning (Amazon ML) submitted a request to describe a `MLModel` .
- * `INPROGRESS` - The request is processing.
- * `FAILED` - The request did not run to completion. It is not usable.
- * `COMPLETED` - The request completed successfully.
- * `DELETED` - The `MLModel` is marked as deleted. It is not usable.

– **SizeInBytes** (*integer*) –

Long integer type that is a 64-bit signed number.

– **EndpointInfo** (*dict*) –

The current endpoint of the `MLModel`

- * **PeakRequestsPerSecond** (*integer*) –

The maximum processing rate for the real-time endpoint for `MLModel` , measured in incoming requests per second.

- * **CreatedAt** (*datetime*) –

The time that the request to create the real-time endpoint for the `MLModel` was received. The time is expressed in epoch time.

- * **EndpointUrl** (*string*) –

The URI that specifies where to send real-time prediction requests for the `MLModel` .

Note: Note

The application must wait until the real-time endpoint is ready before using this URI.

-
- * **EndpointStatus** (*string*) –

The current status of the real-time endpoint for the `MLModel` . This element can have one of the following values:

- `NONE` - Endpoint does not exist or was previously deleted.
- `READY` - Endpoint is ready to be used for real-time predictions.
- `UPDATING` - Updating/creating the endpoint.

– **TrainingParameters** (*dict*) –

A list of the training parameters in the `MLModel` . The list is implemented as a map of key/value pairs.

The following is the current set of training parameters:

- * `sgd.l1RegularizationAmount` - Coefficient regularization L1 norm. It controls overfitting the data by penalizing large coefficients. This tends to drive coefficients to zero, resulting in a sparse feature set. If you use this parameter, specify a small value, such as 1.0E-04 or 1.0E-08. The value is a double that ranges from 0 to `MAX_DOUBLE`. The default is not to use L1 normalization. The parameter cannot be used when L2 is specified. Use this parameter sparingly.
- * `sgd.l2RegularizationAmount` - Coefficient regularization L2 norm. It controls overfitting the data by penalizing large coefficients. This tends to drive coefficients to small, nonzero values. If you use this parameter, specify a small value, such as 1.0E-04 or 1.0E-08. The value is a double that ranges from 0 to

MAX_DOUBLE. The default is not to use L2 normalization. This parameter cannot be used when L1 is specified. Use this parameter sparingly.

- * `sgd.maxPasses` - The number of times that the training process traverses the observations to build the `MLModel`. The value is an integer that ranges from 1 to 10000. The default value is 10.

- * `sgd.maxMLModelSizeInBytes` - The maximum allowed size of the model. Depending on the input data, the model size might affect performance. The value is an integer that ranges from 100000 to 2147483648. The default value is 33554432.

- * *(string)* –

String type.

- *(string)* –

String type.

- **InputDataLocationS3** *(string)* –

The location of the data file or directory in Amazon Simple Storage Service (Amazon S3).

- **MLModelType** *(string)* –

Identifies the `MLModel` category. The following are the available types:

- * REGRESSION – Produces a numeric result. For example, “What listing price should a house have?”

- * BINARY – Produces one of two possible results. For example, “Is this an e-commerce website?”

- * MULTICLASS – Produces more than two possible results. For example, “Is this a HIGH, LOW or MEDIUM risk trade?”

- **ScoreThreshold** *(float)* –

The scoring threshold is used in binary classification `MLModel`s, and marks the boundary between a positive prediction and a negative prediction.

Output values greater than or equal to the threshold receive a positive result from the `MLModel`, such as `true`. Output values less than the threshold receive a negative response from the `MLModel`, such as `false`.

- **ScoreThresholdLastUpdatedAt** *(datetime)* –

The time of the most recent edit to the `ScoreThreshold`. The time is expressed in epoch time.

- **LogUri** *(string)* –

A link to the file that contains logs of the `CreateMLModel` operation.

- **Message** *(string)* –

Description of the most recent details about accessing the `MLModel`.

- **Recipe** *(string)* –

The recipe to use when training the `MLModel`. The `Recipe` provides detailed information about the observation data to use during training, as well as manipulations to perform on the observation data during training.

Note: Note

This parameter is provided as part of the verbose format.

- **Schema** *(string)* –

The schema used by all of the data files referenced by the `DataSource`.

Note: Note

This parameter is provided as part of the verbose format.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises `OperationNotPageableError` Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_waiter (*waiter_name*)

predict (***kwargs*)

Generates a prediction for the observation using the specified ML Model.

Note: Note

Not all response parameters will be populated. Whether a response parameter is populated depends on the type of model requested.

Request Syntax

```
response = client.predict(
    MLModelId='string',
    Record={
        'string': 'string'
    },
    PredictEndpoint='string'
)
```

Parameters

- **MLModelId** (*string*) – [REQUIRED]

A unique identifier of the `MLModel`.

- **Record** (*dict*) – [REQUIRED]

A map of variable name-value pairs that represent an observation.

– (*string*) –

The name of a variable. Currently it's used to specify the name of the target value, label, weight, and tags.

* (*string*) –

The value of a variable. Currently it's used to specify values of the target value, weights, and tag variables and for filtering variable values.

- **PredictEndpoint** (*string*) – [REQUIRED]

Return type `dict`

Returns

Response Syntax

```
{
  'Prediction': {
    'predictedLabel': 'string',
    'predictedValue': ...,
    'predictedScores': {
      'string': ...
    },
    'details': {
      'string': 'string'
    }
  }
}
```

Response Structure

- (*dict*) –
 - **Prediction** (*dict*) –

The output from a `Predict` operation:

- * **Details** - Contains the following attributes: `DetailsAttributes.PREDICTIVE_MODEL_TYPE` - REGRESSION | BINARY | MULTICLASS `DetailsAttributes.ALGORITHM` - SGD
- * **PredictedLabel** - Present for either a BINARY or MULTICLASS `MLModel` request.
- * **PredictedScores** - Contains the raw classification score corresponding to each label.
- * **PredictedValue** - Present for a REGRESSION `MLModel` request.
- * **predictedLabel** (*string*) – The prediction label for either a BINARY or MULTICLASS `MLModel`.
- * **predictedValue** (*float*) – The prediction value for REGRESSION `MLModel`.
- * **predictedScores** (*dict*) – Provides the raw classification score corresponding to each label.
 - (*string*) –
 - (*float*) –
- * **details** (*dict*) – Provides any additional details regarding the prediction.
 - (*string*) – Contains the key values of `DetailsMap`: `PredictiveModelType` - Indicates the type of the `MLModel`. `Algorithm` - Indicates the algorithm was used for the `MLModel`.
 - (*string*) –

`update_batch_prediction` (**kwargs)

Updates the `BatchPredictionName` of a `BatchPrediction`.

You can use the `GetBatchPrediction` operation to view the contents of the updated data element.

Request Syntax

```
response = client.update_batch_prediction(
    BatchPredictionId='string',
    BatchPredictionName='string'
)
```

Parameters

- **BatchPredictionId** (*string*) – [REQUIRED]

The ID assigned to the `BatchPrediction` during creation.

- **BatchPredictionName** (*string*) – [REQUIRED]

A new user-supplied name or description of the `BatchPrediction`.

Return type dict

Returns

Response Syntax

```
{
    'BatchPredictionId': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of an `UpdateBatchPrediction` operation.

You can see the updated content by using the `GetBatchPrediction` operation.

- **BatchPredictionId** (*string*) –

The ID assigned to the `BatchPrediction` during creation. This value should be identical to the value of the `BatchPredictionId` in the request.

update_data_source (***kwargs*)

Updates the `DataSourceName` of a `DataSource`.

You can use the `GetDataSource` operation to view the contents of the updated data element.

Request Syntax

```
response = client.update_data_source(
    DataSourceId='string',
    DataSourceName='string'
)
```

Parameters

- **DataSourceId** (*string*) – [REQUIRED]

The ID assigned to the `DataSource` during creation.

- **DataSourceName** (*string*) – [REQUIRED]

A new user-supplied name or description of the `DataSource` that will replace the current description.

Return type dict

Returns

Response Syntax

```
{
    'DataSourceId': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of an `UpdateDataSource` operation.

You can see the updated content by using the `GetBatchPrediction` operation.

– **DataSourceId** (*string*) –

The ID assigned to the `DataSource` during creation. This value should be identical to the value of the `DataSourceId` in the request.

update_evaluation (***kwargs*)

Updates the `EvaluationName` of an `Evaluation`.

You can use the `GetEvaluation` operation to view the contents of the updated data element.

Request Syntax

```
response = client.update_evaluation(
    EvaluationId='string',
    EvaluationName='string'
)
```

Parameters

- **EvaluationId** (*string*) – [REQUIRED]

The ID assigned to the `Evaluation` during creation.

- **EvaluationName** (*string*) – [REQUIRED]

A new user-supplied name or description of the `Evaluation` that will replace the current content.

Return type dict

Returns

Response Syntax

```
{
    'EvaluationId': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of an `UpdateEvaluation` operation.

You can see the updated content by using the `GetEvaluation` operation.

– **EvaluationId** (*string*) –

The ID assigned to the `Evaluation` during creation. This value should be identical to the value of the `Evaluation` in the request.

update_ml_model (***kwargs*)

Updates the `MLModelName` and the `ScoreThreshold` of an `MLModel`.

You can use the `GetMLModel` operation to view the contents of the updated data element.

Request Syntax

```
response = client.update_ml_model(
    MLModelId='string',
    MLModelName='string',
    ScoreThreshold=...
)
```

Parameters

- **MLModelId** (*string*) – [REQUIRED]

The ID assigned to the `MLModel` during creation.

- **MLModelName** (*string*) – A user-supplied name or description of the `MLModel`.
- **ScoreThreshold** (*float*) – The `ScoreThreshold` used in binary classification `MLModel` that marks the boundary between a positive prediction and a negative prediction.

Output values greater than or equal to the `ScoreThreshold` receive a positive result from the `MLModel`, such as `true`. Output values less than the `ScoreThreshold` receive a negative response from the `MLModel`, such as `false`.

Return type dict

Returns

Response Syntax

```
{
    'MLModelId': 'string'
}
```

Response Structure

- (*dict*) –

Represents the output of an `UpdateMLModel` operation.

You can see the updated content by using the `GetMLModel` operation.

- **MLModelId** (*string*) –

The ID assigned to the `MLModel` during creation. This value should be identical to the value of the `MLModelId` in the request.

Paginators

The available paginators are:

- `MachineLearning.Paginator.DescribeBatchPredictions`
- `MachineLearning.Paginator.DescribeDataSources`
- `MachineLearning.Paginator.DescribeEvaluations`
- `MachineLearning.Paginator.DescribeMLModels`

class `MachineLearning.Paginator.DescribeBatchPredictions`

```
paginator = client.get_paginator('describe_batch_predictions')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `MachineLearning.Client.describe_batch_predictions()`.

Request Syntax

```
response_iterator = paginator.paginate(
    FilterVariable='CreatedAt'|'LastUpdatedAt'|'Status'|'Name'|'IAMUser'|'MLModelId'|'DataS
    EQ='string',
    GT='string',
    LT='string',
    GE='string',
```

```
LE='string',
NE='string',
Prefix='string',
SortOrder='asc'|'dsc',
PaginationConfig={
    'MaxItems': 123,
    'PageSize': 123,
    'StartingToken': 'string'
}
)
```

Parameters

- **FilterVariable** (*string*) – Use one of the following variables to filter a list of BatchPrediction:
 - CreatedAt - Sets the search criteria to the BatchPrediction creation date.
 - Status - Sets the search criteria to the BatchPrediction status.
 - Name - Sets the search criteria to the contents of the BatchPrediction ******* Name.
 - IAMUser - Sets the search criteria to the user account that invoked the BatchPrediction creation.
 - MLModelId - Sets the search criteria to the MLModel used in the BatchPrediction.
 - DataSourceId - Sets the search criteria to the DataSource used in the BatchPrediction.
 - DataURI - Sets the search criteria to the data file(s) used in the BatchPrediction. The URL can identify either a file or an Amazon Simple Storage Solution (Amazon S3) bucket or directory.
- **EQ** (*string*) – The equal to operator. The BatchPrediction results will have FilterVariable values that exactly match the value specified with EQ.
- **GT** (*string*) – The greater than operator. The BatchPrediction results will have FilterVariable values that are greater than the value specified with GT.
- **LT** (*string*) – The less than operator. The BatchPrediction results will have FilterVariable values that are less than the value specified with LT.
- **GE** (*string*) – The greater than or equal to operator. The BatchPrediction results will have FilterVariable values that are greater than or equal to the value specified with GE.
- **LE** (*string*) – The less than or equal to operator. The BatchPrediction results will have FilterVariable values that are less than or equal to the value specified with LE.
- **NE** (*string*) – The not equal to operator. The BatchPrediction results will have FilterVariable values not equal to the value specified with NE.
- **Prefix** (*string*) – A string that is found at the beginning of a variable, such as Name or Id.

For example, a Batch Prediction operation could have the Name 2014-09-09-HolidayGiftMailer. To search for this BatchPrediction, select Name for the FilterVariable and any of the following strings for the Prefix:

- 2014-09
 - 2014-09-09
 - 2014-09-09-Holiday
- **SortOrder** (*string*) – A two-value parameter that determines the sequence of the resulting list of MLModel s.

- `asc` - Arranges the list in ascending order (A-Z, 0-9).
- `dsc` - Arranges the list in descending order (Z-A, 9-0).

Results are sorted by `FilterVariable`.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
  'Results': [
    {
      'BatchPredictionId': 'string',
      'MLModelId': 'string',
      'BatchPredictionDataSourceId': 'string',
      'InputDataLocationS3': 'string',
      'CreatedByIamUser': 'string',
      'CreatedAt': datetime(2015, 1, 1),
      'LastUpdatedAt': datetime(2015, 1, 1),
      'Name': 'string',
      'Status': 'PENDING'|'INPROGRESS'|'FAILED'|'COMPLETED'|'DELETED',
      'OutputUri': 'string',
      'Message': 'string'
    },
  ],
}
```

Response Structure

- (*dict*) –

Represents the output of a `DescribeBatchPredictions` operation. The content is essentially a list of `BatchPrediction`s.

- **Results** (*list*) –

A list of `BatchPrediction` objects that meet the search criteria.

- * (*dict*) –

Represents the output of `GetBatchPrediction` operation.

The content consists of the detailed metadata, the status, and the data file information of a *Batch Prediction*.

- **BatchPredictionId** (*string*) –

The ID assigned to the `BatchPrediction` at creation. This value should be identical to the value of the `BatchPredictionID` in the request.

- **MLModelId** (*string*) –
The ID of the `MLModel` that generated predictions for the `BatchPrediction` request.
- **BatchPredictionDataSourceId** (*string*) –
The ID of the `DataSource` that points to the group of observations to predict.
- **InputDataLocationS3** (*string*) –
The location of the data file or directory in Amazon Simple Storage Service (Amazon S3).
- **CreatedByIamUser** (*string*) –
The AWS user account that invoked the `BatchPrediction`. The account type can be either an AWS root account or an AWS Identity and Access Management (IAM) user account.
- **CreatedAt** (*datetime*) –
The time that the `BatchPrediction` was created. The time is expressed in epoch time.
- **LastUpdatedAt** (*datetime*) –
The time of the most recent edit to the `BatchPrediction`. The time is expressed in epoch time.
- **Name** (*string*) –
A user-supplied name or description of the `BatchPrediction`.
- **Status** (*string*) –
The status of the `BatchPrediction`. This element can have one of the following values:
 - `PENDING` - Amazon Machine Learning (Amazon ML) submitted a request to generate predictions for a batch of observations.
 - `INPROGRESS` - The process is underway.
 - `FAILED` - The request to perform a batch prediction did not run to completion. It is not usable.
 - `COMPLETED` - The batch prediction process completed successfully.
 - `DELETED` - The `BatchPrediction` is marked as deleted. It is not usable.
- **OutputUri** (*string*) –
The location of an Amazon S3 bucket or directory to receive the operation results. The following substrings are not allowed in the s3 key portion of the “outputURI” field: ‘:’, ‘//’, ‘/.’, ‘/./’.
- **Message** (*string*) –
A description of the most recent details about processing the batch prediction request.

```
class MachineLearning.Paginator.DescribeDataSources
```

```
paginator = client.get_paginator('describe_data_sources')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `MachineLearning.Client.describe_data_sources()`.

Request Syntax

```
response_iterator = paginator.paginate(
    FilterVariable='CreatedAt'|'LastUpdatedAt'|'Status'|'Name'|'DataLocationS3'|'IAMUser',
    EQ='string',
    GT='string',
    LT='string',
    GE='string',
    LE='string',
    NE='string',
    Prefix='string',
    SortOrder='asc'|'dsc',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **FilterVariable** (*string*) – Use one of the following variables to filter a list of DataSource:
 - CreatedAt - Sets the search criteria to DataSource creation dates.
 - Status - Sets the search criteria to DataSource statuses.
 - Name - Sets the search criteria to the contents of DataSource ******* Name.
 - DataUri - Sets the search criteria to the URI of data files used to create the DataSource. The URI can identify either a file or an Amazon Simple Storage Service (Amazon S3) bucket or directory.
 - IAMUser - Sets the search criteria to the user account that invoked the DataSource creation.
- **EQ** (*string*) – The equal to operator. The DataSource results will have FilterVariable values that exactly match the value specified with EQ.
- **GT** (*string*) – The greater than operator. The DataSource results will have FilterVariable values that are greater than the value specified with GT.
- **LT** (*string*) – The less than operator. The DataSource results will have FilterVariable values that are less than the value specified with LT.
- **GE** (*string*) – The greater than or equal to operator. The DataSource results will have FilterVariable values that are greater than or equal to the value specified with GE.
- **LE** (*string*) – The less than or equal to operator. The DataSource results will have FilterVariable values that are less than or equal to the value specified with LE.
- **NE** (*string*) – The not equal to operator. The DataSource results will have FilterVariable values not equal to the value specified with NE.
- **Prefix** (*string*) – A string that is found at the beginning of a variable, such as Name or Id.

For example, a DataSource could have the Name 2014-09-09-HolidayGiftMailer. To search for this DataSource, select Name for the FilterVariable and any of the following strings for the Prefix:

- 2014-09
- 2014-09-09
- 2014-09-09-Holiday
- **SortOrder** (*string*) – A two-value parameter that determines the sequence of the resulting list of `DataSource`.
 - `asc` - Arranges the list in ascending order (A-Z, 0-9).
 - `dsc` - Arranges the list in descending order (Z-A, 9-0).Results are sorted by `FilterVariable`.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –

The size of each page.
 - **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
  'Results': [
    {
      'DataSourceId': 'string',
      'DataLocationS3': 'string',
      'DataRearrangement': 'string',
      'CreatedByIamUser': 'string',
      'CreatedAt': datetime(2015, 1, 1),
      'LastUpdatedAt': datetime(2015, 1, 1),
      'DataSizeInBytes': 123,
      'NumberOfFiles': 123,
      'Name': 'string',
      'Status': 'PENDING'|'INPROGRESS'|'FAILED'|'COMPLETED'|'DELETED',
      'Message': 'string',
      'RedshiftMetadata': {
        'RedshiftDatabase': {
          'DatabaseName': 'string',
          'ClusterIdentifier': 'string'
        },
        'DatabaseUserName': 'string',
        'SelectSqlQuery': 'string'
      },
      'RDSMetadata': {
        'Database': {
          'InstanceIdentifier': 'string',
          'DatabaseName': 'string'
        },
        'DatabaseUserName': 'string',
        'SelectSqlQuery': 'string',
        'ResourceRole': 'string',
        'ServiceRole': 'string',
```

```

        'DataPipelineId': 'string'
    },
    'RoleARN': 'string',
    'ComputeStatistics': True|False
},
],
}

```

Response Structure

- (*dict*) –

Represents the query results from a DescribeDataSources operation. The content is essentially a list of DataSource .

- **Results** (*list*) –

A list of DataSource that meet the search criteria.

- * (*dict*) –

Represents the output of the GetDataSource operation.

The content consists of the detailed metadata and data file information and the current status of the DataSource .

- **DataSourceId** (*string*) –

The ID that is assigned to the DataSource during creation.

- **DataLocationS3** (*string*) –

The location and name of the data in Amazon Simple Storage Service (Amazon S3) that is used by a DataSource .

- **DataRearrangement** (*string*) –

A JSON string that represents the splitting requirement of a DataSource .

- **CreatedByIamUser** (*string*) –

The AWS user account from which the DataSource was created. The account type can be either an AWS root account or an AWS Identity and Access Management (IAM) user account.

- **CreatedAt** (*datetime*) –

The time that the DataSource was created. The time is expressed in epoch time.

- **LastUpdatedAt** (*datetime*) –

The time of the most recent edit to the BatchPrediction . The time is expressed in epoch time.

- **DataSourceSizeInBytes** (*integer*) –

The total number of observations contained in the data files that the DataSource references.

- **NumberOfFiles** (*integer*) –

The number of data files referenced by the DataSource .

- **Name** (*string*) –

A user-supplied name or description of the DataSource .

- **Status** (*string*) –

The current status of the `DataSource`. This element can have one of the following values:

- `PENDING` - Amazon Machine Learning (Amazon ML) submitted a request to create a `DataSource`.
- `INPROGRESS` - The creation process is underway.
- `FAILED` - The request to create a `DataSource` did not run to completion. It is not usable.
- `COMPLETED` - The creation process completed successfully.
- `DELETED` - The `DataSource` is marked as deleted. It is not usable.
- **Message** (*string*) –

A description of the most recent details about creating the `DataSource`.

- **RedshiftMetadata** (*dict*) –

Describes the `DataSource` details specific to Amazon Redshift.

- **RedshiftDatabase** (*dict*) –

Describes the database details required to connect to an Amazon Redshift database.

- **DatabaseName** (*string*) –

The name of a database hosted on an Amazon Redshift cluster.

- **ClusterIdentifier** (*string*) –

The ID of an Amazon Redshift cluster.

- **DatabaseUserName** (*string*) –

A username to be used by Amazon Machine Learning (Amazon ML) to connect to a database on an Amazon Redshift cluster. The username should have sufficient permissions to execute the `RedshiftSelectSqlQuery` query. The username should be valid for an Amazon Redshift [USER](#).

- **SelectSqlQuery** (*string*) –

The SQL query that is specified during `CreateDataSourceFromRedshift`. Returns only if `Verbose` is true in `GetDataSourceInput`.

- **RDSMetadata** (*dict*) –

The datasource details that are specific to Amazon RDS.

- **Database** (*dict*) –

The database details required to connect to an Amazon RDS.

- **InstanceIdentifier** (*string*) –

The ID of an RDS DB instance.

- **DatabaseName** (*string*) –

The name of a database hosted on an RDS DB instance.

- **DatabaseUserName** (*string*) –

The username to be used by Amazon ML to connect to database on an Amazon RDS instance. The username should have sufficient permissions to execute an `RDSSelectSqlQuery` query.

- **SelectSqlQuery** (*string*) –

The SQL query that is supplied during `CreateDataSourceFromRDS`. Returns only if `Verbose` is `true` in `GetDataSourceInput`.

- **ResourceRole** (*string*) –

The role (`DataPipelineDefaultResourceRole`) assumed by an Amazon EC2 instance to carry out the copy task from Amazon RDS to Amazon S3. For more information, see [Role templates](#) for data pipelines.

- **ServiceRole** (*string*) –

The role (`DataPipelineDefaultRole`) assumed by the Data Pipeline service to monitor the progress of the copy task from Amazon RDS to Amazon S3. For more information, see [Role templates](#) for data pipelines.

- **DataPipelineId** (*string*) –

The ID of the Data Pipeline instance that is used to carry to copy data from Amazon RDS to Amazon S3. You can use the ID to find details about the instance in the Data Pipeline console.

- **RoleARN** (*string*) –

The Amazon Resource Name (ARN) of an [AWS IAM Role](#) such as the following: `arn:aws:iam::account:role/rolename`.

- **ComputeStatistics** (*boolean*) –

The parameter is `true` if statistics need to be generated from the observation data.

class `MachineLearning.Paginator.DescribeEvaluations`

```
paginator = client.get_paginator('describe_evaluations')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `MachineLearning.Client.describe_evaluations()`.

Request Syntax

```
response_iterator = paginator.paginate(
    FilterVariable='CreatedAt' | 'LastUpdatedAt' | 'Status' | 'Name' | 'IAMUser' | 'MLModelId' | 'DataS
    EQ='string',
    GT='string',
    LT='string',
    GE='string',
    LE='string',
    NE='string',
    Prefix='string',
    SortOrder='asc' | 'desc',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **FilterVariable** (*string*) – Use one of the following variable to filter a list of Evaluation objects:
 - **CreatedAt** - Sets the search criteria to the Evaluation creation date.
 - **Status** - Sets the search criteria to the Evaluation status.
 - **Name** - Sets the search criteria to the contents of Evaluation ******* Name .
 - **IAMUser** - Sets the search criteria to the user account that invoked an Evaluation .
 - **MLModelId** - Sets the search criteria to the MLModel that was evaluated.
 - **DataSourceId** - Sets the search criteria to the DataSource used in Evaluation .
 - **DataUri** - Sets the search criteria to the data file(s) used in Evaluation . The URL can identify either a file or an Amazon Simple Storage Solution (Amazon S3) bucket or directory.
- **EQ** (*string*) – The equal to operator. The Evaluation results will have FilterVariable values that exactly match the value specified with EQ .
- **GT** (*string*) – The greater than operator. The Evaluation results will have FilterVariable values that are greater than the value specified with GT .
- **LT** (*string*) – The less than operator. The Evaluation results will have FilterVariable values that are less than the value specified with LT .
- **GE** (*string*) – The greater than or equal to operator. The Evaluation results will have FilterVariable values that are greater than or equal to the value specified with GE .
- **LE** (*string*) – The less than or equal to operator. The Evaluation results will have FilterVariable values that are less than or equal to the value specified with LE .
- **NE** (*string*) – The not equal to operator. The Evaluation results will have FilterVariable values not equal to the value specified with NE .
- **Prefix** (*string*) – A string that is found at the beginning of a variable, such as Name or Id .

For example, an Evaluation could have the Name 2014-09-09-HolidayGiftMailer . To search for this Evaluation , select Name for the FilterVariable and any of the following strings for the Prefix :

- 2014-09
 - 2014-09-09
 - 2014-09-09-Holiday
- **SortOrder** (*string*) – A two-value parameter that determines the sequence of the resulting list of Evaluation .
 - **asc** - Arranges the list in ascending order (A-Z, 0-9).
 - **dsc** - Arranges the list in descending order (Z-A, 9-0).

Results are sorted by FilterVariable .

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a NextToken will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Results': [
    {
      'EvaluationId': 'string',
      'MLModelId': 'string',
      'EvaluationDataSourceId': 'string',
      'InputDataLocationS3': 'string',
      'CreatedByIamUser': 'string',
      'CreatedAt': datetime(2015, 1, 1),
      'LastUpdatedAt': datetime(2015, 1, 1),
      'Name': 'string',
      'Status': 'PENDING' | 'INPROGRESS' | 'FAILED' | 'COMPLETED' | 'DELETED',
      'PerformanceMetrics': {
        'Properties': {
          'string': 'string'
        }
      },
      'Message': 'string'
    },
  ],
}
```

Response Structure

- (*dict*) –

Represents the query results from a `DescribeEvaluations` operation. The content is essentially a list of `Evaluation`.

- **Results** (*list*) –

A list of `Evaluation` that meet the search criteria.

- * (*dict*) –

Represents the output of `GetEvaluation` operation.

The content consists of the detailed metadata and data file information and the current status of the `Evaluation`.

- **EvaluationId** (*string*) –

The ID that is assigned to the `Evaluation` at creation.

- **MLModelId** (*string*) –

The ID of the `MLModel` that is the focus of the evaluation.

- **EvaluationDataSourceId** (*string*) –

The ID of the `DataSource` that is used to evaluate the `MLModel`.

- **InputDataLocationS3** (*string*) –

The location and name of the data in Amazon Simple Storage Server (Amazon S3) that is used in the evaluation.

- **CreatedByIamUser** (*string*) –

The AWS user account that invoked the evaluation. The account type can be either an AWS root account or an AWS Identity and Access Management (IAM) user account.

- **CreatedAt** (*datetime*) –

The time that the `Evaluation` was created. The time is expressed in epoch time.

- **LastUpdatedAt** (*datetime*) –

The time of the most recent edit to the `Evaluation`. The time is expressed in epoch time.

- **Name** (*string*) –

A user-supplied name or description of the `Evaluation`.

- **Status** (*string*) –

The status of the evaluation. This element can have one of the following values:

- `PENDING` - Amazon Machine Learning (Amazon ML) submitted a request to evaluate an `MLModel`.
- `INPROGRESS` - The evaluation is underway.
- `FAILED` - The request to evaluate an `MLModel` did not run to completion. It is not usable.
- `COMPLETED` - The evaluation process completed successfully.
- `DELETED` - The `Evaluation` is marked as deleted. It is not usable.

- **PerformanceMetrics** (*dict*) –

Measurements of how well the `MLModel` performed, using observations referenced by the `DataSource`. One of the following metrics is returned, based on the type of the `MLModel`:

- `BinaryAUC`: A binary `MLModel` uses the Area Under the Curve (AUC) technique to measure performance.
- `RegressionRMSE`: A regression `MLModel` uses the Root Mean Square Error (RMSE) technique to measure performance. RMSE measures the difference between predicted and actual values for a single variable.
- `MulticlassAvgFScore`: A multiclass `MLModel` uses the F1 score technique to measure performance.

For more information about performance metrics, please see the [Amazon Machine Learning Developer Guide](#).

- **Properties** (*dict*) –

- (*string*) –

- (*string*) –

- **Message** (*string*) –

A description of the most recent details about evaluating the `MLModel`.

class `MachineLearning.Paginator.DescribeMLModels`

```
paginator = client.get_paginator('describe_ml_models')
```

```
paginate(**kwargs)
```

Creates an iterator that will paginate through responses from `MachineLearning.Client.describe_ml_models()`.

Request Syntax

```
response_iterator = paginator.paginate(
    FilterVariable='CreatedAt'|'LastUpdatedAt'|'Status'|'Name'|'IAMUser'|'TrainingDataSourceId',
    EQ='string',
    GT='string',
    LT='string',
    GE='string',
    LE='string',
    NE='string',
    Prefix='string',
    SortOrder='asc'|'dsc',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **FilterVariable** (*string*) – Use one of the following variables to filter a list of `MLModel`:
 - `CreatedAt` - Sets the search criteria to `MLModel` creation date.
 - `Status` - Sets the search criteria to `MLModel` status.
 - `Name` - Sets the search criteria to the contents of `MLModel` ******* `Name`.
 - `IAMUser` - Sets the search criteria to the user account that invoked the `MLModel` creation.
 - `TrainingDataSourceId` - Sets the search criteria to the `DataSource` used to train one or more `MLModel`.
 - `RealtimeEndpointStatus` - Sets the search criteria to the `MLModel` real-time endpoint status.
 - `MLModelType` - Sets the search criteria to `MLModel` type: binary, regression, or multi-class.
 - `Algorithm` - Sets the search criteria to the algorithm that the `MLModel` uses.
 - `TrainingDataURI` - Sets the search criteria to the data file(s) used in training a `MLModel`. The URL can identify either a file or an Amazon Simple Storage Service (Amazon S3) bucket or directory.
- **EQ** (*string*) – The equal to operator. The `MLModel` results will have `FilterVariable` values that exactly match the value specified with `EQ`.
- **GT** (*string*) – The greater than operator. The `MLModel` results will have `FilterVariable` values that are greater than the value specified with `GT`.
- **LT** (*string*) – The less than operator. The `MLModel` results will have `FilterVariable` values that are less than the value specified with `LT`.
- **GE** (*string*) – The greater than or equal to operator. The `MLModel` results will have `FilterVariable` values that are greater than or equal to the value specified with `GE`.
- **LE** (*string*) – The less than or equal to operator. The `MLModel` results will have `FilterVariable` values that are less than or equal to the value specified with `LE`.
- **NE** (*string*) – The not equal to operator. The `MLModel` results will have `FilterVariable` values not equal to the value specified with `NE`.

- **Prefix** (*string*) – A string that is found at the beginning of a variable, such as Name or Id.

For example, an `MLModel` could have the Name `2014-09-09-HolidayGiftMailer`. To search for this `MLModel`, select Name for the `FilterVariable` and any of the following strings for the `Prefix`:

- `2014-09`
- `2014-09-09`
- `2014-09-09-Holiday`

- **SortOrder** (*string*) – A two-value parameter that determines the sequence of the resulting list of `MLModel`.

- `asc` - Arranges the list in ascending order (A-Z, 0-9).
- `dsc` - Arranges the list in descending order (Z-A, 9-0).

Results are sorted by `FilterVariable`.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Results': [
    {
      'MLModelId': 'string',
      'TrainingDataSourceId': 'string',
      'CreatedByIamUser': 'string',
      'CreatedAt': datetime(2015, 1, 1),
      'LastUpdatedAt': datetime(2015, 1, 1),
      'Name': 'string',
      'Status': 'PENDING'|'INPROGRESS'|'FAILED'|'COMPLETED'|'DELETED',
      'SizeInBytes': 123,
      'EndpointInfo': {
        'PeakRequestsPerSecond': 123,
        'CreatedAt': datetime(2015, 1, 1),
        'EndpointUrl': 'string',
        'EndpointStatus': 'NONE'|'READY'|'UPDATING'|'FAILED'
      },
      'TrainingParameters': {
        'string': 'string'
      },
      'InputDataLocationS3': 'string',
      'Algorithm': 'sgd',
      'MLModelType': 'REGRESSION'|'BINARY'|'MULTICLASS',
      'ScoreThreshold': ...,
    }
  ]
}
```

```

        'ScoreThresholdLastUpdatedAt': datetime(2015, 1, 1),
        'Message': 'string'
    },
],
}

```

Response Structure

- *(dict)* –

Represents the output of a DescribeMLModels operation. The content is essentially a list of MLModel .

- **Results** (*list*) –

A list of MLModel that meet the search criteria.

- * *(dict)* –

Represents the output of a GetMLModel operation.

The content consists of the detailed metadata and the current status of the MLModel .

- **MLModelId** (*string*) –

The ID assigned to the MLModel at creation.

- **TrainingDataSourceId** (*string*) –

The ID of the training DataSource . The CreateMLModel operation uses the TrainingDataSourceId .

- **CreatedByIamUser** (*string*) –

The AWS user account from which the MLModel was created. The account type can be either an AWS root account or an AWS Identity and Access Management (IAM) user account.

- **CreatedAt** (*datetime*) –

The time that the MLModel was created. The time is expressed in epoch time.

- **LastUpdatedAt** (*datetime*) –

The time of the most recent edit to the MLModel . The time is expressed in epoch time.

- **Name** (*string*) –

A user-supplied name or description of the MLModel .

- **Status** (*string*) –

The current status of an MLModel . This element can have one of the following values:

- **PENDING** - Amazon Machine Learning (Amazon ML) submitted a request to create an MLModel .
- **INPROGRESS** - The creation process is underway.
- **FAILED** - The request to create an MLModel did not run to completion. It is not usable.
- **COMPLETED** - The creation process completed successfully.
- **DELETED** - The MLModel is marked as deleted. It is not usable.
- **SizeInBytes** (*integer*) –

Long integer type that is a 64-bit signed number.

- **EndpointInfo** (*dict*) –

The current endpoint of the `MLModel` .

- **PeakRequestsPerSecond** (*integer*) –

The maximum processing rate for the real-time endpoint for `MLModel` , measured in incoming requests per second.

- **CreatedAt** (*datetime*) –

The time that the request to create the real-time endpoint for the `MLModel` was received. The time is expressed in epoch time.

- **EndpointUrl** (*string*) –

The URI that specifies where to send real-time prediction requests for the `MLModel` .

Note: Note

The application must wait until the real-time endpoint is ready before using this URI.

- **EndpointStatus** (*string*) –

The current status of the real-time endpoint for the `MLModel` . This element can have one of the following values:

- NONE - Endpoint does not exist or was previously deleted.
- READY - Endpoint is ready to be used for real-time predictions.
- UPDATING - Updating/creating the endpoint.
- **TrainingParameters** (*dict*) –

A list of the training parameters in the `MLModel` . The list is implemented as a map of key/value pairs.

The following is the current set of training parameters:

- `sgd.l1RegularizationAmount` - Coefficient regularization L1 norm. It controls overfitting the data by penalizing large coefficients. This tends to drive coefficients to zero, resulting in a sparse feature set. If you use this parameter, specify a small value, such as 1.0E-04 or 1.0E-08. The value is a double that ranges from 0 to MAX_DOUBLE. The default is not to use L1 normalization. The parameter cannot be used when L2 is specified. Use this parameter sparingly.
- `sgd.l2RegularizationAmount` - Coefficient regularization L2 norm. It controls overfitting the data by penalizing large coefficients. This tends to drive coefficients to small, nonzero values. If you use this parameter, specify a small value, such as 1.0E-04 or 1.0E-08. The value is a double that ranges from 0 to MAX_DOUBLE. The default is not to use L2 normalization. This cannot be used when L1 is specified. Use this parameter sparingly.
- `sgd.maxPasses` - Number of times that the training process traverses the observations to build the `MLModel` . The value is an integer that ranges from 1 to 10000. The default value is 10.
- `sgd.maxMLModelSizeInBytes` - Maximum allowed

size of the model. Depending on the input data, the model size might affect performance. The value is an integer that ranges from 100000 to 2147483648. The default value is 33554432.

- *(string)* –

String type.

- *(string)* –

String type.

- **InputDataLocationS3** *(string)* –

The location of the data file or directory in Amazon Simple Storage Service (Amazon S3).

- **Algorithm** *(string)* –

The algorithm used to train the `MLModel`. The following algorithm is supported:

- SGD – Stochastic gradient descent. The goal of SGD is to minimize the gradient of the loss function.

- **MLModelType** *(string)* –

Identifies the `MLModel` category. The following are the available types:

- REGRESSION - Produces a numeric result. For example, “What listing price should a house have?”.
- BINARY - Produces one of two possible results. For example, “Is this a child-friendly web site?”.
- MULTICLASS - Produces more than two possible results. For example, “Is this a HIGH, LOW or MEDIUM risk trade?”.

- **ScoreThreshold** *(float)* –

- **ScoreThresholdLastUpdatedAt** *(datetime)* –

The time of the most recent edit to the `ScoreThreshold`. The time is expressed in epoch time.

- **Message** *(string)* –

A description of the most recent details about accessing the `MLModel`.

Waiters

The available waiters are:

- `MachineLearning.Waiter.BatchPredictionAvailable`
- `MachineLearning.Waiter.DataSourceAvailable`
- `MachineLearning.Waiter.EvaluationAvailable`
- `MachineLearning.Waiter.MLModelAvailable`

class `MachineLearning.Waiter.BatchPredictionAvailable`

```
waiter = client.get_waiter('batch_prediction_available')
```

wait (***kwargs*)

Polls `MachineLearning.Client.describe_batch_predictions()` every 30 seconds until a successful state is reached. An error is returned after 60 failed checks.

Request Syntax

```
waiter.wait(
    FilterVariable='CreatedAt'|'LastUpdatedAt'|'Status'|'Name'|'IAMUser'|'MLModelId'|'DataSourceId'|'DataURI',
    EQ='string',
    GT='string',
    LT='string',
    GE='string',
    LE='string',
    NE='string',
    Prefix='string',
    SortOrder='asc'|'desc',
    NextToken='string',
    Limit=123
)
```

Parameters

- **FilterVariable** (*string*) – Use one of the following variables to filter a list of BatchPrediction:
 - CreatedAt - Sets the search criteria to the BatchPrediction creation date.
 - Status - Sets the search criteria to the BatchPrediction status.
 - Name - Sets the search criteria to the contents of the BatchPrediction ******* Name.
 - IAMUser - Sets the search criteria to the user account that invoked the BatchPrediction creation.
 - MLModelId - Sets the search criteria to the MLModel used in the BatchPrediction.
 - DataSourceId - Sets the search criteria to the DataSource used in the BatchPrediction.
 - DataURI - Sets the search criteria to the data file(s) used in the BatchPrediction. The URL can identify either a file or an Amazon Simple Storage Solution (Amazon S3) bucket or directory.
- **EQ** (*string*) – The equal to operator. The BatchPrediction results will have FilterVariable values that exactly match the value specified with EQ.
- **GT** (*string*) – The greater than operator. The BatchPrediction results will have FilterVariable values that are greater than the value specified with GT.
- **LT** (*string*) – The less than operator. The BatchPrediction results will have FilterVariable values that are less than the value specified with LT.
- **GE** (*string*) – The greater than or equal to operator. The BatchPrediction results will have FilterVariable values that are greater than or equal to the value specified with GE.
- **LE** (*string*) – The less than or equal to operator. The BatchPrediction results will have FilterVariable values that are less than or equal to the value specified with LE.
- **NE** (*string*) – The not equal to operator. The BatchPrediction results will have FilterVariable values not equal to the value specified with NE.
- **Prefix** (*string*) – A string that is found at the beginning of a variable, such as Name or Id.

For example, a Batch Prediction operation could have the

Name 2014-09-09-HolidayGiftMailer . To search for this BatchPrediction , select Name for the FilterVariable and any of the following strings for the Prefix :

- 2014-09
- 2014-09-09
- 2014-09-09-Holiday

- **SortOrder** (*string*) – A two-value parameter that determines the sequence of the resulting list of MLModel s.
 - asc - Arranges the list in ascending order (A-Z, 0-9).
 - dsc - Arranges the list in descending order (Z-A, 9-0).

Results are sorted by FilterVariable .

- **NextToken** (*string*) – An ID of the page in the paginated results.
- **Limit** (*integer*) – The number of pages of information to include in the result. The range of acceptable values is 1 through 100. The default value is 100.

Returns None

class MachineLearning.Waiter.**DataSourceAvailable**

```
waiter = client.get_waiter('data_source_available')
```

wait (***kwargs*)

Polls `MachineLearning.Client.describe_data_sources()` every 30 seconds until a successful state is reached. An error is returned after 60 failed checks.

Request Syntax

```
waiter.wait(
    FilterVariable='CreatedAt'|'LastUpdatedAt'|'Status'|'Name'|'DataLocationS3'|'IAMUser',
    EQ='string',
    GT='string',
    LT='string',
    GE='string',
    LE='string',
    NE='string',
    Prefix='string',
    SortOrder='asc'|'dsc',
    NextToken='string',
    Limit=123
)
```

Parameters

- **FilterVariable** (*string*) – Use one of the following variables to filter a list of DataSource :
 - CreatedAt - Sets the search criteria to DataSource creation dates.
 - Status - Sets the search criteria to DataSource statuses.
 - Name - Sets the search criteria to the contents of DataSource ******* Name .
 - DataUri - Sets the search criteria to the URI of data files used to create the DataSource . The URI can identify either a file or an Amazon Simple Storage Service (Amazon S3) bucket or directory.
 - IAMUser - Sets the search criteria to the user account that invoked the DataSource creation.
- **EQ** (*string*) – The equal to operator. The DataSource results will have FilterVariable values that exactly match the value specified with EQ .

- **GT** (*string*) – The greater than operator. The `DataSource` results will have `FilterVariable` values that are greater than the value specified with `GT`.
- **LT** (*string*) – The less than operator. The `DataSource` results will have `FilterVariable` values that are less than the value specified with `LT`.
- **GE** (*string*) – The greater than or equal to operator. The `DataSource` results will have `FilterVariable` values that are greater than or equal to the value specified with `GE`.
- **LE** (*string*) – The less than or equal to operator. The `DataSource` results will have `FilterVariable` values that are less than or equal to the value specified with `LE`.
- **NE** (*string*) – The not equal to operator. The `DataSource` results will have `FilterVariable` values not equal to the value specified with `NE`.
- **Prefix** (*string*) – A string that is found at the beginning of a variable, such as `Name` or `Id`.

For example, a `DataSource` could have the `Name` `2014-09-09-HolidayGiftMailer`. To search for this `DataSource`, select `Name` for the `FilterVariable` and any of the following strings for the `Prefix`:

- `2014-09`
- `2014-09-09`
- `2014-09-09-Holiday`

- **SortOrder** (*string*) – A two-value parameter that determines the sequence of the resulting list of `DataSource`.
 - `asc` - Arranges the list in ascending order (A-Z, 0-9).
 - `dsc` - Arranges the list in descending order (Z-A, 9-0).

Results are sorted by `FilterVariable`.

- **NextToken** (*string*) – The ID of the page in the paginated results.
- **Limit** (*integer*) – The maximum number of `DataSource` to include in the result.

Returns `None`

`class MachineLearning.Waiter.EvaluationAvailable`

```
waiter = client.get_waiter('evaluation_available')
```

wait (***kwargs*)

Polls `MachineLearning.Client.describe_evaluations()` every 30 seconds until a successful state is reached. An error is returned after 60 failed checks.

Request Syntax

```
waiter.wait(  
    FilterVariable='CreatedAt'|'LastUpdatedAt'|'Status'|'Name'|'IAMUser'|'MLModelId'|'DataS  
    EQ='string',  
    GT='string',  
    LT='string',  
    GE='string',  
    LE='string',  
    NE='string',  
    Prefix='string',  
    SortOrder='asc'|'dsc',  
    NextToken='string',  
    Limit=123  
)
```

Parameters

- **FilterVariable** (*string*) – Use one of the following variable to filter a list of Evaluation objects:
 - CreatedAt - Sets the search criteria to the Evaluation creation date.
 - Status - Sets the search criteria to the Evaluation status.
 - Name - Sets the search criteria to the contents of Evaluation **Name**.
 - IAMUser - Sets the search criteria to the user account that invoked an Evaluation.
 - MLModelId - Sets the search criteria to the MLModel that was evaluated.
 - DataSourceId - Sets the search criteria to the DataSource used in Evaluation.
 - DataUri - Sets the search criteria to the data file(s) used in Evaluation. The URL can identify either a file or an Amazon Simple Storage Solution (Amazon S3) bucket or directory.
- **EQ** (*string*) – The equal to operator. The Evaluation results will have FilterVariable values that exactly match the value specified with EQ.
- **GT** (*string*) – The greater than operator. The Evaluation results will have FilterVariable values that are greater than the value specified with GT.
- **LT** (*string*) – The less than operator. The Evaluation results will have FilterVariable values that are less than the value specified with LT.
- **GE** (*string*) – The greater than or equal to operator. The Evaluation results will have FilterVariable values that are greater than or equal to the value specified with GE.
- **LE** (*string*) – The less than or equal to operator. The Evaluation results will have FilterVariable values that are less than or equal to the value specified with LE.
- **NE** (*string*) – The not equal to operator. The Evaluation results will have FilterVariable values not equal to the value specified with NE.
- **Prefix** (*string*) – A string that is found at the beginning of a variable, such as Name or Id.

For example, an Evaluation could have the Name 2014-09-09-HolidayGiftMailer. To search for this Evaluation, select Name for the FilterVariable and any of the following strings for the Prefix:

- 2014-09
- 2014-09-09
- 2014-09-09-Holiday
- **SortOrder** (*string*) – A two-value parameter that determines the sequence of the resulting list of Evaluation.
 - asc - Arranges the list in ascending order (A-Z, 0-9).
 - desc - Arranges the list in descending order (Z-A, 9-0).
 Results are sorted by FilterVariable.
- **NextToken** (*string*) – The ID of the page in the paginated results.
- **Limit** (*integer*) – The maximum number of Evaluation to include in the result.

Returns None

```
class MachineLearning.Waiter.MLModelAvailable
```

```
waiter = client.get_waiter('ml_model_available')
```

```
wait (**kwargs)
```

Polls `MachineLearning.Client.describe_ml_models()` every 30 seconds until a successful state is reached. An error is returned after 60 failed checks.

Request Syntax

```
waiter.wait(
    FilterVariable='CreatedAt'|'LastUpdatedAt'|'Status'|'Name'|'IAMUser'|'TrainingDataSourceId',
    EQ='string',
    GT='string',
    LT='string',
    GE='string',
    LE='string',
    NE='string',
    Prefix='string',
    SortOrder='asc'|'dsc',
    NextToken='string',
    Limit=123
)
```

Parameters

- **FilterVariable** (*string*) – Use one of the following variables to filter a list of `MLModel` :
 - `CreatedAt` - Sets the search criteria to `MLModel` creation date.
 - `Status` - Sets the search criteria to `MLModel` status.
 - `Name` - Sets the search criteria to the contents of `MLModel` ******* `Name` .
 - `IAMUser` - Sets the search criteria to the user account that invoked the `MLModel` creation.
 - `TrainingDataSourceId` - Sets the search criteria to the `DataSource` used to train one or more `MLModel` .
 - `RealtimeEndpointStatus` - Sets the search criteria to the `MLModel` real-time endpoint status.
 - `MLModelType` - Sets the search criteria to `MLModel` type: binary, regression, or multi-class.
 - `Algorithm` - Sets the search criteria to the algorithm that the `MLModel` uses.
 - `TrainingDataURI` - Sets the search criteria to the data file(s) used in training a `MLModel` . The URL can identify either a file or an Amazon Simple Storage Service (Amazon S3) bucket or directory.
- **EQ** (*string*) – The equal to operator. The `MLModel` results will have `FilterVariable` values that exactly match the value specified with `EQ` .
- **GT** (*string*) – The greater than operator. The `MLModel` results will have `FilterVariable` values that are greater than the value specified with `GT` .
- **LT** (*string*) – The less than operator. The `MLModel` results will have `FilterVariable` values that are less than the value specified with `LT` .
- **GE** (*string*) – The greater than or equal to operator. The `MLModel` results will have `FilterVariable` values that are greater than or equal to the value specified with `GE` .
- **LE** (*string*) – The less than or equal to operator. The `MLModel` results will have `FilterVariable` values that are less than or equal to the value specified with `LE` .
- **NE** (*string*) – The not equal to operator. The `MLModel` results will have `FilterVariable` values not equal to the value specified with `NE` .
- **Prefix** (*string*) – A string that is found at the beginning of a variable, such as `Name` or `Id` .

For example, an `MLModel` could have the Name `2014-09-09-HolidayGiftMailer`. To search for this `MLModel`, select Name for the `FilterVariable` and any of the following strings for the `Prefix`:

- 2014-09
- 2014-09-09
- 2014-09-09-Holiday
- **SortOrder** (*string*) – A two-value parameter that determines the sequence of the resulting list of `MLModel`.
 - `asc` - Arranges the list in ascending order (A-Z, 0-9).
 - `dsc` - Arranges the list in descending order (Z-A, 9-0).
 Results are sorted by `FilterVariable`.
- **NextToken** (*string*) – The ID of the page in the paginated results.
- **Limit** (*integer*) – The number of pages of information to include in the result. The range of acceptable values is 1 through 100. The default value is 100.

Returns None

OpsWorks

Table of Contents

- [OpsWorks](#)
 - [Client](#)
 - [Service Resource](#)
 - [Layer](#)
 - [Stack](#)
 - [StackSummary](#)

Client

class `OpsWorks.Client`

A low-level client representing AWS OpsWorks:

```
import boto3

client = boto3.client('opsworks')
```

These are the available methods:

- `assign_instance()`
- `assign_volume()`
- `associate_elastic_ip()`
- `attach_elastic_load_balancer()`
- `can_paginate()`
- `clone_stack()`
- `create_app()`
- `create_deployment()`
- `create_instance()`
- `create_layer()`
- `create_stack()`
- `create_user_profile()`
- `delete_app()`

- `delete_instance()`
- `delete_layer()`
- `delete_stack()`
- `delete_user_profile()`
- `deregister_ecs_cluster()`
- `deregister_elastic_ip()`
- `deregister_instance()`
- `deregister_rds_db_instance()`
- `deregister_volume()`
- `describe_agent_versions()`
- `describe_apps()`
- `describe_commands()`
- `describe_deployments()`
- `describe_ecs_clusters()`
- `describe_elastic_ips()`
- `describe_elastic_load_balancers()`
- `describe_instances()`
- `describe_layers()`
- `describe_load_based_auto_scaling()`
- `describe_my_user_profile()`
- `describe_permissions()`
- `describe_raid_arrays()`
- `describe_rds_db_instances()`
- `describe_service_errors()`
- `describe_stack_provisioning_parameters()`
- `describe_stack_summary()`
- `describe_stacks()`
- `describe_time_based_auto_scaling()`
- `describe_user_profiles()`
- `describe_volumes()`
- `detach_elastic_load_balancer()`
- `disassociate_elastic_ip()`
- `generate_presigned_url()`
- `get_hostname_suggestion()`
- `get_paginator()`
- `get_waiter()`
- `grant_access()`
- `reboot_instance()`
- `register_ecs_cluster()`
- `register_elastic_ip()`
- `register_instance()`
- `register_rds_db_instance()`
- `register_volume()`
- `set_load_based_auto_scaling()`
- `set_permission()`
- `set_time_based_auto_scaling()`
- `start_instance()`
- `start_stack()`
- `stop_instance()`
- `stop_stack()`
- `unassign_instance()`
- `unassign_volume()`
- `update_app()`
- `update_elastic_ip()`

- `update_instance()`
- `update_layer()`
- `update_my_user_profile()`
- `update_rds_db_instance()`
- `update_stack()`
- `update_user_profile()`
- `update_volume()`

assign_instance (**kwargs)

Assign a registered instance to a layer.

- You can assign registered on-premises instances to any layer type.
- You can assign registered Amazon EC2 instances only to custom layers.
- You cannot use this action with instances that were created with AWS OpsWorks.

Required Permissions : To use this action, an AWS Identity and Access Management (IAM) user must have a Manage permissions level for the stack or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.assign_instance(
    InstanceId='string',
    LayerIds=[
        'string',
    ]
)
```

Parameters

- **InstanceId** (*string*) – [REQUIRED]

The instance ID.

- **LayerIds** (*list*) – [REQUIRED]

The layer ID, which must correspond to a custom layer. You cannot assign a registered instance to a built-in layer.

– (*string*) –

Returns None

assign_volume (**kwargs)

Assigns one of the stack's registered Amazon EBS volumes to a specified instance. The volume must first be registered with the stack by calling `RegisterVolume` . After you register the volume, you must call `UpdateVolume` to specify a mount point before calling `AssignVolume` . For more information, see [Resource Management](#) .

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.assign_volume(
    VolumeId='string',
    InstanceId='string'
)
```

Parameters

- **VolumeId** (*string*) – [REQUIRED]

The volume ID.

- **InstanceId** (*string*) – The instance ID.

Returns None

associate_elastic_ip (***kwargs*)

Associates one of the stack's registered Elastic IP addresses with a specified instance. The address must first be registered with the stack by calling `RegisterElasticIp`. For more information, see [Resource Management](#).

Required Permissions : To use this action, an IAM user must have a `Manage` permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.associate_elastic_ip(
    ElasticIp='string',
    InstanceId='string'
)
```

Parameters

- **ElasticIp** (*string*) – [REQUIRED]
The Elastic IP address.
- **InstanceId** (*string*) – The instance ID.

Returns None

attach_elastic_load_balancer (***kwargs*)

Attaches an Elastic Load Balancing load balancer to a specified layer. For more information, see [Elastic Load Balancing](#).

Note: You must create the Elastic Load Balancing instance separately, by using the Elastic Load Balancing console, API, or CLI. For more information, see [Elastic Load Balancing Developer Guide](#).

Required Permissions : To use this action, an IAM user must have a `Manage` permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.attach_elastic_load_balancer(
    ElasticLoadBalancerName='string',
    LayerId='string'
)
```

Parameters

- **ElasticLoadBalancerName** (*string*) – [REQUIRED]
The Elastic Load Balancing instance's name.
- **LayerId** (*string*) – [REQUIRED]
The ID of the layer that the Elastic Load Balancing instance is to be attached to.

Returns None

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

clone_stack (**kwargs)

Creates a clone of a specified stack. For more information, see [Clone a Stack](#) . By default, all parameters are set to the values used by the parent stack.

Required Permissions : To use this action, an IAM user must have an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.clone_stack(
    SourceStackId='string',
    Name='string',
    Region='string',
    VpcId='string',
    Attributes={
        'string': 'string'
    },
    ServiceRoleArn='string',
    DefaultInstanceProfileArn='string',
    DefaultOs='string',
    HostnameTheme='string',
    DefaultAvailabilityZone='string',
    DefaultSubnetId='string',
    CustomJson='string',
    ConfigurationManager={
        'Name': 'string',
        'Version': 'string'
    },
    ChefConfiguration={
        'ManageBerkshelf': True|False,
        'BerkshelfVersion': 'string'
    },
    UseCustomCookbooks=True|False,
    UseOpsworksSecurityGroups=True|False,
    CustomCookbooksSource={
        'Type': 'git'|'svn'|'archive'|'s3',
        'Url': 'string',
        'Username': 'string',
        'Password': 'string',
        'SshKey': 'string',
        'Revision': 'string'
    },
    DefaultSshKeyName='string',
    ClonePermissions=True|False,
    CloneAppIds=[
        'string',
    ],
    DefaultRootDeviceType='ebs'|'instance-store',
    AgentVersion='string'
)
```

Parameters

- **SourceStackId** (*string*) – [REQUIRED]
The source stack ID.
- **Name** (*string*) – The cloned stack name.

- **Region** (*string*) – The cloned stack AWS region, such as “us-east-1”. For more information about AWS regions, see [Regions and Endpoints](#).
- **VpcId** (*string*) – The ID of the VPC that the cloned stack is to be launched into. It must be in the specified region. All instances are launched into this VPC, and you cannot change the ID later.
 - If your account supports EC2 Classic, the default value is no VPC.
 - If your account does not support EC2 Classic, the default value is the default VPC for the specified region.

If the VPC ID corresponds to a default VPC and you have specified either the `DefaultAvailabilityZone` or the `DefaultSubnetId` parameter only, AWS OpsWorks infers the value of the other parameter. If you specify neither parameter, AWS OpsWorks sets these parameters to the first valid Availability Zone for the specified region and the corresponding default VPC subnet ID, respectively.

If you specify a nondefault VPC ID, note the following:

- It must belong to a VPC in your account that is in the specified region.
- You must specify a value for `DefaultSubnetId`.

For more information on how to use AWS OpsWorks with a VPC, see [Running a Stack in a VPC](#). For more information on default VPC and EC2 Classic, see [Supported Platforms](#).

- **Attributes** (*dict*) – A list of stack attributes and values as key/value pairs to be added to the cloned stack.
 - (*string*) –
 - * (*string*) –
- **ServiceRoleArn** (*string*) – **[REQUIRED]**

The stack AWS Identity and Access Management (IAM) role, which allows AWS OpsWorks to work with AWS resources on your behalf. You must set this parameter to the Amazon Resource Name (ARN) for an existing IAM role. If you create a stack by using the AWS OpsWorks console, it creates the role for you. You can obtain an existing stack’s IAM ARN programmatically by calling `DescribePermissions`. For more information about IAM ARNs, see [Using Identifiers](#).

Note: You must set this parameter to a valid service role ARN or the action will fail; there is no default value. You can specify the source stack’s service role ARN, if you prefer, but you must do so explicitly.

- **DefaultInstanceProfileArn** (*string*) – The Amazon Resource Name (ARN) of an IAM profile that is the default profile for all of the stack’s EC2 instances. For more information about IAM ARNs, see [Using Identifiers](#).
- **DefaultOs** (*string*) – The stack’s operating system, which must be set to one of the following.
 - A supported Linux operating system: An Amazon Linux version, such as Amazon Linux 2015.03, Red Hat Enterprise Linux 7, Ubuntu 12.04 LTS, or Ubuntu 14.04 LTS.
 - Microsoft Windows Server 2012 R2 Base.
 - A custom AMI: `Custom`. You specify the custom AMI you want to use when you create instances. For more information on how to use custom AMIs with OpsWorks, see [Using Custom AMIs](#).

The default option is the parent stack’s operating system. For more information on the supported operating systems, see [AWS OpsWorks Operating Systems](#).

Note: You can specify a different Linux operating system for the cloned stack,

but you cannot change from Linux to Windows or Windows to Linux.

- **HostnameTheme** (*string*) – The stack’s host name theme, with spaces are replaced by underscores. The theme is used to generate host names for the stack’s instances. By default, `HostnameTheme` is set to `Layer_Dependent`, which creates host names by appending integers to the layer’s short name. The other themes are:

- `Baked_Goods`
- `Clouds`
- `Europe_Cities`
- `Fruits`
- `Greek_Deities`
- `Legendary_creatures_from_Japan`
- `Planets_and_Moons`
- `Roman_Deities`
- `Scottish_Islands`
- `US_Cities`
- `Wild_Cats`

To obtain a generated host name, call `GetHostNameSuggestion`, which returns a host name based on the current theme.

- **DefaultAvailabilityZone** (*string*) – The cloned stack’s default Availability Zone, which must be in the specified region. For more information, see [Regions and Endpoints](#). If you also specify a value for `DefaultSubnetId`, the subnet must be in the same zone. For more information, see the `VpcId` parameter description.
- **DefaultSubnetId** (*string*) – The stack’s default VPC subnet ID. This parameter is required if you specify a value for the `VpcId` parameter. All instances are launched into this subnet unless you specify otherwise when you create the instance. If you also specify a value for `DefaultAvailabilityZone`, the subnet must be in that zone. For information on default values and when this parameter is required, see the `VpcId` parameter description.
- **CustomJson** (*string*) – A string that contains user-defined, custom JSON. It is used to override the corresponding default stack configuration JSON values. The string should be in the following format and must escape characters such as “”:

```
{\"key1\": \"value1\", \"key2\":  
  \"value2\", ...}
```

For more information on custom JSON, see [Use Custom JSON to Modify the Stack Configuration Attributes](#)

- **ConfigurationManager** (*dict*) – The configuration manager. When you clone a Linux stack we recommend that you use the configuration manager to specify the Chef version: 0.9, 11.4, or 11.10. The default value is currently 11.10.
 - **Name** (*string*) –
 The name. This parameter must be set to “Chef”.
 - **Version** (*string*) –
 The Chef version. This parameter must be set to 0.9, 11.4, or 11.10. The default value is 11.4.
- **ChefConfiguration** (*dict*) – A `ChefConfiguration` object that specifies whether to enable Berkshelf and the Berkshelf version on Chef 11.10 stacks. For more information, see [Create a New Stack](#).
 - **ManageBerkshelf** (*boolean*) –
 Whether to enable Berkshelf.
 - **BerkshelfVersion** (*string*) –

The Berkshelf version.

- **UseCustomCookbooks** (*boolean*) – Whether to use custom cookbooks.
- **UseOpsworksSecurityGroups** (*boolean*) – Whether to associate the AWS OpsWorks built-in security groups with the stack’s layers.

AWS OpsWorks provides a standard set of built-in security groups, one for each layer, which are associated with layers by default. With `UseOpsworksSecurityGroups` you can instead provide your own custom security groups. `UseOpsworksSecurityGroups` has the following settings:

- **True** - AWS OpsWorks automatically associates the appropriate built-in security group with each layer (default setting). You can associate additional security groups with a layer after you create it but you cannot delete the built-in security group.
- **False** - AWS OpsWorks does not associate built-in security groups with layers. You must create appropriate Amazon Elastic Compute Cloud (Amazon EC2) security groups and associate a security group with each layer that you create. However, you can still manually associate a built-in security group with a layer on creation; custom security groups are required only for those layers that need custom settings.

For more information, see [Create a New Stack](#).

- **CustomCookbooksSource** (*dict*) – Contains the information required to retrieve an app or cookbook from a repository. For more information, see [Creating Apps](#) or [Custom Recipes and Cookbooks](#).

- **Type** (*string*) –

The repository type.

- **Url** (*string*) –

The source URL.

- **Username** (*string*) –

This parameter depends on the repository type.

- * For Amazon S3 bundles, set `Username` to the appropriate IAM access key ID.
- * For HTTP bundles, Git repositories, and Subversion repositories, set `Username` to the user name.

- **Password** (*string*) –

When included in a request, the parameter depends on the repository type.

- * For Amazon S3 bundles, set `Password` to the appropriate IAM secret access key.
- * For HTTP bundles and Subversion repositories, set `Password` to the password.

For more information on how to safely handle IAM credentials, see <http://docs.aws.amazon.com/general/latest/gr/aws-access-keys-best-practices.html>.

In responses, AWS OpsWorks returns *****FILTERED***** instead of the actual value.

- **SshKey** (*string*) –

In requests, the repository’s SSH key.

In responses, AWS OpsWorks returns *****FILTERED***** instead of the actual value.

- **Revision** (*string*) –

The application's version. AWS OpsWorks enables you to easily deploy new versions of an application. One of the simplest approaches is to have branches or revisions in your repository that represent different versions that can potentially be deployed.

- **DefaultSshKeyName** (*string*) – A default Amazon EC2 key pair name. The default value is none. If you specify a key pair name, AWS OpsWorks installs the public key on the instance and you can use the private key with an SSH client to log in to the instance. For more information, see [Using SSH to Communicate with an Instance](#) and [Managing SSH Access](#) . You can override this setting by specifying a different key pair, or no key pair, when you [create an instance](#) .
- **ClonePermissions** (*boolean*) – Whether to clone the source stack's permissions.
- **CloneAppIds** (*list*) – A list of source stack app IDs to be included in the cloned stack.
 - (*string*) –
- **DefaultRootDeviceType** (*string*) – The default root device type. This value is used by default for all instances in the cloned stack, but you can override it when you create an instance. For more information, see [Storage for the Root Device](#) .
- **AgentVersion** (*string*) – The default AWS OpsWorks agent version. You have the following options:
 - Auto-update - Set this parameter to `LATEST` . AWS OpsWorks automatically installs new agent versions on the stack's instances as soon as they are available.
 - Fixed version - Set this parameter to your preferred agent version. To update the agent version, you must edit the stack configuration and specify a new version. AWS OpsWorks then automatically installs that version on the stack's instances.

The default setting is `LATEST` . To specify an agent version, you must use the complete version number, not the abbreviated number shown on the console. For a list of available agent version numbers, call `DescribeAgentVersions` .

Note: You can also specify an agent version when you create or update an instance, which overrides the stack's default setting.

Return type dict
Returns

Response Syntax

```
{
    'StackId': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a `CloneStack` request.

- **StackId** (*string*) –

The cloned stack ID.

create_app (***kwargs*)

Creates an app for a specified stack. For more information, see [Creating Apps](#) .

Required Permissions : To use this action, an IAM user must have a `Manage` permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.create_app(  
    StackId='string',  
    Shortname='string',  
    Name='string',  
    Description='string',  
    DataSources=[  
        {  
            'Type': 'string',  
            'Arn': 'string',  
            'DatabaseName': 'string'  
        },  
    ],  
    Type='aws-flow-ruby'|'java'|'rails'|'php'|'nodejs'|'static'|'other',  
    AppSource={  
        'Type': 'git'|'svn'|'archive'|'s3',  
        'Url': 'string',  
        'Username': 'string',  
        'Password': 'string',  
        'SshKey': 'string',  
        'Revision': 'string'  
    },  
    Domains=[  
        'string',  
    ],  
    EnableSsl=True|False,  
    SslConfiguration={  
        'Certificate': 'string',  
        'PrivateKey': 'string',  
        'Chain': 'string'  
    },  
    Attributes={  
        'string': 'string'  
    },  
    Environment=[  
        {  
            'Key': 'string',  
            'Value': 'string',  
            'Secure': True|False  
        },  
    ]  
)
```

Parameters

- **StackId** (*string*) – [REQUIRED]

The stack ID.

- **Shortname** (*string*) – The app’s short name.
- **Name** (*string*) – [REQUIRED]

The app name.

- **Description** (*string*) – A description of the app.
- **DataSources** (*list*) – The app’s data source.
 - (*dict*) –

Describes an app’s data source.

- * **Type** (*string*) –

The data source’s type, `AutoSelectOpsworksMysqlInstance`, `OpsworksMysqlInstance`, or `RdsDbInstance`.

- * **Arn** (*string*) –
The data source’s ARN.
- * **DatabaseName** (*string*) –
The database name.
- **Type** (*string*) – [REQUIRED]
The app type. Each supported type is associated with a particular layer. For example, PHP applications are associated with a PHP layer. AWS OpsWorks deploys an application to those instances that are members of the corresponding layer. If your app isn’t one of the standard types, or you prefer to implement your own Deploy recipes, specify `other`.
- **AppSource** (*dict*) – A `Source` object that specifies the app repository.
 - **Type** (*string*) –
The repository type.
 - **Url** (*string*) –
The source URL.
 - **Username** (*string*) –
This parameter depends on the repository type.
 - * For Amazon S3 bundles, set `Username` to the appropriate IAM access key ID.
 - * For HTTP bundles, Git repositories, and Subversion repositories, set `Username` to the user name.
 - **Password** (*string*) –
When included in a request, the parameter depends on the repository type.
 - * For Amazon S3 bundles, set `Password` to the appropriate IAM secret access key.
 - * For HTTP bundles and Subversion repositories, set `Password` to the password.
 For more information on how to safely handle IAM credentials, see <http://docs.aws.amazon.com/general/latest/gr/aws-access-keys-best-practices.html>.
In responses, AWS OpsWorks returns `*****FILTERED*****` instead of the actual value.
 - **SshKey** (*string*) –
In requests, the repository’s SSH key.
In responses, AWS OpsWorks returns `*****FILTERED*****` instead of the actual value.
 - **Revision** (*string*) –
The application’s version. AWS OpsWorks enables you to easily deploy new versions of an application. One of the simplest approaches is to have branches or revisions in your repository that represent different versions that can potentially be deployed.
- **Domains** (*list*) – The app virtual host settings, with multiple domains separated by commas. For example: `'www.example.com, example.com'`
 - (*string*) –
- **EnableSsl** (*boolean*) – Whether to enable SSL for the app.
- **SslConfiguration** (*dict*) – An `SslConfiguration` object with the SSL configuration.
 - **Certificate** (*string*) – [REQUIRED]

The contents of the certificate's domain.crt file.

- **PrivateKey** (*string*) – **[REQUIRED]**

The private key; the contents of the certificate's domain.key file.

- **Chain** (*string*) –

Optional. Can be used to specify an intermediate certificate authority key or client authentication.

- **Attributes** (*dict*) – One or more user-defined key/value pairs to be added to the stack attributes.
 - (*string*) –
 - * (*string*) –
- **Environment** (*list*) – An array of `EnvironmentVariable` objects that specify environment variables to be associated with the app. After you deploy the app, these variables are defined on the associated app server instance. For more information, see [Environment Variables](#).

There is no specific limit on the number of environment variables. However, the size of the associated data structure - which includes the variables' names, values, and protected flag values - cannot exceed 10 KB (10240 Bytes). This limit should accommodate most if not all use cases. Exceeding it will cause an exception with the message, "Environment: is too large (maximum is 10KB)."

Note: This parameter is supported only by Chef 11.10 stacks. If you have specified one or more environment variables, you cannot modify the stack's Chef version.

- (*dict*) –

Represents an app's environment variable.

- * **Key** (*string*) – **[REQUIRED]**

(Required) The environment variable's name, which can consist of up to 64 characters and must be specified. The name can contain upper- and lowercase letters, numbers, and underscores (`_`), but it must start with a letter or underscore.

- * **Value** (*string*) – **[REQUIRED]**

(Optional) The environment variable's value, which can be left empty. If you specify a value, it can contain up to 256 characters, which must all be printable.

- * **Secure** (*boolean*) –

(Optional) Whether the variable's value will be returned by the `DescribeApps` action. To conceal an environment variable's value, set `Secure` to `true`. `DescribeApps` then returns `*****FILTERED*****` instead of the actual value. The default value for `Secure` is `false`.

Return type `dict`

Returns

Response Syntax

```
{
    'AppId': 'string'
}
```

Response Structure

- *(dict)* –
Contains the response to a `CreateApp` request.
 - **AppId** (*string*) –
The app ID.

create_deployment (***kwargs*)

Runs deployment or stack commands. For more information, see [Deploying Apps](#) and [Run Stack Commands](#).

Required Permissions : To use this action, an IAM user must have a `Deploy` or `Manage` permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.create_deployment(
    StackId='string',
    AppId='string',
    InstanceIds=[
        'string',
    ],
    Command={
        'Name': 'install_dependencies'|'update_dependencies'|'update_custom_cookbooks'|'execute_recipes',
        'Args': {
            'string': [
                'string',
            ]
        }
    },
    Comment='string',
    CustomJson='string'
)
```

Parameters

- **StackId** (*string*) – **[REQUIRED]**
The stack ID.
- **AppId** (*string*) – The app ID. This parameter is required for app deployments, but not for other deployment commands.
- **InstanceIds** (*list*) – The instance IDs for the deployment targets.
 - (*string*) –
- **Command** (*dict*) – **[REQUIRED]**

A `DeploymentCommand` object that specifies the deployment command and any associated arguments.

- **Name** (*string*) – **[REQUIRED]**

Specifies the operation. You can specify only one command.

For stacks, the following commands are available:

- * `execute_recipes` : Execute one or more recipes. To specify the recipes, set an `Args` parameter named `recipes` to the list of recipes to be executed. For example, to execute `phpapp::appsetup`, set `Args` to `{"recipes": ["phpapp::appsetup"]}`.
- * `install_dependencies` : Install the stack's dependencies.
- * `update_custom_cookbooks` : Update the stack's custom cookbooks.

* `update_dependencies` : Update the stack's dependencies.

Note: The `update_dependencies` and `install_dependencies` commands are supported only for Linux instances. You can run the commands successfully on Windows instances, but they do nothing.

For apps, the following commands are available:

- * `deploy` : Deploy an app. Ruby on Rails apps have an optional `Args` parameter named `migrate` . Set `Args` to `{"migrate":["true"]}` to migrate the database. The default setting is `{"migrate":["false"]}`.
- * `rollback` Roll the app back to the previous version. When you update an app, AWS OpsWorks stores the previous version, up to a maximum of five versions. You can use this command to roll an app back as many as four versions.
- * `start` : Start the app's web or application server.
- * `stop` : Stop the app's web or application server.
- * `restart` : Restart the app's web or application server.
- * `undeploy` : Undeploy the app.

– **Args** (*dict*) –

The arguments of those commands that take arguments. It should be set to a JSON object with the following format:

```
{"arg_name1" : ["value1", "value2",
...], "arg_name2" : ["value1",
"value2", ...], ...}
```

The `update_dependencies` command takes two arguments:

- * `upgrade_os_to` - Specifies the desired Amazon Linux version for instances whose OS you want to upgrade, such as Amazon Linux 2014.09 . You must also set the `allow_reboot` argument to `true`.
- * `allow_reboot` - Specifies whether to allow AWS OpsWorks to reboot the instances if necessary, after installing the updates. This argument can be set to either `true` or `false` . The default value is `false` .

For example, to upgrade an instance to Amazon Linux 2014.09, set `Args` to the following.

```
{ "upgrade_os_to":["Amazon Linux
2014.09"], "allow_reboot":["true"] }
```

- * (*string*) –
- (*list*) –
- (*string*) –

- **Comment** (*string*) – A user-defined comment.
- **CustomJson** (*string*) – A string that contains user-defined, custom JSON. It is used to override the corresponding default stack configuration JSON values. The string should be in the following format and must escape characters such as `""`:

```
"{\"key1\": \"value1\", \"key2\":
\"value2\", ...}"
```

For more information on custom JSON, see [Use Custom JSON to Modify the Stack Configuration Attributes](#) .

Return type `dict`

Returns

Response Syntax

```
{
    'DeploymentId': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a `CreateDeployment` request.

- **DeploymentId** (*string*) –

The deployment ID, which can be used with other requests to identify the deployment.

`create_instance` (**kwargs)

Creates an instance in a specified stack. For more information, see [Adding an Instance to a Layer](#) .

Required Permissions : To use this action, an IAM user must have a `Manage` permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.create_instance(
    StackId='string',
    LayerIds=[
        'string',
    ],
    InstanceType='string',
    AutoScalingType='load'|'timer',
    Hostname='string',
    Os='string',
    AmiId='string',
    SshKeyName='string',
    AvailabilityZone='string',
    VirtualizationType='string',
    SubnetId='string',
    Architecture='x86_64'|'i386',
    RootDeviceType='ebs'|'instance-store',
    BlockDeviceMappings=[
        {
            'DeviceName': 'string',
            'NoDevice': 'string',
            'VirtualName': 'string',
            'Ebs': {
                'SnapshotId': 'string',
                'Iops': 123,
                'VolumeSize': 123,
                'VolumeType': 'gp2'|'io1'|'standard',
                'DeleteOnTermination': True|False
            }
        },
    ],
    InstallUpdatesOnBoot=True|False,
    EbsOptimized=True|False,
    AgentVersion='string'
)
```

Parameters

- **StackId** (*string*) – [REQUIRED]

The stack ID.

- **LayerIds** (*list*) – [REQUIRED]

An array that contains the instance's layer IDs.

– (*string*) –

- **InstanceType** (*string*) – [REQUIRED]

The instance type, such as `t2.micro`. For a list of supported instance types, open the stack in the console, choose **Instances**, and choose **+ Instance**. The **Size** list contains the currently supported types. For more information, see [Instance Families and Types](#). The parameter values that you use to specify the various types are in the **API Name** column of the **Available Instance Types** table.

- **AutoScalingType** (*string*) – For load-based or time-based instances, the type. Windows stacks can use only time-based instances.

- **Hostname** (*string*) – The instance host name.

- **Os** (*string*) – The instance's operating system, which must be set to one of the following.

- A supported Linux operating system: An Amazon Linux version, such as Amazon Linux 2015.03, Red Hat Enterprise Linux 7, Ubuntu 12.04 LTS, or Ubuntu 14.04 LTS.
- Microsoft Windows Server 2012 R2 Base.
- A custom AMI: `Custom`.

For more information on the supported operating systems, see [AWS OpsWorks Operating Systems](#).

The default option is the current Amazon Linux version. If you set this parameter to `Custom`, you must use the `CreateInstance` action's `AmiId` parameter to specify the custom AMI that you want to use. For more information on the supported operating systems, see [Operating Systems](#). For more information on how to use custom AMIs with AWS OpsWorks, see [Using Custom AMIs](#).

- **AmiId** (*string*) – A custom AMI ID to be used to create the instance. The AMI should be based on one of the supported operating systems. For more information, see [Using Custom AMIs](#).

Note: If you specify a custom AMI, you must set `Os` to `Custom`.

- **SshKeyName** (*string*) – The instance's Amazon EC2 key-pair name.

- **AvailabilityZone** (*string*) – The instance Availability Zone. For more information, see [Regions and Endpoints](#).

- **VirtualizationType** (*string*) – The instance's virtualization type, `paravirtual` or `hvm`.

- **SubnetId** (*string*) – The ID of the instance's subnet. If the stack is running in a VPC, you can use this parameter to override the stack's default subnet ID value and direct AWS OpsWorks to launch the instance in a different subnet.

- **Architecture** (*string*) – The instance architecture. The default option is `x86_64`. Instance types do not necessarily support both architectures. For a list of the architectures that are supported by the different instance types, see [Instance Families and Types](#).

- **RootDeviceType** (*string*) – The instance root device type. For more information, see [Storage for the Root Device](#).

- **BlockDeviceMappings** (*list*) – An array of `BlockDeviceMapping` objects that specify the instance's block devices. For more information, see [Block Device Mapping](#).

– (*dict*) –

Describes a block device mapping. This data type maps directly to the Amazon EC2 [BlockDeviceMapping](#) data type.

* **DeviceName** (*string*) –

The device name that is exposed to the instance, such as `/dev/sdh`. For the root device, you can use the explicit device name or you can set this parameter to `ROOT_DEVICE` and AWS OpsWorks will provide the correct device name.

* **NoDevice** (*string*) –

Suppresses the specified device included in the AMI's block device mapping.

* **VirtualName** (*string*) –

The virtual device name. For more information, see [BlockDeviceMapping](#).

* **Ebs** (*dict*) –

An `EBSBlockDevice` that defines how to configure an Amazon EBS volume when the instance is launched.

· **SnapshotId** (*string*) –

The snapshot ID.

· **Iops** (*integer*) –

The number of I/O operations per second (IOPS) that the volume supports. For more information, see [EbsBlockDevice](#).

· **VolumeSize** (*integer*) –

The volume size, in GiB. For more information, see [EbsBlockDevice](#).

· **VolumeType** (*string*) –

The volume type. `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, and `standard` for Magnetic volumes.

· **DeleteOnTermination** (*boolean*) –

Whether the volume is deleted on instance termination.

- **InstallUpdatesOnBoot** (*boolean*) – Whether to install operating system and package updates when the instance boots. The default value is `true`. To control when updates are installed, set this value to `false`. You must then update your instances manually by using `CreateDeployment` to run the `update_dependencies` stack command or by manually running `yum` (Amazon Linux) or `apt-get` (Ubuntu) on the instances.

Note: We strongly recommend using the default value of `true` to ensure that your instances have the latest security updates.

- **EbsOptimized** (*boolean*) – Whether to create an Amazon EBS-optimized instance.
- **AgentVersion** (*string*) – The default AWS OpsWorks agent version. You have the following options:
 - `INHERIT` - Use the stack's default agent version setting.
 - `version_number` - Use the specified agent version. This value overrides the stack's default setting. To update the agent version, edit the instance configuration and specify a new version. AWS OpsWorks then automatically

installs that version on the instance.

The default setting is `INHERIT`. To specify an agent version, you must use the complete version number, not the abbreviated number shown on the console. For a list of available agent version numbers, call `DescribeAgentVersions`.

Return type dict

Returns

Response Syntax

```
{
    'InstanceId': 'string'
}
```

Response Structure

- (dict) –

Contains the response to a `CreateInstance` request.

- **InstanceId** (*string*) –

The instance ID.

create_layer (**kwargs)

Creates a layer. For more information, see [How to Create a Layer](#).

Note: You should use **CreateLayer** for noncustom layer types such as PHP App Server only if the stack does not have an existing layer of that type. A stack can have at most one instance of each noncustom layer; if you attempt to create a second instance, **CreateLayer** fails. A stack can have an arbitrary number of custom layers, so you can call **CreateLayer** as many times as you like for that layer type.

Required Permissions : To use this action, an IAM user must have a `Manage` permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.create_layer(
    StackId='string',
    Type='aws-flow-ruby'|'ecs-cluster'|'java-app'|'lb'|'web'|'php-app'|'rails-app'|'nodejs-',
    Name='string',
    Shortname='string',
    Attributes={
        'string': 'string'
    },
    CustomInstanceProfileArn='string',
    CustomJson='string',
    CustomSecurityGroupIds=[
        'string',
    ],
    Packages=[
        'string',
    ],
    VolumeConfigurations=[
        {
            'MountPoint': 'string',
            'RaidLevel': 123,
            'NumberOfDisks': 123,
            'Size': 123,
            'VolumeType': 'string',
```

```

        'Iops': 123
    },
],
EnableAutoHealing=True|False,
AutoAssignElasticIps=True|False,
AutoAssignPublicIps=True|False,
CustomRecipes={
    'Setup': [
        'string',
    ],
    'Configure': [
        'string',
    ],
    'Deploy': [
        'string',
    ],
    'Undeploy': [
        'string',
    ],
    'Shutdown': [
        'string',
    ]
},
InstallUpdatesOnBoot=True|False,
UseEbsOptimizedInstances=True|False,
LifecycleEventConfiguration={
    'Shutdown': {
        'ExecutionTimeout': 123,
        'DelayUntilElbConnectionsDrained': True|False
    }
}
)

```

Parameters

- **StackId** (*string*) – [REQUIRED]

The layer stack ID.

- **Type** (*string*) – [REQUIRED]

The layer type. A stack cannot have more than one built-in layer of the same type. It can have any number of custom layers.

- **Name** (*string*) – [REQUIRED]

The layer name, which is used by the console.

- **Shortname** (*string*) – [REQUIRED]

For custom layers only, use this parameter to specify the layer's short name, which is used internally by AWS OpsWorks and by Chef recipes. The short name is also used as the name for the directory where your app files are installed. It can have a maximum of 200 characters, which are limited to the alphanumeric characters, '-', '_', and '.'.

The built-in layers' short names are defined by AWS OpsWorks. For more information, see the [Layer Reference](#).

- **Attributes** (*dict*) – One or more user-defined key-value pairs to be added to the stack attributes.

To create a cluster layer, set the `EcsClusterArn` attribute to the cluster's ARN.

- (*string*) –
 - * (*string*) –
- **CustomInstanceProfileArn** (*string*) – The ARN of an IAM profile to be used for the layer’s EC2 instances. For more information about IAM ARNs, see [Using Identifiers](#) .
- **CustomJson** (*string*) – A JSON-formatted string containing custom stack configuration and deployment attributes to be installed on the layer’s instances. For more information, see [Using Custom JSON](#) .
- **CustomSecurityGroupIds** (*list*) – An array containing the layer custom security group IDs.
 - (*string*) –
- **Packages** (*list*) – An array of `Package` objects that describes the layer packages.
 - (*string*) –
- **VolumeConfigurations** (*list*) – A `VolumeConfigurations` object that describes the layer’s Amazon EBS volumes.
 - (*dict*) –
 - Describes an Amazon EBS volume configuration.
 - * **MountPoint** (*string*) – **[REQUIRED]**
 - The volume mount point. For example “/dev/sdh”.
 - * **RaidLevel** (*integer*) –
 - The volume [RAID level](#) .
 - * **NumberOfDisks** (*integer*) – **[REQUIRED]**
 - The number of disks in the volume.
 - * **Size** (*integer*) – **[REQUIRED]**
 - The volume size.
 - * **VolumeType** (*string*) –
 - The volume type:
 - `standard` - Magnetic
 - `io1` - Provisioned IOPS (SSD)
 - `gp2` - General Purpose (SSD)
 - * **Iops** (*integer*) –
 - For PIOPS volumes, the IOPS per disk.
- **EnableAutoHealing** (*boolean*) – Whether to disable auto healing for the layer.
- **AutoAssignElasticIps** (*boolean*) – Whether to automatically assign an [Elastic IP address](#) to the layer’s instances. For more information, see [How to Edit a Layer](#) .
- **AutoAssignPublicIps** (*boolean*) – For stacks that are running in a VPC, whether to automatically assign a public IP address to the layer’s instances. For more information, see [How to Edit a Layer](#) .
- **CustomRecipes** (*dict*) – A `LayerCustomRecipes` object that specifies the layer custom recipes.
 - **Setup** (*list*) –
 - An array of custom recipe names to be run following a `setup` event.
 - * (*string*) –
 - **Configure** (*list*) –
 - An array of custom recipe names to be run following a `configure` event.
 - * (*string*) –
 - **Deploy** (*list*) –

An array of custom recipe names to be run following a deploy event.

* (*string*) –

– **Undeploy** (*list*) –

An array of custom recipe names to be run following a undeploy event.

* (*string*) –

– **Shutdown** (*list*) –

An array of custom recipe names to be run following a shutdown event.

* (*string*) –

- **InstallUpdatesOnBoot** (*boolean*) – Whether to install operating system and package updates when the instance boots. The default value is `true`. To control when updates are installed, set this value to `false`. You must then update your instances manually by using `CreateDeployment` to run the `update_dependencies` stack command or by manually running `yum` (Amazon Linux) or `apt-get` (Ubuntu) on the instances.

Note: To ensure that your instances have the latest security updates, we strongly recommend using the default value of `true`.

- **UseEbsOptimizedInstances** (*boolean*) – Whether to use Amazon EBS-optimized instances.
- **LifecycleEventConfiguration** (*dict*) – A `LifeCycleEventConfiguration` object that you can use to configure the Shutdown event to specify an execution timeout and enable or disable Elastic Load Balancer connection draining.
 - **Shutdown** (*dict*) –

A `ShutdownEventConfiguration` object that specifies the Shutdown event configuration.

* **ExecutionTimeout** (*integer*) –

The time, in seconds, that AWS OpsWorks will wait after triggering a Shutdown event before shutting down an instance.

* **DelayUntilElbConnectionsDrained** (*boolean*) –

Whether to enable Elastic Load Balancing connection draining. For more information, see [Connection Draining](#)

Return type dict

Returns

Response Syntax

```
{
    'LayerId': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a `CreateLayer` request.

– **LayerId** (*string*) –

The layer ID.

create_stack (***kwargs*)

Creates a new stack. For more information, see [Create a New Stack](#).

Required Permissions : To use this action, an IAM user must have an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.create_stack(  
    Name='string',  
    Region='string',  
    VpcId='string',  
    Attributes={  
        'string': 'string'  
    },  
    ServiceRoleArn='string',  
    DefaultInstanceProfileArn='string',  
    DefaultOs='string',  
    HostnameTheme='string',  
    DefaultAvailabilityZone='string',  
    DefaultSubnetId='string',  
    CustomJson='string',  
    ConfigurationManager={  
        'Name': 'string',  
        'Version': 'string'  
    },  
    ChefConfiguration={  
        'ManageBerkshelf': True|False,  
        'BerkshelfVersion': 'string'  
    },  
    UseCustomCookbooks=True|False,  
    UseOpsworksSecurityGroups=True|False,  
    CustomCookbooksSource={  
        'Type': 'git'|'svn'|'archive'|'s3',  
        'Url': 'string',  
        'Username': 'string',  
        'Password': 'string',  
        'SshKey': 'string',  
        'Revision': 'string'  
    },  
    DefaultSshKeyName='string',  
    DefaultRootDeviceType='ebs'|'instance-store',  
    AgentVersion='string'  
)
```

Parameters

- **Name** (*string*) – [REQUIRED]
The stack name.
- **Region** (*string*) – [REQUIRED]
The stack’s AWS region, such as “us-east-1”. For more information about Amazon regions, see [Regions and Endpoints](#) .
- **VpcId** (*string*) – The ID of the VPC that the stack is to be launched into. The VPC must be in the stack’s region. All instances are launched into this VPC. You cannot change the ID later.
 - If your account supports EC2-Classic, the default value is no VPC .
 - If your account does not support EC2-Classic, the default value is the default VPC for the specified region.

If the VPC ID corresponds to a default VPC and you have specified either the `DefaultAvailabilityZone` or the `DefaultSubnetId` parameter only, AWS OpsWorks infers the value of the other parameter. If you specify neither parameter, AWS OpsWorks sets these parameters to the first valid Availability

Zone for the specified region and the corresponding default VPC subnet ID, respectively.

If you specify a nondefault VPC ID, note the following:

- It must belong to a VPC in your account that is in the specified region.
- You must specify a value for `DefaultSubnetId`.

For more information on how to use AWS OpsWorks with a VPC, see [Running a Stack in a VPC](#). For more information on default VPC and EC2-Classical, see [Supported Platforms](#).

- **Attributes** (*dict*) – One or more user-defined key-value pairs to be added to the stack attributes.
 - (*string*) –
 - * (*string*) –
- **ServiceRoleArn** (*string*) – [REQUIRED]

The stack's AWS Identity and Access Management (IAM) role, which allows AWS OpsWorks to work with AWS resources on your behalf. You must set this parameter to the Amazon Resource Name (ARN) for an existing IAM role. For more information about IAM ARNs, see [Using Identifiers](#).

- **DefaultInstanceProfileArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of an IAM profile that is the default profile for all of the stack's EC2 instances. For more information about IAM ARNs, see [Using Identifiers](#).

- **DefaultOs** (*string*) – The stack's default operating system, which is installed on every instance unless you specify a different operating system when you create the instance. You can specify one of the following.
 - A supported Linux operating system: An Amazon Linux version, such as Amazon Linux 2015.03, Red Hat Enterprise Linux 7, Ubuntu 12.04 LTS, or Ubuntu 14.04 LTS.
 - Microsoft Windows Server 2012 R2 Base.
 - A custom AMI: `Custom`. You specify the custom AMI you want to use when you create instances. For more information, see [Using Custom AMIs](#).

The default option is the current Amazon Linux version. For more information on the supported operating systems, see [AWS OpsWorks Operating Systems](#).

- **HostnameTheme** (*string*) – The stack's host name theme, with spaces replaced by underscores. The theme is used to generate host names for the stack's instances. By default, `HostnameTheme` is set to `Layer_Dependent`, which creates host names by appending integers to the layer's short name. The other themes are:
 - `Baked_Goods`
 - `Clouds`
 - `Europe_Cities`
 - `Fruits`
 - `Greek_Deities`
 - `Legendary_creatures_from_Japan`
 - `Planets_and_Moons`
 - `Roman_Deities`
 - `Scottish_Islands`
 - `US_Cities`
 - `Wild_Cats`

To obtain a generated host name, call `GetHostNameSuggestion`, which returns a host name based on the current theme.

- **DefaultAvailabilityZone** (*string*) – The stack's default Availability Zone, which

must be in the specified region. For more information, see [Regions and Endpoints](#). If you also specify a value for `DefaultSubnetId`, the subnet must be in the same zone. For more information, see the `VpcId` parameter description.

- **DefaultSubnetId** (*string*) – The stack’s default VPC subnet ID. This parameter is required if you specify a value for the `VpcId` parameter. All instances are launched into this subnet unless you specify otherwise when you create the instance. If you also specify a value for `DefaultAvailabilityZone`, the subnet must be in that zone. For information on default values and when this parameter is required, see the `VpcId` parameter description.
- **CustomJson** (*string*) – A string that contains user-defined, custom JSON. It can be used to override the corresponding default stack configuration attribute values or to pass data to recipes. The string should be in the following escape characters such as “”:

```
{\"key1\": \"value1\", \"key2\":  
  \"value2\", ...}
```

For more information on custom JSON, see [Use Custom JSON to Modify the Stack Configuration Attributes](#).

- **ConfigurationManager** (*dict*) – The configuration manager. When you clone a stack we recommend that you use the configuration manager to specify the Chef version: 0.9, 11.4, or 11.10. The default value is currently 11.4.
 - **Name** (*string*) –
The name. This parameter must be set to “Chef”.
 - **Version** (*string*) –
The Chef version. This parameter must be set to 0.9, 11.4, or 11.10. The default value is 11.4.
- **ChefConfiguration** (*dict*) – A `ChefConfiguration` object that specifies whether to enable Berkshelf and the Berkshelf version on Chef 11.10 stacks. For more information, see [Create a New Stack](#).
 - **ManageBerkshelf** (*boolean*) –
Whether to enable Berkshelf.
 - **BerkshelfVersion** (*string*) –
The Berkshelf version.
- **UseCustomCookbooks** (*boolean*) – Whether the stack uses custom cookbooks.
- **UseOpsworksSecurityGroups** (*boolean*) – Whether to associate the AWS OpsWorks built-in security groups with the stack’s layers.

AWS OpsWorks provides a standard set of built-in security groups, one for each layer, which are associated with layers by default. With `UseOpsworksSecurityGroups` you can instead provide your own custom security groups. `UseOpsworksSecurityGroups` has the following settings:

- True - AWS OpsWorks automatically associates the appropriate built-in security group with each layer (default setting). You can associate additional security groups with a layer after you create it, but you cannot delete the built-in security group.
- False - AWS OpsWorks does not associate built-in security groups with layers. You must create appropriate EC2 security groups and associate a security group with each layer that you create. However, you can still manually associate a built-in security group with a layer on creation; custom security groups are required only for those layers that need custom settings.

For more information, see [Create a New Stack](#) .

- **CustomCookbooksSource** (*dict*) – Contains the information required to retrieve an app or cookbook from a repository. For more information, see [Creating Apps](#) or [Custom Recipes and Cookbooks](#) .

- **Type** (*string*) –

The repository type.

- **Url** (*string*) –

The source URL.

- **Username** (*string*) –

This parameter depends on the repository type.

- * For Amazon S3 bundles, set `Username` to the appropriate IAM access key ID.
- * For HTTP bundles, Git repositories, and Subversion repositories, set `Username` to the user name.

- **Password** (*string*) –

When included in a request, the parameter depends on the repository type.

- * For Amazon S3 bundles, set `Password` to the appropriate IAM secret access key.
- * For HTTP bundles and Subversion repositories, set `Password` to the password.

For more information on how to safely handle IAM credentials, see <http://docs.aws.amazon.com/general/latest/gr/aws-access-keys-best-practices.html> .

In responses, AWS OpsWorks returns `*****FILTERED*****` instead of the actual value.

- **SshKey** (*string*) –

In requests, the repository’s SSH key.

In responses, AWS OpsWorks returns `*****FILTERED*****` instead of the actual value.

- **Revision** (*string*) –

The application’s version. AWS OpsWorks enables you to easily deploy new versions of an application. One of the simplest approaches is to have branches or revisions in your repository that represent different versions that can potentially be deployed.

- **DefaultSshKeyName** (*string*) – A default Amazon EC2 key pair name. The default value is none. If you specify a key pair name, AWS OpsWorks installs the public key on the instance and you can use the private key with an SSH client to log in to the instance. For more information, see [Using SSH to Communicate with an Instance](#) and [Managing SSH Access](#) . You can override this setting by specifying a different key pair, or no key pair, when you [create an instance](#) .
- **DefaultRootDeviceType** (*string*) – The default root device type. This value is the default for all instances in the stack, but you can override it when you create an instance. The default option is `instance-store` . For more information, see [Storage for the Root Device](#) .
- **AgentVersion** (*string*) – The default AWS OpsWorks agent version. You have the following options:
 - Auto-update - Set this parameter to `LATEST` . AWS OpsWorks automatically installs new agent versions on the stack’s instances as soon as they are available.

- Fixed version - Set this parameter to your preferred agent version. To update the agent version, you must edit the stack configuration and specify a new version. AWS OpsWorks then automatically installs that version on the stack's instances.

The default setting is `LATEST` . To specify an agent version, you must use the complete version number, not the abbreviated number shown on the console. For a list of available agent version numbers, call `DescribeAgentVersions` .

Note: You can also specify an agent version when you create or update an instance, which overrides the stack's default setting.

Return type dict

Returns

Response Syntax

```
{
    'StackId': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a `CreateStack` request.

- **StackId** (*string*) –

The stack ID, which is an opaque string that you use to identify the stack when performing actions such as `DescribeStacks` .

create_user_profile (***kwargs*)

Creates a new user profile.

Required Permissions : To use this action, an IAM user must have an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.create_user_profile(
    IamUserArn='string',
    SshUsername='string',
    SshPublicKey='string',
    AllowSelfManagement=True|False
)
```

Parameters

- **IamUserArn** (*string*) – **[REQUIRED]**

The user's IAM ARN.

- **SshUsername** (*string*) – The user's SSH user name. The allowable characters are [a-z], [A-Z], [0-9], '-', and '_'. If the specified name includes other punctuation marks, AWS OpsWorks removes them. For example, `my.name` will be changed to `myname` . If you do not specify an SSH user name, AWS OpsWorks generates one from the IAM user name.
- **SshPublicKey** (*string*) – The user's public SSH key.
- **AllowSelfManagement** (*boolean*) – Whether users can specify their own SSH public key through the My Settings page. For more information, see [Setting an IAM User's Public SSH Key](#) .

Return type dict

Returns**Response Syntax**

```
{
    'IamUserArn': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a `CreateUserProfile` request.

- **IamUserArn** (*string*) –

The user's IAM ARN.

delete_app (***kwargs*)

Deletes a specified app.

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.delete_app(
    AppId='string'
)
```

Parameters **AppId** (*string*) – [REQUIRED]

The app ID.

Returns None

delete_instance (***kwargs*)

Deletes a specified instance, which terminates the associated Amazon EC2 instance. You must stop an instance before you can delete it.

For more information, see [Deleting Instances](#) .

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.delete_instance(
    InstanceId='string',
    DeleteElasticIp=True|False,
    DeleteVolumes=True|False
)
```

Parameters

- **InstanceId** (*string*) – [REQUIRED]

The instance ID.

- **DeleteElasticIp** (*boolean*) – Whether to delete the instance Elastic IP address.
- **DeleteVolumes** (*boolean*) – Whether to delete the instance's Amazon EBS volumes.

Returns None

delete_layer (**kwargs)

Deletes a specified layer. You must first stop and then delete all associated instances or unassign registered instances. For more information, see [How to Delete a Layer](#) .

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.delete_layer(  
    LayerId='string'  
)
```

Parameters **LayerId** (*string*) – [REQUIRED]

The layer ID.

Returns None

delete_stack (**kwargs)

Deletes a specified stack. You must first delete all instances, layers, and apps or deregister registered instances. For more information, see [Shut Down a Stack](#) .

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.delete_stack(  
    StackId='string'  
)
```

Parameters **StackId** (*string*) – [REQUIRED]

The stack ID.

Returns None

delete_user_profile (**kwargs)

Deletes a user profile.

Required Permissions : To use this action, an IAM user must have an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.delete_user_profile(  
    IamUserArn='string'  
)
```

Parameters **IamUserArn** (*string*) – [REQUIRED]

The user's IAM ARN.

Returns None

deregister_ecs_cluster (**kwargs)

Deregisters a specified Amazon ECS cluster from a stack. For more information, see [Resource Management](#) .

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack or an attached policy that explicitly grants permissions. For more information on user permissions, see <http://docs.aws.amazon.com/opsworks/latest/userguide/opsworks-security-users.html> .

Request Syntax

```
response = client.deregister_ecs_cluster(
    EcsClusterArn='string'
)
```

Parameters `EcsClusterArn` (*string*) – [REQUIRED]

The cluster's ARN.

Returns None

deregister_elastic_ip (***kwargs*)

Deregisters a specified Elastic IP address. The address can then be registered by another stack. For more information, see [Resource Management](#) .

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.deregister_elastic_ip(
    ElasticIp='string'
)
```

Parameters `ElasticIp` (*string*) – [REQUIRED]

The Elastic IP address.

Returns None

deregister_instance (***kwargs*)

Deregister a registered Amazon EC2 or on-premises instance. This action removes the instance from the stack and returns it to your control. This action can not be used with instances that were created with AWS OpsWorks.

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.deregister_instance(
    InstanceId='string'
)
```

Parameters `InstanceId` (*string*) – [REQUIRED]

The instance ID.

Returns None

deregister_rds_db_instance (***kwargs*)

Deregisters an Amazon RDS instance.

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.deregister_rds_db_instance(  
    RdsDbInstanceArn='string'  
)
```

Parameters `RdsDbInstanceArn` (*string*) – [REQUIRED]

The Amazon RDS instance’s ARN.

Returns None**deregister_volume** (***kwargs*)

Deregisters an Amazon EBS volume. The volume can then be registered by another stack. For more information, see [Resource Management](#).

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.deregister_volume(  
    VolumeId='string'  
)
```

Parameters `VolumeId` (*string*) – [REQUIRED]

The AWS OpsWorks volume ID, which is the GUID that AWS OpsWorks assigned to the instance when you registered the volume with the stack, not the Amazon EC2 volume ID.

Returns None**describe_agent_versions** (***kwargs*)

Describes the available AWS OpsWorks agent versions. You must specify a stack ID or a configuration manager. `DescribeAgentVersions` returns a list of available agent versions for the specified stack or configuration manager.

Request Syntax

```
response = client.describe_agent_versions(  
    StackId='string',  
    ConfigurationManager={  
        'Name': 'string',  
        'Version': 'string'  
    }  
)
```

Parameters

- **StackId** (*string*) – The stack ID.
- **ConfigurationManager** (*dict*) – The configuration manager.
 - **Name** (*string*) –
The name. This parameter must be set to “Chef”.
 - **Version** (*string*) –

The Chef version. This parameter must be set to 0.9, 11.4, or 11.10. The default value is 11.4.

Return type dict

Returns

Response Syntax

```
{
  'AgentVersions': [
    {
      'Version': 'string',
      'ConfigurationManager': {
        'Name': 'string',
        'Version': 'string'
      }
    },
  ]
}
```

Response Structure

- (*dict*) –

Contains the response to a `DescribeAgentVersions` request.

- **AgentVersions** (*list*) –

The agent versions for the specified stack or configuration manager. Note that this value is the complete version number, not the abbreviated number used by the console.

- * (*dict*) –

Describes an agent version.

- **Version** (*string*) –

The agent version.

- **ConfigurationManager** (*dict*) –

The configuration manager.

- **Name** (*string*) –

The name. This parameter must be set to “Chef”.

- **Version** (*string*) –

The Chef version. This parameter must be set to 0.9, 11.4, or 11.10. The default value is 11.4.

`describe_apps` (***kwargs*)

Requests a description of a specified set of apps.

Note: You must specify at least one of the parameters.

Required Permissions : To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.describe_apps(
    StackId='string',
    AppIds=[
        'string',
    ]
)
```

Parameters

- **StackId** (*string*) – The app stack ID. If you use this parameter, DescribeApps returns a description of the apps in the specified stack.
- **AppIds** (*list*) – An array of app IDs for the apps to be described. If you use this parameter, DescribeApps returns a description of the specified apps. Otherwise, it returns a description of every app.
 - (*string*) –

Return type dict**Returns****Response Syntax**

```
{
  'Apps': [
    {
      'AppId': 'string',
      'StackId': 'string',
      'Shortname': 'string',
      'Name': 'string',
      'Description': 'string',
      'DataSources': [
        {
          'Type': 'string',
          'Arn': 'string',
          'DatabaseName': 'string'
        },
      ],
      'Type': 'aws-flow-ruby'|'java'|'rails'|'php'|'nodejs'|'static'|'other',
      'AppSource': {
        'Type': 'git'|'svn'|'archive'|'s3',
        'Url': 'string',
        'Username': 'string',
        'Password': 'string',
        'SshKey': 'string',
        'Revision': 'string'
      },
      'Domains': [
        'string',
      ],
      'EnableSsl': True|False,
      'SslConfiguration': {
        'Certificate': 'string',
        'PrivateKey': 'string',
        'Chain': 'string'
      },
      'Attributes': {
        'string': 'string'
      },
      'CreatedAt': 'string',
      'Environment': [
        {
          'Key': 'string',
          'Value': 'string',
          'Secure': True|False
        },
      ],
    },
  ]
}
```

```
}

```

Response Structure

- *(dict)* –

Contains the response to a `DescribeApps` request.

- **Apps** (*list*) –

An array of `App` objects that describe the specified apps.

- * *(dict)* –

A description of the app.

- **AppId** (*string*) –

The app ID.

- **StackId** (*string*) –

The app stack ID.

- **Shortname** (*string*) –

The app's short name.

- **Name** (*string*) –

The app name.

- **Description** (*string*) –

A description of the app.

- **DataSources** (*list*) –

The app's data sources.

- *(dict)* –

Describes an app's data source.

- **Type** (*string*) –

The data source's type, `AutoSelectOpsworksMySQLInstance`, `OpsworksMySQLInstance`, or `RdsDbInstance`.

- **Arn** (*string*) –

The data source's ARN.

- **DatabaseName** (*string*) –

The database name.

- **Type** (*string*) –

The app type.

- **AppSource** (*dict*) –

A `Source` object that describes the app repository.

- **Type** (*string*) –

The repository type.

- **Url** (*string*) –

The source URL.

- **Username** (*string*) –

This parameter depends on the repository type.

- For Amazon S3 bundles, set `Username` to the appropriate IAM access key ID.

- For HTTP bundles, Git repositories, and Subversion repositories, set `Username` to the user name.

- **Password** (*string*) –

When included in a request, the parameter depends on the repository type.

- For Amazon S3 bundles, set `Password` to the appropriate IAM secret access key.
- For HTTP bundles and Subversion repositories, set `Password` to the password.

For more information on how to safely handle IAM credentials, see <http://docs.aws.amazon.com/general/latest/gr/aws-access-keys-best-practices.html>.

In responses, AWS OpsWorks returns *****FILTERED***** instead of the actual value.

- **SshKey** (*string*) –

In requests, the repository's SSH key.

In responses, AWS OpsWorks returns *****FILTERED***** instead of the actual value.

- **Revision** (*string*) –

The application's version. AWS OpsWorks enables you to easily deploy new versions of an application. One of the simplest approaches is to have branches or revisions in your repository that represent different versions that can potentially be deployed.

- **Domains** (*list*) –

The app vhost settings with multiple domains separated by commas. For example: `'www.example.com, example.com'`

- (*string*) –

- **EnableSsl** (*boolean*) –

Whether to enable SSL for the app.

- **SslConfiguration** (*dict*) –

An `SslConfiguration` object with the SSL configuration.

- **Certificate** (*string*) –

The contents of the certificate's `domain.crt` file.

- **PrivateKey** (*string*) –

The private key; the contents of the certificate's `domain.key` file.

- **Chain** (*string*) –

Optional. Can be used to specify an intermediate certificate authority key or client authentication.

- **Attributes** (*dict*) –

The stack attributes.

- (*string*) –

- (*string*) –

- **CreatedAt** (*string*) –

When the app was created.

- **Environment** (*list*) –

An array of `EnvironmentVariable` objects that specify environment variables to be associated with the app. After you deploy the app, these variables are defined on the associated app server instances. For more information, see [Environment Variables](#).

Note: There is no specific limit on the number of environment variables. However, the size of the associated data structure - which includes the variables' names, values, and protected flag values - cannot exceed 10 KB (10240 Bytes). This limit should accommodate most if not all use cases, but if you do exceed it, you will cause an exception (API) with an "Environment: is too large (maximum is 10KB)" message.

- *(dict)* –

Represents an app's environment variable.

- **Key** (*string*) –

(Required) The environment variable's name, which can consist of up to 64 characters and must be specified. The name can contain upper- and lowercase letters, numbers, and underscores (`_`), but it must start with a letter or underscore.

- **Value** (*string*) –

(Optional) The environment variable's value, which can be left empty. If you specify a value, it can contain up to 256 characters, which must all be printable.

- **Secure** (*boolean*) –

(Optional) Whether the variable's value will be returned by the `DescribeApps` action. To conceal an environment variable's value, set `Secure` to `true`. `DescribeApps` then returns `*****FILTERED*****` instead of the actual value. The default value for `Secure` is `false`.

describe_commands (***kwargs*)

Describes the results of specified commands.

Note: You must specify at least one of the parameters.

Required Permissions : To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.describe_commands(
    DeploymentId='string',
    InstanceId='string',
    CommandIds=[
        'string',
    ]
)
```

Parameters

- **DeploymentId** (*string*) – The deployment ID. If you include this parameter, `DescribeCommands` returns a description of the commands associated with the specified deployment.
- **InstanceId** (*string*) – The instance ID. If you include this parameter, `DescribeCommands` returns a description of the commands associated with the specified instance.
- **CommandIds** (*list*) – An array of command IDs. If you include this parameter, `DescribeCommands` returns a description of the specified commands. Otherwise, it returns a description of every command.
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'Commands': [
    {
      'CommandId': 'string',
      'InstanceId': 'string',
      'DeploymentId': 'string',
      'CreatedAt': 'string',
      'AcknowledgedAt': 'string',
      'CompletedAt': 'string',
      'Status': 'string',
      'ExitCode': 123,
      'LogUrl': 'string',
      'Type': 'string'
    },
  ]
}
```

Response Structure

- (*dict*) –

Contains the response to a `DescribeCommands` request.

- **Commands** (*list*) –

An array of `Command` objects that describe each of the specified commands.

- * (*dict*) –

Describes a command.

- **CommandId** (*string*) –
The command ID.
- **InstanceId** (*string*) –
The ID of the instance where the command was executed.
- **DeploymentId** (*string*) –
The command deployment ID.
- **CreatedAt** (*string*) –
Date and time when the command was run.
- **AcknowledgedAt** (*string*) –
Date and time when the command was acknowledged.
- **CompletedAt** (*string*) –

- Date when the command completed.
- **Status** (*string*) –
 - The command status:
 - failed
 - successful
 - skipped
 - pending
- **ExitCode** (*integer*) –
 - The command exit code.
- **LogUrl** (*string*) –
 - The URL of the command log.
- **Type** (*string*) –
 - The command type:
 - deploy
 - rollback
 - start
 - stop
 - restart
 - undeploy
 - update_dependencies
 - install_dependencies
 - update_custom_cookbooks
 - execute_recipes

describe_deployments (***kwargs*)

Requests a description of a specified set of deployments.

Note: You must specify at least one of the parameters.

Required Permissions : To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.describe_deployments(
    StackId='string',
    AppId='string',
    DeploymentIds=[
        'string',
    ]
)
```

Parameters

- **StackId** (*string*) – The stack ID. If you include this parameter, `DescribeDeployments` returns a description of the commands associated with the specified stack.
- **AppId** (*string*) – The app ID. If you include this parameter, `DescribeDeployments` returns a description of the commands associated with the specified app.
- **DeploymentIds** (*list*) – An array of deployment IDs to be described. If you include this parameter, `DescribeDeployments` returns a description of the specified deployments. Otherwise, it returns a description of every deployment.

– (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'Deployments': [
    {
      'DeploymentId': 'string',
      'StackId': 'string',
      'AppId': 'string',
      'CreatedAt': 'string',
      'CompletedAt': 'string',
      'Duration': 123,
      'IamUserArn': 'string',
      'Comment': 'string',
      'Command': {
        'Name': 'install_dependencies'|'update_dependencies'|'update_custom_resources',
        'Args': {
          'string': [
            'string',
          ]
        }
      },
      'Status': 'string',
      'CustomJson': 'string',
      'InstanceIds': [
        'string',
      ]
    },
  ]
}
```

Response Structure

- (*dict*) –

Contains the response to a `DescribeDeployments` request.

– **Deployments** (*list*) –

An array of `Deployment` objects that describe the deployments.

- * (*dict*) –

Describes a deployment of a stack or app.

- **DeploymentId** (*string*) –

The deployment ID.

- **StackId** (*string*) –

The stack ID.

- **AppId** (*string*) –

The app ID.

- **CreatedAt** (*string*) –

Date when the deployment was created.

- **CompletedAt** (*string*) –

Date when the deployment completed.

- **Duration** (*integer*) –

The deployment duration.

- **IamUserArn** (*string*) –

The user’s IAM ARN.

- **Comment** (*string*) –

A user-defined comment.

- **Command** (*dict*) –

Used to specify a stack or deployment command.

- **Name** (*string*) –

Specifies the operation. You can specify only one command.

For stacks, the following commands are available:

- **execute_recipes** : Execute one or more recipes. To specify the recipes, set an **Args** parameter named **recipes** to the list of recipes to be executed. For example, to execute `phpapp::appsetup`, set **Args** to `{"recipes": ["phpapp::appsetup"]}`.
- **install_dependencies** : Install the stack’s dependencies.
- **update_custom_cookbooks** : Update the stack’s custom cookbooks.
- **update_dependencies** : Update the stack’s dependencies.

Note: The `update_dependencies` and `install_dependencies` commands are supported only for Linux instances. You can run the commands successfully on Windows instances, but they do nothing.

For apps, the following commands are available:

- **deploy** : Deploy an app. Ruby on Rails apps have an optional **Args** parameter named **migrate**. Set **Args** to `{"migrate": ["true"]}` to migrate the database. The default setting is `{"migrate": ["false"]}`.
- **rollback** Roll the app back to the previous version. When you update an app, AWS OpsWorks stores the previous version, up to a maximum of five versions. You can use this command to roll an app back as many as four versions.
- **start** : Start the app’s web or application server.
- **stop** : Stop the app’s web or application server.
- **restart** : Restart the app’s web or application server.
- **undeploy** : Undeploy the app.
- **Args** (*dict*) –

The arguments of those commands that take arguments. It should be set to a JSON object with the following format:

```
{ "arg_name1" : [ "value1",
  "value2", ... ], "arg_name2" :
  [ "value1", "value2", ... ], ... }
```

The `update_dependencies` command takes two arguments:

- **upgrade_os_to** - Specifies the desired Amazon Linux version for instances whose OS you want to upgrade, such as Amazon Linux 2014.09 . You must also set the

`allow_reboot` argument to `true`.

- `allow_reboot` - Specifies whether to allow AWS OpsWorks to reboot the instances if necessary, after installing the updates. This argument can be set to either `true` or `false`. The default value is `false`.

For example, to upgrade an instance to Amazon Linux 2014.09, set `Args` to the following.

```
{ "upgrade_os_to": ["Amazon
Linux 2014.09"],
  "allow_reboot": ["true"] }
```

- *(string)* –
- *(list)* –
- *(string)* –
- **Status** *(string)* –

The deployment status:

- `running`
- `successful`
- `failed`
- **CustomJson** *(string)* –

A string that contains user-defined custom JSON. It can be used to override the corresponding default stack configuration attribute values for stack or to pass data to recipes. The string should be in the following format and must escape characters such as `""`:

```
{"key1\": \"value1\",
 \"key2\": \"value2\", ...}
```

For more information on custom JSON, see [Use Custom JSON to Modify the Stack Configuration Attributes](#).

- **InstanceIds** *(list)* –

The IDs of the target instances.

- *(string)* –

describe_ecs_clusters (***kwargs*)

Describes Amazon ECS clusters that are registered with a stack. If you specify only a stack ID, you can use the `MaxResults` and `NextToken` parameters to paginate the response. However, AWS OpsWorks currently supports only one cluster per layer, so the result set has a maximum of one element.

Required Permissions : To use this action, an IAM user must have a `Show`, `Deploy`, or `Manage` permissions level for the stack or an attached policy that explicitly grants permission. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.describe_ecs_clusters(
    EcsClusterArns=[
        'string',
    ],
    StackId='string',
    NextToken='string',
    MaxResults=123
)
```

Parameters

- **EcsClusterArns** *(list)* – A list of ARNs, one for each cluster to be described.
 - *(string)* –

- **StackId** (*string*) – A stack ID. `DescribeEcsClusters` returns a description of the cluster that is registered with the stack.
- **NextToken** (*string*) – If the previous paginated request did not return all of the remaining results, the response object's `NextToken` parameter value is set to a token. To retrieve the next set of results, call `DescribeEcsClusters` again and assign that token to the request object's `NextToken` parameter. If there are no remaining results, the previous response object's `NextToken` parameter is set to `null`.
- **MaxResults** (*integer*) – To receive a paginated response, use this parameter to specify the maximum number of results to be returned with a single call. If the number of available results exceeds this maximum, the response includes a `NextToken` value that you can assign to the `NextToken` request parameter to get the next set of results.

Return type dict

Returns

Response Syntax

```
{
  'EcsClusters': [
    {
      'EcsClusterArn': 'string',
      'EcsClusterName': 'string',
      'StackId': 'string',
      'RegisteredAt': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a `DescribeEcsClusters` request.

- **EcsClusters** (*list*) –

A list of `EcsCluster` objects containing the cluster descriptions.

- * (*dict*) –

Describes a registered Amazon ECS cluster.

- **EcsClusterArn** (*string*) –

The cluster's ARN.

- **EcsClusterName** (*string*) –

The cluster name.

- **StackId** (*string*) –

The stack ID.

- **RegisteredAt** (*string*) –

The time and date that the cluster was registered with the stack.

- **NextToken** (*string*) –

If a paginated request does not return all of the remaining results, this parameter is set to a token that you can assign to the request object's `NextToken` parameter to retrieve the next set of results. If the previ-

ous paginated request returned all of the remaining results, this parameter is set to `null`.

describe_elastic_ips (**kwargs)

Describes Elastic IP addresses.

Note: You must specify at least one of the parameters.

Required Permissions : To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.describe_elastic_ips(
    InstanceId='string',
    StackId='string',
    Ips=[
        'string',
    ]
)
```

Parameters

- **InstanceId** (*string*) – The instance ID. If you include this parameter, `DescribeElasticIps` returns a description of the Elastic IP addresses associated with the specified instance.
- **StackId** (*string*) – A stack ID. If you include this parameter, `DescribeElasticIps` returns a description of the Elastic IP addresses that are registered with the specified stack.
- **Ips** (*list*) – An array of Elastic IP addresses to be described. If you include this parameter, `DescribeElasticIps` returns a description of the specified Elastic IP addresses. Otherwise, it returns a description of every Elastic IP address.
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'ElasticIps': [
    {
      'Ip': 'string',
      'Name': 'string',
      'Domain': 'string',
      'Region': 'string',
      'InstanceId': 'string'
    },
  ]
}
```

Response Structure

- (*dict*) –
 - Contains the response to a `DescribeElasticIps` request.
 - **ElasticIps** (*list*) –
 - An `ElasticIps` object that describes the specified Elastic IP addresses.

* (*dict*) –

Describes an Elastic IP address.

- **Ip** (*string*) –

The IP address.

- **Name** (*string*) –

The name.

- **Domain** (*string*) –

The domain.

- **Region** (*string*) –

The AWS region. For more information, see [Regions and Endpoints](#).

- **InstanceId** (*string*) –

The ID of the instance that the address is attached to.

describe_elastic_load_balancers (***kwargs*)

Describes a stack's Elastic Load Balancing instances.

Note: You must specify at least one of the parameters.

Required Permissions : To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.describe_elastic_load_balancers(
    StackId='string',
    LayerIds=[
        'string',
    ]
)
```

Parameters

- **StackId** (*string*) – A stack ID. The action describes the stack's Elastic Load Balancing instances.
- **LayerIds** (*list*) – A list of layer IDs. The action describes the Elastic Load Balancing instances for the specified layers.
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'ElasticLoadBalancers': [
        {
            'ElasticLoadBalancerName': 'string',
            'Region': 'string',
            'DnsName': 'string',
            'StackId': 'string',
            'LayerId': 'string',
            'VpcId': 'string',
            'AvailabilityZones': [
```

```
        'string',
    ],
    'SubnetIds': [
        'string',
    ],
    'Ec2InstanceIds': [
        'string',
    ]
},
]
```

Response Structure

- *(dict)* –

Contains the response to a `DescribeElasticLoadBalancers` request.

- **ElasticLoadBalancers** (*list*) –

A list of `ElasticLoadBalancer` objects that describe the specified Elastic Load Balancing instances.

- * *(dict)* –

Describes an Elastic Load Balancing instance.

- **ElasticLoadBalancerName** (*string*) –

The Elastic Load Balancing instance’s name.

- **Region** (*string*) –

The instance’s AWS region.

- **DnsName** (*string*) –

The instance’s public DNS name.

- **StackId** (*string*) –

The ID of the stack that the instance is associated with.

- **LayerId** (*string*) –

The ID of the layer that the instance is attached to.

- **VpcId** (*string*) –

The VPC ID.

- **AvailabilityZones** (*list*) –

A list of Availability Zones.

- (*string*) –

- **SubnetIds** (*list*) –

A list of subnet IDs, if the stack is running in a VPC.

- (*string*) –

- **Ec2InstanceIds** (*list*) –

A list of the EC2 instances that the Elastic Load Balancing instance is managing traffic for.

- (*string*) –

describe_instances (***kwargs*)

Requests a description of a set of instances.

Note: You must specify at least one of the parameters.

Required Permissions : To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.describe_instances(
    StackId='string',
    LayerId='string',
    InstanceIds=[
        'string',
    ]
)
```

Parameters

- **StackId** (*string*) – A stack ID. If you use this parameter, DescribeInstances returns descriptions of the instances associated with the specified stack.
- **LayerId** (*string*) – A layer ID. If you use this parameter, DescribeInstances returns descriptions of the instances associated with the specified layer.
- **InstanceIds** (*list*) – An array of instance IDs to be described. If you use this parameter, DescribeInstances returns a description of the specified instances. Otherwise, it returns a description of every instance.
– (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'Instances': [
        {
            'AgentVersion': 'string',
            'AmiId': 'string',
            'Architecture': 'x86_64'|'i386',
            'AutoScalingType': 'load'|'timer',
            'AvailabilityZone': 'string',
            'BlockDeviceMappings': [
                {
                    'DeviceName': 'string',
                    'NoDevice': 'string',
                    'VirtualName': 'string',
                    'Ebs': {
                        'SnapshotId': 'string',
                        'Iops': 123,
                        'VolumeSize': 123,
                        'VolumeType': 'gp2'|'io1'|'standard',
                        'DeleteOnTermination': True|False
                    }
                }
            ],
            'CreatedAt': 'string',
            'EbsOptimized': True|False,
            'Ec2InstanceId': 'string',
            'EcsClusterArn': 'string',
            'EcsContainerInstanceArn': 'string',
            'ElasticIp': 'string',
```

```
        'Hostname': 'string',
        'InfrastructureClass': 'string',
        'InstallUpdatesOnBoot': True|False,
        'InstanceId': 'string',
        'InstanceProfileArn': 'string',
        'InstanceType': 'string',
        'LastServiceErrorId': 'string',
        'LayerIds': [
            'string',
        ],
        'Os': 'string',
        'Platform': 'string',
        'PrivateDns': 'string',
        'PrivateIp': 'string',
        'PublicDns': 'string',
        'PublicIp': 'string',
        'RegisteredBy': 'string',
        'ReportedAgentVersion': 'string',
        'ReportedOs': {
            'Family': 'string',
            'Name': 'string',
            'Version': 'string'
        },
        'RootDeviceType': 'ebs'|'instance-store',
        'RootDeviceVolumeId': 'string',
        'SecurityGroupIds': [
            'string',
        ],
        'SshHostDsaKeyFingerprint': 'string',
        'SshHostRsaKeyFingerprint': 'string',
        'SshKeyName': 'string',
        'StackId': 'string',
        'Status': 'string',
        'SubnetId': 'string',
        'VirtualizationType': 'paravirtual'|'hvm'
    },
]
}
```

Response Structure

- (*dict*) –

Contains the response to a `DescribeInstances` request.

- **Instances** (*list*) –

An array of Instance objects that describe the instances.

- * (*dict*) –

Describes an instance.

- **AgentVersion** (*string*) –

The agent version. This parameter is set to `INHERIT` if the instance inherits the default stack setting or to a version number for a fixed agent version.

- **AmiId** (*string*) –

A custom AMI ID to be used to create the instance. For more information, see [Instances](#)

- **Architecture** (*string*) –

The instance architecture: “i386” or “x86_64”.

- **AutoScalingType** (*string*) –

For load-based or time-based instances, the type.

- **AvailabilityZone** (*string*) –

The instance Availability Zone. For more information, see [Regions and Endpoints](#).

- **BlockDeviceMappings** (*list*) –

An array of `BlockDeviceMapping` objects that specify the instance’s block device mappings.

- (*dict*) –

Describes a block device mapping. This data type maps directly to the Amazon EC2 [BlockDeviceMapping](#) data type.

- **DeviceName** (*string*) –

The device name that is exposed to the instance, such as `/dev/sdh`. For the root device, you can use the explicit device name or you can set this parameter to `ROOT_DEVICE` and AWS OpsWorks will provide the correct device name.

- **NoDevice** (*string*) –

Suppresses the specified device included in the AMI’s block device mapping.

- **VirtualName** (*string*) –

The virtual device name. For more information, see [BlockDeviceMapping](#).

- **Ebs** (*dict*) –

An `EBSBlockDevice` that defines how to configure an Amazon EBS volume when the instance is launched.

- **SnapshotId** (*string*) –

The snapshot ID.

- **Iops** (*integer*) –

The number of I/O operations per second (IOPS) that the volume supports. For more information, see [EbsBlockDevice](#).

- **VolumeSize** (*integer*) –

The volume size, in GiB. For more information, see [EbsBlockDevice](#).

- **VolumeType** (*string*) –

The volume type. `gp2` for General Purpose (SSD) volumes, `io1` for Provisioned IOPS (SSD) volumes, and `standard` for Magnetic volumes.

- **DeleteOnTermination** (*boolean*) –

Whether the volume is deleted on instance termination.

- **CreatedAt** (*string*) –

The time that the instance was created.

- **EbsOptimized** (*boolean*) –

Whether this is an Amazon EBS-optimized instance.

- **Ec2InstanceId** (*string*) –

The ID of the associated Amazon EC2 instance.

- **EcsClusterArn** (*string*) –
For container instances, the Amazon ECS cluster’s ARN.
- **EcsContainerInstanceArn** (*string*) –
For container instances, the instance’s ARN.
- **ElasticIp** (*string*) –
The instance [Elastic IP address](#) .
- **Hostname** (*string*) –
The instance host name.
- **InfrastructureClass** (*string*) –
For registered instances, the infrastructure class: `ec2` or `on-premises` .
- **InstallUpdatesOnBoot** (*boolean*) –
Whether to install operating system and package updates when the instance boots. The default value is `true` . If this value is set to `false` , you must then update your instances manually by using `CreateDeployment` to run the `update_dependencies` stack command or by manually running `yum` (Amazon Linux) or `apt-get` (Ubuntu) on the instances.

Note: We strongly recommend using the default value of `true` , to ensure that your instances have the latest security updates.

- **InstanceId** (*string*) –
The instance ID.
- **InstanceProfileArn** (*string*) –
The ARN of the instance’s IAM profile. For more information about IAM ARNs, see [Using Identifiers](#) .
- **InstanceType** (*string*) –
The instance type, such as `t2.micro` .
- **LastServiceErrorId** (*string*) –
The ID of the last service error. For more information, call `DescribeServiceErrors` .
- **LayerIds** (*list*) –
An array containing the instance layer IDs.
- (*string*) –
- **Os** (*string*) –
The instance’s operating system.
- **Platform** (*string*) –
The instance’s platform.
- **PrivateDns** (*string*) –
The The instance’s private DNS name.
- **PrivateIp** (*string*) –
The instance’s private IP address.
- **PublicDns** (*string*) –

- The instance public DNS name.
- **PublicIp** (*string*) –
- The instance public IP address.
- **RegisteredBy** (*string*) –
- For registered instances, who performed the registration.
- **ReportedAgentVersion** (*string*) –
- The instance’s reported AWS OpsWorks agent version.
- **ReportedOs** (*dict*) –
- For registered instances, the reported operating system.
- **Family** (*string*) –
- The operating system family.
- **Name** (*string*) –
- The operating system name.
- **Version** (*string*) –
- The operating system version.
- **RootDeviceType** (*string*) –
- The instance’s root device type. For more information, see [Storage for the Root Device](#).
- **RootDeviceVolumeId** (*string*) –
- The root device volume ID.
- **SecurityGroupIds** (*list*) –
- An array containing the instance security group IDs.
- (*string*) –
- **SshHostDsaKeyFingerprint** (*string*) –
- The SSH key’s Deep Security Agent (DSA) fingerprint.
- **SshHostRsaKeyFingerprint** (*string*) –
- The SSH key’s RSA fingerprint.
- **SshKeyName** (*string*) –
- The instance’s Amazon EC2 key-pair name.
- **StackId** (*string*) –
- The stack ID.
- **Status** (*string*) –
- The instance status:
- booting
- connection_lost
- online
- pending
- rebooting
- requested
- running_setup
- setup_failed
- shutting_down
- start_failed
- stopped
- stopping
- terminated

- `terminating`
- **SubnetId** (*string*) –
The instance's subnet ID; applicable only if the stack is running in a VPC.
- **VirtualizationType** (*string*) –
The instance's virtualization type: `paravirtual` or `hvm`.

describe_layers (***kwargs*)

Requests a description of one or more layers in a specified stack.

Note: You must specify at least one of the parameters.

Required Permissions : To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.describe_layers(  
    StackId='string',  
    LayerIds=[  
        'string',  
    ]  
)
```

Parameters

- **StackId** (*string*) – The stack ID.
- **LayerIds** (*list*) – An array of layer IDs that specify the layers to be described. If you omit this parameter, `DescribeLayers` returns a description of every layer in the specified stack.
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{  
    'Layers': [  
        {  
            'StackId': 'string',  
            'LayerId': 'string',  
            'Type': 'aws-flow-ruby'|'ecs-cluster'|'java-app'|'lb'|'web'|'php-app'|  
            'Name': 'string',  
            'Shortname': 'string',  
            'Attributes': {  
                'string': 'string'  
            },  
            'CustomInstanceProfileArn': 'string',  
            'CustomJson': 'string',  
            'CustomSecurityGroupIds': [  
                'string',  
            ],  
            'DefaultSecurityGroupNames': [  
                'string',  
            ],  
            'Packages': [  

```

```

        'string',
    ],
    'VolumeConfigurations': [
        {
            'MountPoint': 'string',
            'RaidLevel': 123,
            'NumberOfDisks': 123,
            'Size': 123,
            'VolumeType': 'string',
            'Iops': 123
        },
    ],
    'EnableAutoHealing': True|False,
    'AutoAssignElasticIps': True|False,
    'AutoAssignPublicIps': True|False,
    'DefaultRecipes': {
        'Setup': [
            'string',
        ],
        'Configure': [
            'string',
        ],
        'Deploy': [
            'string',
        ],
        'Undeploy': [
            'string',
        ],
        'Shutdown': [
            'string',
        ]
    },
    'CustomRecipes': {
        'Setup': [
            'string',
        ],
        'Configure': [
            'string',
        ],
        'Deploy': [
            'string',
        ],
        'Undeploy': [
            'string',
        ],
        'Shutdown': [
            'string',
        ]
    },
    'CreatedAt': 'string',
    'InstallUpdatesOnBoot': True|False,
    'UseEbsOptimizedInstances': True|False,
    'LifecycleEventConfiguration': {
        'Shutdown': {
            'ExecutionTimeout': 123,
            'DelayUntilElbConnectionsDrained': True|False
        }
    }
}

```

```
    },  
  ]  
}
```

Response Structure

- (*dict*) –

Contains the response to a `DescribeLayers` request.

- **Layers** (*list*) –

An array of `Layer` objects that describe the layers.

- * (*dict*) –

Describes a layer.

- **StackId** (*string*) –

The layer stack ID.

- **LayerId** (*string*) –

The layer ID.

- **Type** (*string*) –

The layer type.

- **Name** (*string*) –

The layer name.

- **Shortname** (*string*) –

The layer short name.

- **Attributes** (*dict*) –

The layer attributes.

For the `HaproxyStatsPassword`, `MysqlRootPassword`, and `GangliaPassword` attributes, AWS OpsWorks returns `*****FILTERED*****` instead of the actual value

For an `ECS Cluster` layer, AWS OpsWorks the `EcsClusterArn` attribute is set to the cluster's ARN.

- (*string*) –

- (*string*) –

- **CustomInstanceProfileArn** (*string*) –

The ARN of the default IAM profile to be used for the layer's EC2 instances. For more information about IAM ARNs, see [Using Identifiers](#).

- **CustomJson** (*string*) –

A JSON formatted string containing the layer's custom stack configuration and deployment attributes.

- **CustomSecurityGroupIds** (*list*) –

An array containing the layer's custom security group IDs.

- (*string*) –

- **DefaultSecurityGroupNames** (*list*) –

An array containing the layer's security group names.

- (*string*) –

- **Packages** (*list*) –

An array of `Package` objects that describe the layer's packages.

- *(string)* –
- **VolumeConfigurations** (*list*) –

A `VolumeConfigurations` object that describes the layer's Amazon EBS volumes.

- *(dict)* –

Describes an Amazon EBS volume configuration.

- **MountPoint** (*string*) –

The volume mount point. For example “/dev/sdh”.

- **RaidLevel** (*integer*) –

The volume RAID level .

- **NumberOfDisks** (*integer*) –

The number of disks in the volume.

- **Size** (*integer*) –

The volume size.

- **VolumeType** (*string*) –

The volume type:

- `standard - Magnetic`
- `io1 - Provisioned IOPS (SSD)`
- `gp2 - General Purpose (SSD)`
- **Iops** (*integer*) –

For PIOPS volumes, the IOPS per disk.

- **EnableAutoHealing** (*boolean*) –

Whether auto healing is disabled for the layer.

- **AutoAssignElasticIps** (*boolean*) –

Whether to automatically assign an [Elastic IP address](#) to the layer's instances. For more information, see [How to Edit a Layer](#) .

- **AutoAssignPublicIps** (*boolean*) –

For stacks that are running in a VPC, whether to automatically assign a public IP address to the layer's instances. For more information, see [How to Edit a Layer](#) .

- **DefaultRecipes** (*dict*) –

AWS OpsWorks supports five lifecycle events: **setup** , **configuration** , **deploy** , **undeploy** , and **shutdown** . For each layer, AWS OpsWorks runs a set of standard recipes for each event. In addition, you can provide custom recipes for any or all layers and events. AWS OpsWorks runs custom event recipes after the standard recipes. `LayerCustomRecipes` specifies the custom recipes for a particular layer to be run in response to each of the five events.

To specify a recipe, use the cookbook's directory name in the repository followed by two colons and the recipe name, which is the recipe's file name without the `.rb` extension. For example: `phpapp2::dbsetup` specifies the `dbsetup.rb` recipe in the repository's `phpapp2` folder.

- **Setup** (*list*) –
An array of custom recipe names to be run following a setup event.
- (*string*) –
- **Configure** (*list*) –
An array of custom recipe names to be run following a configure event.
- (*string*) –
- **Deploy** (*list*) –
An array of custom recipe names to be run following a deploy event.
- (*string*) –
- **Undeploy** (*list*) –
An array of custom recipe names to be run following a undeploy event.
- (*string*) –
- **Shutdown** (*list*) –
An array of custom recipe names to be run following a shutdown event.
- (*string*) –
- **CustomRecipes** (*dict*) –
A `LayerCustomRecipes` object that specifies the layer's custom recipes.
- **Setup** (*list*) –
An array of custom recipe names to be run following a setup event.
- (*string*) –
- **Configure** (*list*) –
An array of custom recipe names to be run following a configure event.
- (*string*) –
- **Deploy** (*list*) –
An array of custom recipe names to be run following a deploy event.
- (*string*) –
- **Undeploy** (*list*) –
An array of custom recipe names to be run following a undeploy event.
- (*string*) –
- **Shutdown** (*list*) –
An array of custom recipe names to be run following a shutdown event.
- (*string*) –
- **CreatedAt** (*string*) –
Date when the layer was created.
- **InstallUpdatesOnBoot** (*boolean*) –
Whether to install operating system and package updates

when the instance boots. The default value is `true`. If this value is set to `false`, you must then update your instances manually by using `CreateDeployment` to run the `update_dependencies` stack command or manually running `yum` (Amazon Linux) or `apt-get` (Ubuntu) on the instances.

Note: We strongly recommend using the default value of `true`, to ensure that your instances have the latest security updates.

- **UseEbsOptimizedInstances** (*boolean*) –

Whether the layer uses Amazon EBS-optimized instances.

- **LifecycleEventConfiguration** (*dict*) –

A `LifeCycleEventConfiguration` object that specifies the Shutdown event configuration.

- **Shutdown** (*dict*) –

A `ShutdownEventConfiguration` object that specifies the Shutdown event configuration.

- **ExecutionTimeout** (*integer*) –

The time, in seconds, that AWS OpsWorks will wait after triggering a Shutdown event before shutting down an instance.

- **DelayUntilElbConnectionsDrained** (*boolean*) –

Whether to enable Elastic Load Balancing connection draining. For more information, see [Connection Draining](#)

describe_load_based_auto_scaling (***kwargs*)

Describes load-based auto scaling configurations for specified layers.

Note: You must specify at least one of the parameters.

Required Permissions : To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.describe_load_based_auto_scaling(
    LayerIds=[
        'string',
    ]
)
```

Parameters `LayerIds` (*list*) – [REQUIRED]

An array of layer IDs.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'LoadBasedAutoScalingConfigurations': [
```

```
{
  'LayerId': 'string',
  'Enable': True|False,
  'UpScaling': {
    'InstanceCount': 123,
    'ThresholdsWaitTime': 123,
    'IgnoreMetricsTime': 123,
    'CpuThreshold': 123.0,
    'MemoryThreshold': 123.0,
    'LoadThreshold': 123.0,
    'Alarms': [
      'string',
    ]
  },
  'DownScaling': {
    'InstanceCount': 123,
    'ThresholdsWaitTime': 123,
    'IgnoreMetricsTime': 123,
    'CpuThreshold': 123.0,
    'MemoryThreshold': 123.0,
    'LoadThreshold': 123.0,
    'Alarms': [
      'string',
    ]
  }
},
]
```

Response Structure

- (*dict*) –

Contains the response to a `DescribeLoadBasedAutoScaling` request.

- **LoadBasedAutoScalingConfigurations** (*list*) –

An array of `LoadBasedAutoScalingConfiguration` objects that describe each layer's configuration.

- * (*dict*) –

Describes a layer's load-based auto scaling configuration.

- **LayerId** (*string*) –

The layer ID.

- **Enable** (*boolean*) –

Whether load-based auto scaling is enabled for the layer.

- **UpScaling** (*dict*) –

An `AutoScalingThresholds` object that describes the upscaling configuration, which defines how and when AWS OpsWorks increases the number of instances.

- **InstanceCount** (*integer*) –

The number of instances to add or remove when the load exceeds a threshold.

- **ThresholdsWaitTime** (*integer*) –

The amount of time, in minutes, that the load must exceed a threshold before more instances are added or removed.

- **IgnoreMetricsTime** (*integer*) –

The amount of time (in minutes) after a scaling event occurs that AWS OpsWorks should ignore metrics and suppress additional scaling events. For example, AWS OpsWorks adds new instances following an upscaling event but the instances won't start reducing the load until they have been booted and configured. There is no point in raising additional scaling events during that operation, which typically takes several minutes. `IgnoreMetricsTime` allows you to direct AWS OpsWorks to suppress scaling events long enough to get the new instances online.

- **CpuThreshold** (*float*) –

The CPU utilization threshold, as a percent of the available CPU.

- **MemoryThreshold** (*float*) –

The memory utilization threshold, as a percent of the available memory.

- **LoadThreshold** (*float*) –

The load threshold. For more information about how load is computed, see [Load \(computing\)](#).

- **Alarms** (*list*) –

Custom Cloudwatch auto scaling alarms, to be used as thresholds. This parameter takes a list of up to five alarm names, which are case sensitive and must be in the same region as the stack.

Note: To use custom alarms, you must update your service role to allow `cloudwatch:DescribeAlarms`. You can either have AWS OpsWorks update the role for you when you first use this feature or you can edit the role manually. For more information, see [Allowing AWS OpsWorks to Act on Your Behalf](#).

- (*string*) –

- **DownScaling** (*dict*) –

An `AutoScalingThresholds` object that describes the downscaling configuration, which defines how and when AWS OpsWorks reduces the number of instances.

- **InstanceCount** (*integer*) –

The number of instances to add or remove when the load exceeds a threshold.

- **ThresholdsWaitTime** (*integer*) –

The amount of time, in minutes, that the load must exceed a threshold before more instances are added or removed.

- **IgnoreMetricsTime** (*integer*) –

The amount of time (in minutes) after a scaling event occurs that AWS OpsWorks should ignore metrics and suppress additional scaling events. For example, AWS OpsWorks adds new instances following an upscaling event but the instances won't start reducing the load until they have been booted and configured. There is no point in raising additional scaling

events during that operation, which typically takes several minutes. `IgnoreMetricsTime` allows you to direct AWS OpsWorks to suppress scaling events long enough to get the new instances online.

- **CpuThreshold** (*float*) –

The CPU utilization threshold, as a percent of the available CPU.

- **MemoryThreshold** (*float*) –

The memory utilization threshold, as a percent of the available memory.

- **LoadThreshold** (*float*) –

The load threshold. For more information about how load is computed, see [Load \(computing\)](#).

- **Alarms** (*list*) –

Custom Cloudwatch auto scaling alarms, to be used as thresholds. This parameter takes a list of up to five alarm names, which are case sensitive and must be in the same region as the stack.

Note: To use custom alarms, you must update your service role to allow `cloudwatch:DescribeAlarms`. You can either have AWS OpsWorks update the role for you when you first use this feature or you can edit the role manually. For more information, see [Allowing AWS OpsWorks to Act on Your Behalf](#).

- (*string*) –

describe_my_user_profile()

Describes a user's SSH information.

Required Permissions : To use this action, an IAM user must have self-management enabled or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.describe_my_user_profile()
```

Return type dict

Returns

Response Syntax

```
{
  'UserProfile': {
    'IamUserArn': 'string',
    'Name': 'string',
    'SshUsername': 'string',
    'SshPublicKey': 'string'
  }
}
```

Response Structure

- (*dict*) –

Contains the response to a `DescribeMyUserProfile` request.

- **UserProfile** (*dict*) –

A `UserProfile` object that describes the user's SSH information.

- * **IamUserArn** (*string*) –

The user's IAM ARN.

- * **Name** (*string*) –

The user's name.

- * **SshUsername** (*string*) –

The user's SSH user name.

- * **SshPublicKey** (*string*) –

The user's SSH public key.

describe_permissions (***kwargs*)

Describes the permissions for a specified stack.

Required Permissions : To use this action, an IAM user must have a `Manage` permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.describe_permissions(
    IamUserArn='string',
    StackId='string'
)
```

Parameters

- **IamUserArn** (*string*) – The user's IAM ARN. For more information about IAM ARNs, see [Using Identifiers](#).
- **StackId** (*string*) – The stack ID.

Return type `dict`

Returns

Response Syntax

```
{
  'Permissions': [
    {
      'StackId': 'string',
      'IamUserArn': 'string',
      'AllowSsh': True|False,
      'AllowSudo': True|False,
      'Level': 'string'
    },
  ],
}
```

Response Structure

- (*dict*) –

Contains the response to a `DescribePermissions` request.

- **Permissions** (*list*) –

An array of `Permission` objects that describe the stack permissions.

- * If the request object contains only a stack ID, the array contains a `Permission` object with permissions for each of the stack IAM ARNs.
- * If the request object contains only an IAM ARN, the array contains a `Permission` object with permissions for each of the user's stack IDs.
- * If the request contains a stack ID and an IAM ARN, the array contains a single `Permission` object with permissions for the specified stack and IAM ARN.
- * *(dict)* –

Describes stack or user permissions.

- **StackId** (*string*) –

A stack ID.

- **IamUserArn** (*string*) –

The Amazon Resource Name (ARN) for an AWS Identity and Access Management (IAM) role. For more information about IAM ARNs, see [Using Identifiers](#) .

- **AllowSsh** (*boolean*) –

Whether the user can use SSH.

- **AllowSudo** (*boolean*) –

Whether the user can use **sudo** .

- **Level** (*string*) –

The user's permission level, which must be the following:

- deny
- show
- deploy
- manage
- iam_only

For more information on the permissions associated with these levels, see [Managing User Permissions](#)

describe_raid_arrays (***kwargs*)
Describe an instance's RAID arrays.

Note: You must specify at least one of the parameters.

Required Permissions : To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.describe_raid_arrays(
    InstanceId='string',
    StackId='string',
    RaidArrayIds=[
        'string',
    ]
)
```

Parameters

- **InstanceId** (*string*) – The instance ID. If you use this parameter, `DescribeRaidArrays` returns descriptions of the RAID arrays associated with the specified instance.
- **StackId** (*string*) – The stack ID.
- **RaidArrayIds** (*list*) – An array of RAID array IDs. If you use this parameter, `DescribeRaidArrays` returns descriptions of the specified arrays. Otherwise, it returns a description of every array.

– (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'RaidArrays': [
    {
      'RaidArrayId': 'string',
      'InstanceId': 'string',
      'Name': 'string',
      'RaidLevel': 123,
      'NumberOfDisks': 123,
      'Size': 123,
      'Device': 'string',
      'MountPoint': 'string',
      'AvailabilityZone': 'string',
      'CreatedAt': 'string',
      'StackId': 'string',
      'VolumeType': 'string',
      'Iops': 123
    },
  ],
}
```

Response Structure

- (*dict*) –

Contains the response to a `DescribeRaidArrays` request.

- **RaidArrays** (*list*) –

A `RaidArrays` object that describes the specified RAID arrays.

- * (*dict*) –

Describes an instance's RAID array.

- **RaidArrayId** (*string*) –

The array ID.

- **InstanceId** (*string*) –

The instance ID.

- **Name** (*string*) –

The array name.

- **RaidLevel** (*integer*) –

The RAID level.

- **NumberOfDisks** (*integer*) –

The number of disks in the array.

- **Size** (*integer*) –

The array's size.

- **Device** (*string*) –
The array's Linux device. For example /dev/mdadm0.
- **MountPoint** (*string*) –
The array's mount point.
- **AvailabilityZone** (*string*) –
The array's Availability Zone. For more information, see [Regions and Endpoints](#).
- **CreatedAt** (*string*) –
When the RAID array was created.
- **StackId** (*string*) –
The stack ID.
- **VolumeType** (*string*) –
The volume type, standard or PIOPS.
- **Iops** (*integer*) –
For PIOPS volumes, the IOPS per disk.

describe_rds_db_instances (***kwargs*)

Describes Amazon RDS instances.

Required Permissions : To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.describe_rds_db_instances(  
    StackId='string',  
    RdsDbInstanceArns=[  
        'string',  
    ]  
)
```

Parameters

- **StackId** (*string*) – **[REQUIRED]**

The stack ID that the instances are registered with. The operation returns descriptions of all registered Amazon RDS instances.

- **RdsDbInstanceArns** (*list*) – An array containing the ARNs of the instances to be described.
– (*string*) –

Return type dict

Returns

Response Syntax

```
{  
    'RdsDbInstances': [  
        {  
            'RdsDbInstanceArn': 'string',  
            'DbInstanceIdentifier': 'string',  
            'DbUser': 'string',  
            'DbPassword': 'string',  
            'Region': 'string',  
            'Address': 'string',
```

```

        'Engine': 'string',
        'StackId': 'string',
        'MissingOnRds': True|False
    },
]
}

```

Response Structure

- *(dict)* –

Contains the response to a `DescribeRdsDbInstances` request.

- **RdsDbInstances** (*list*) –

An array of `RdsDbInstance` objects that describe the instances.

- * *(dict)* –

Describes an Amazon RDS instance.

- **RdsDbInstanceArn** (*string*) –

The instance's ARN.

- **DbInstanceIdentifier** (*string*) –

The DB instance identifier.

- **DbUser** (*string*) –

The master user name.

- **DbPassword** (*string*) –

AWS OpsWorks returns `*****FILTERED*****` instead of the actual value.

- **Region** (*string*) –

The instance's AWS region.

- **Address** (*string*) –

The instance's address.

- **Engine** (*string*) –

The instance's database engine.

- **StackId** (*string*) –

The ID of the stack that the instance is registered with.

- **MissingOnRds** (*boolean*) –

Set to `true` if AWS OpsWorks was unable to discover the Amazon RDS instance. AWS OpsWorks attempts to discover the instance only once. If this value is set to `true`, you must deregister the instance and then register it again.

describe_service_errors (***kwargs*)

Describes AWS OpsWorks service errors.

Required Permissions : To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```

response = client.describe_service_errors(
    StackId='string',
    InstanceId='string',
    ServiceErrorIds=[

```

```
        'string',  
    ]  
)
```

Parameters

- **StackId** (*string*) – The stack ID. If you use this parameter, `DescribeServiceErrors` returns descriptions of the errors associated with the specified stack.
- **InstanceId** (*string*) – The instance ID. If you use this parameter, `DescribeServiceErrors` returns descriptions of the errors associated with the specified instance.
- **ServiceErrorIds** (*list*) – An array of service error IDs. If you use this parameter, `DescribeServiceErrors` returns descriptions of the specified errors. Otherwise, it returns a description of every error.
 - (*string*) –

Return type dict**Returns****Response Syntax**

```
{  
    'ServiceErrors': [  
        {  
            'ServiceErrorId': 'string',  
            'StackId': 'string',  
            'InstanceId': 'string',  
            'Type': 'string',  
            'Message': 'string',  
            'CreatedAt': 'string'  
        },  
    ]  
}
```

Response Structure

- (*dict*) –

Contains the response to a `DescribeServiceErrors` request.

- **ServiceErrors** (*list*) –

An array of `ServiceError` objects that describe the specified service errors.

- * (*dict*) –

Describes an AWS OpsWorks service error.

- **ServiceErrorId** (*string*) –

The error ID.

- **StackId** (*string*) –

The stack ID.

- **InstanceId** (*string*) –

The instance ID.

- **Type** (*string*) –

The error type.

- **Message** (*string*) –

A message that describes the error.

- **CreatedAt** (*string*) –
When the error occurred.

describe_stack_provisioning_parameters (***kwargs*)

Requests a description of a stack's provisioning parameters.

Required Permissions : To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.describe_stack_provisioning_parameters(
    StackId='string'
)
```

Parameters **StackId** (*string*) – [REQUIRED]

The stack ID

Return type dict

Returns

Response Syntax

```
{
    'AgentInstallerUrl': 'string',
    'Parameters': {
        'string': 'string'
    }
}
```

Response Structure

- (*dict*) –

Contains the response to a DescribeStackProvisioningParameters request.

- **AgentInstallerUrl** (*string*) –

The AWS OpsWorks agent installer's URL.

- **Parameters** (*dict*) –

An embedded object that contains the provisioning parameters.

- * (*string*) –
- (*string*) –

describe_stack_summary (***kwargs*)

Describes the number of layers and apps in a specified stack, and the number of instances in each state, such as `running_setup` or `online` .

Required Permissions : To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.describe_stack_summary(
    StackId='string'
)
```

Parameters `StackId` (*string*) – [REQUIRED]

The stack ID.

Return type dict**Returns****Response Syntax**

```
{
  'StackSummary': {
    'StackId': 'string',
    'Name': 'string',
    'Arn': 'string',
    'LayersCount': 123,
    'AppsCount': 123,
    'InstancesCount': {
      'Assigning': 123,
      'Booting': 123,
      'ConnectionLost': 123,
      'Deregistering': 123,
      'Online': 123,
      'Pending': 123,
      'Rebooting': 123,
      'Registered': 123,
      'Registering': 123,
      'Requested': 123,
      'RunningSetup': 123,
      'SetupFailed': 123,
      'ShuttingDown': 123,
      'StartFailed': 123,
      'Stopped': 123,
      'Stopping': 123,
      'Terminated': 123,
      'Terminating': 123,
      'Unassigning': 123
    }
  }
}
```

Response Structure

- (*dict*) –

Contains the response to a `DescribeStackSummary` request.

- **StackSummary** (*dict*) –

A `StackSummary` object that contains the results.

- * **StackId** (*string*) –

The stack ID.

- * **Name** (*string*) –

The stack name.

- * **Arn** (*string*) –

The stack's ARN.

- * **LayersCount** (*integer*) –

The number of layers.

- * **AppsCount** (*integer*) –

The number of apps.

* **InstancesCount** (*dict*) –

An `InstancesCount` object with the number of instances in each status.

- **Assigning** (*integer*) –
The number of instances in the Assigning state.
- **Booting** (*integer*) –
The number of instances with `booting` status.
- **ConnectionLost** (*integer*) –
The number of instances with `connection_lost` status.
- **Deregistering** (*integer*) –
The number of instances in the Deregistering state.
- **Online** (*integer*) –
The number of instances with `online` status.
- **Pending** (*integer*) –
The number of instances with `pending` status.
- **Rebooting** (*integer*) –
The number of instances with `rebooting` status.
- **Registered** (*integer*) –
The number of instances in the Registered state.
- **Registering** (*integer*) –
The number of instances in the Registering state.
- **Requested** (*integer*) –
The number of instances with `requested` status.
- **RunningSetup** (*integer*) –
The number of instances with `running_setup` status.
- **SetupFailed** (*integer*) –
The number of instances with `setup_failed` status.
- **ShuttingDown** (*integer*) –
The number of instances with `shutting_down` status.
- **StartFailed** (*integer*) –
The number of instances with `start_failed` status.
- **Stopped** (*integer*) –
The number of instances with `stopped` status.
- **Stopping** (*integer*) –
The number of instances with `stopping` status.
- **Terminated** (*integer*) –
The number of instances with `terminated` status.
- **Terminating** (*integer*) –
The number of instances with `terminating` status.
- **Unassigning** (*integer*) –
The number of instances in the Unassigning state.

describe_stacks (***kwargs*)

Requests a description of one or more stacks.

Required Permissions : To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.describe_stacks(  
    StackIds=[  
        'string',  
    ]  
)
```

Parameters **StackIds** (*list*) – An array of stack IDs that specify the stacks to be described.

If you omit this parameter, DescribeStacks returns a description of every stack.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{  
    'Stacks': [  
        {  
            'StackId': 'string',  
            'Name': 'string',  
            'Arn': 'string',  
            'Region': 'string',  
            'VpcId': 'string',  
            'Attributes': {  
                'string': 'string'  
            },  
            'ServiceRoleArn': 'string',  
            'DefaultInstanceProfileArn': 'string',  
            'DefaultOs': 'string',  
            'HostnameTheme': 'string',  
            'DefaultAvailabilityZone': 'string',  
            'DefaultSubnetId': 'string',  
            'CustomJson': 'string',  
            'ConfigurationManager': {  
                'Name': 'string',  
                'Version': 'string'  
            },  
            'ChefConfiguration': {  
                'ManageBerkshelf': True|False,  
                'BerkshelfVersion': 'string'  
            },  
            'UseCustomCookbooks': True|False,  
            'UseOpsworksSecurityGroups': True|False,  
            'CustomCookbooksSource': {  
                'Type': 'git'|'svn'|'archive'|'s3',  
                'Url': 'string',  
                'Username': 'string',  
                'Password': 'string',  
                'SshKey': 'string',  
                'Revision': 'string'  
            },  
            'DefaultSshKeyName': 'string',  
            'CreatedAt': 'string',
```



```

        'DefaultRootDeviceType': 'ebs'|'instance-store',
        'AgentVersion': 'string'
    },
]
}

```

Response Structure

- *(dict)* –

Contains the response to a `DescribeStacks` request.

- **Stacks** (*list*) –

An array of `Stack` objects that describe the stacks.

- * *(dict)* –

Describes a stack.

- **StackId** (*string*) –

The stack ID.

- **Name** (*string*) –

The stack name.

- **Arn** (*string*) –

The stack's ARN.

- **Region** (*string*) –

The stack AWS region, such as “us-east-1”. For more information about AWS regions, see [Regions and Endpoints](#) .

- **VpcId** (*string*) –

The VPC ID; applicable only if the stack is running in a VPC.

- **Attributes** (*dict*) –

The stack's attributes.

- (*string*) –

- (*string*) –

- **ServiceRoleArn** (*string*) –

The stack AWS Identity and Access Management (IAM) role.

- **DefaultInstanceProfileArn** (*string*) –

The ARN of an IAM profile that is the default profile for all of the stack's EC2 instances. For more information about IAM ARNs, see [Using Identifiers](#) .

- **DefaultOs** (*string*) –

The stack's default operating system.

- **HostnameTheme** (*string*) –

The stack host name theme, with spaces replaced by underscores.

- **DefaultAvailabilityZone** (*string*) –

The stack's default Availability Zone. For more information, see [Regions and Endpoints](#) .

- **DefaultSubnetId** (*string*) –

The default subnet ID; applicable only if the stack is running in a VPC.

- **CustomJson** (*string*) –

A JSON object that contains user-defined attributes to be added to the stack configuration and deployment attributes. You can use custom JSON to override the corresponding default stack configuration attribute values or to pass data to recipes. The string should be in the following format and must escape characters such as “”:

```
"{\\"key1\\": \\"value1\\",
 \\"key2\\": \\"value2\\", ...}"
```

For more information on custom JSON, see [Use Custom JSON to Modify the Stack Configuration Attributes](#) .

- **ConfigurationManager** (*dict*) –

The configuration manager.

- **Name** (*string*) –

The name. This parameter must be set to “Chef”.

- **Version** (*string*) –

The Chef version. This parameter must be set to 0.9, 11.4, or 11.10. The default value is 11.4.

- **ChefConfiguration** (*dict*) –

A `ChefConfiguration` object that specifies whether to enable Berkshelf and the Berkshelf version. For more information, see [Create a New Stack](#) .

- **ManageBerkshelf** (*boolean*) –

Whether to enable Berkshelf.

- **BerkshelfVersion** (*string*) –

The Berkshelf version.

- **UseCustomCookbooks** (*boolean*) –

Whether the stack uses custom cookbooks.

- **UseOpsworksSecurityGroups** (*boolean*) –

Whether the stack automatically associates the AWS OpsWorks built-in security groups with the stack’s layers.

- **CustomCookbooksSource** (*dict*) –

Contains the information required to retrieve an app or cookbook from a repository. For more information, see [Creating Apps](#) or [Custom Recipes and Cookbooks](#) .

- **Type** (*string*) –

The repository type.

- **Url** (*string*) –

The source URL.

- **Username** (*string*) –

This parameter depends on the repository type.

- For Amazon S3 bundles, set `Username` to the appropriate IAM access key ID.

- For HTTP bundles, Git repositories, and Subversion repositories, set `Username` to the user name.

- **Password** (*string*) –

When included in a request, the parameter depends on the repository type.

- For Amazon S3 bundles, set `Password` to the appropriate IAM secret access key.
- For HTTP bundles and Subversion repositories, set `Password` to the password.

For more information on how to safely handle IAM credentials, see <http://docs.aws.amazon.com/general/latest/gr/aws-access-keys-best-practices.html>.

In responses, AWS OpsWorks returns *****FILTERED***** instead of the actual value.

- **SshKey** (*string*) –

In requests, the repository's SSH key.

In responses, AWS OpsWorks returns *****FILTERED***** instead of the actual value.

- **Revision** (*string*) –

The application's version. AWS OpsWorks enables you to easily deploy new versions of an application. One of the simplest approaches is to have branches or revisions in your repository that represent different versions that can potentially be deployed.

- **DefaultSshKeyName** (*string*) –

A default Amazon EC2 key pair for the stack's instances. You can override this value when you create or update an instance.

- **CreatedAt** (*string*) –

The date when the stack was created.

- **DefaultRootDeviceType** (*string*) –

The default root device type. This value is used by default for all instances in the stack, but you can override it when you create an instance. For more information, see [Storage for the Root Device](#).

- **AgentVersion** (*string*) –

The agent version. This parameter is set to `LATEST` for auto-update, or a version number for a fixed agent version.

describe_time_based_auto_scaling (***kwargs*)

Describes time-based auto scaling configurations for specified instances.

Note: You must specify at least one of the parameters.

Required Permissions : To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.describe_time_based_auto_scaling(
    InstanceIds=[
        'string',
    ]
)
```

Parameters `InstanceIds` (*list*) – **[REQUIRED]**

An array of instance IDs.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'TimeBasedAutoScalingConfigurations': [
    {
      'InstanceId': 'string',
      'AutoScalingSchedule': {
        'Monday': {
          'string': 'string'
        },
        'Tuesday': {
          'string': 'string'
        },
        'Wednesday': {
          'string': 'string'
        },
        'Thursday': {
          'string': 'string'
        },
        'Friday': {
          'string': 'string'
        },
        'Saturday': {
          'string': 'string'
        },
        'Sunday': {
          'string': 'string'
        }
      }
    },
  ]
}
```

Response Structure

- (*dict*) –

Contains the response to a `DescribeTimeBasedAutoScaling` request.

- **TimeBasedAutoScalingConfigurations** (*list*) –

An array of `TimeBasedAutoScalingConfiguration` objects that describe the configuration for the specified instances.

- * (*dict*) –

Describes an instance's time-based auto scaling configuration.

- **InstanceId** (*string*) –

The instance ID.

- **AutoScalingSchedule** (*dict*) –

A `WeeklyAutoScalingSchedule` object with the instance schedule.

- **Monday** (*dict*) –

The schedule for Monday.

- *(string)* –
- *(string)* –
- **Tuesday** (*dict*) –

The schedule for Tuesday.

- *(string)* –
- *(string)* –
- **Wednesday** (*dict*) –

The schedule for Wednesday.

- *(string)* –
- *(string)* –
- **Thursday** (*dict*) –

The schedule for Thursday.

- *(string)* –
- *(string)* –
- **Friday** (*dict*) –

The schedule for Friday.

- *(string)* –
- *(string)* –
- **Saturday** (*dict*) –

The schedule for Saturday.

- *(string)* –
- *(string)* –
- **Sunday** (*dict*) –

The schedule for Sunday.

- *(string)* –
- *(string)* –

describe_user_profiles (***kwargs*)

Describe specified users.

Required Permissions : To use this action, an IAM user must have an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.describe_user_profiles(
    IamUserArns=[
        'string',
    ]
)
```

Parameters **IamUserArns** (*list*) – An array of IAM user ARNs that identify the users to be described.

- *(string)* –

Return type dict

Returns

Response Syntax

```
{
    'UserProfiles': [
        {
```

```
        'IamUserArn': 'string',
        'Name': 'string',
        'SshUsername': 'string',
        'SshPublicKey': 'string',
        'AllowSelfManagement': True|False
    },
]
}
```

Response Structure

- (*dict*) –

Contains the response to a `DescribeUserProfiles` request.

- **UserProfiles** (*list*) –

A `Users` object that describes the specified users.

- * (*dict*) –

Describes a user's SSH information.

- **IamUserArn** (*string*) –

The user's IAM ARN.

- **Name** (*string*) –

The user's name.

- **SshUsername** (*string*) –

The user's SSH user name.

- **SshPublicKey** (*string*) –

The user's SSH public key.

- **AllowSelfManagement** (*boolean*) –

Whether users can specify their own SSH public key through the My Settings page. For more information, see [Managing User Permissions](#).

describe_volumes (***kwargs*)

Describes an instance's Amazon EBS volumes.

Note: You must specify at least one of the parameters.

Required Permissions : To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.describe_volumes(
    InstanceId='string',
    StackId='string',
    RaidArrayId='string',
    VolumeIds=[
        'string',
    ]
)
```

Parameters

- **InstanceId** (*string*) – The instance ID. If you use this parameter, `DescribeVolumes` returns descriptions of the volumes associated with the specified instance.
 - **StackId** (*string*) – A stack ID. The action describes the stack’s registered Amazon EBS volumes.
 - **RaidArrayId** (*string*) – The RAID array ID. If you use this parameter, `DescribeVolumes` returns descriptions of the volumes associated with the specified RAID array.
 - **VolumeIds** (*list*) – An array of volume IDs. If you use this parameter, `DescribeVolumes` returns descriptions of the specified volumes. Otherwise, it returns a description of every volume.
- (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'Volumes': [
    {
      'VolumeId': 'string',
      'Ec2VolumeId': 'string',
      'Name': 'string',
      'RaidArrayId': 'string',
      'InstanceId': 'string',
      'Status': 'string',
      'Size': 123,
      'Device': 'string',
      'MountPoint': 'string',
      'Region': 'string',
      'AvailabilityZone': 'string',
      'VolumeType': 'string',
      'Iops': 123
    },
  ],
}
```

Response Structure

- (*dict*) –

Contains the response to a `DescribeVolumes` request.

- **Volumes** (*list*) –

An array of volume IDs.

- * (*dict*) –

Describes an instance’s Amazon EBS volume.

- **VolumeId** (*string*) –

The volume ID.

- **Ec2VolumeId** (*string*) –

The Amazon EC2 volume ID.

- **Name** (*string*) –

The volume name.

- **RaidArrayId** (*string*) –

The RAID array ID.

- **InstanceId** (*string*) –
The instance ID.
- **Status** (*string*) –
The value returned by [DescribeVolumes](#).
- **Size** (*integer*) –
The volume size.
- **Device** (*string*) –
The device name.
- **MountPoint** (*string*) –
The volume mount point. For example “/dev/sdh”.
- **Region** (*string*) –
The AWS region. For more information about AWS regions, see [Regions and Endpoints](#).
- **AvailabilityZone** (*string*) –
The volume Availability Zone. For more information, see [Regions and Endpoints](#).
- **VolumeType** (*string*) –
The volume type, standard or PIOPS.
- **Iops** (*integer*) –
For PIOPS volumes, the IOPS per disk.

detach_elastic_load_balancer (***kwargs*)

Detaches a specified Elastic Load Balancing instance from its layer.

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.detach_elastic_load_balancer(  
    ElasticLoadBalancerName='string',  
    LayerId='string'  
)
```

Parameters

- **ElasticLoadBalancerName** (*string*) – [REQUIRED]

The Elastic Load Balancing instance’s name.

- **LayerId** (*string*) – [REQUIRED]

The ID of the layer that the Elastic Load Balancing instance is attached to.

Returns None

disassociate_elastic_ip (***kwargs*)

Disassociates an Elastic IP address from its instance. The address remains registered with the stack. For more information, see [Resource Management](#).

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax


```
response = client.disassociate_elastic_ip(
    ElasticIp='string'
)
```

Parameters **ElasticIp** (*string*) – [REQUIRED]

The Elastic IP address.

Returns None

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to *ClientMethod*.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_hostname_suggestion (***kwargs*)

Gets a generated host name for the specified layer, based on the current host name theme.

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.get_hostname_suggestion(
    LayerId='string'
)
```

Parameters **LayerId** (*string*) – [REQUIRED]

The layer ID.

Return type dict

Returns

Response Syntax

```
{
    'LayerId': 'string',
    'Hostname': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a *GetHostnameSuggestion* request.

- **LayerId** (*string*) –

The layer ID.

- **Hostname** (*string*) –

The generated host name.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters `operation_name` (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises `OperationNotPageableError` Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

`get_waiter` (*waiter_name*)

`grant_access` (***kwargs*)

Note: This action can be used only with Windows stacks.

Grants RDP access to a Windows instance for a specified time period.

Request Syntax

```
response = client.grant_access(
    InstanceId='string',
    ValidForInMinutes=123
)
```

Parameters

- **InstanceId** (*string*) – [REQUIRED]

The instance's AWS OpsWorks ID.

- **ValidForInMinutes** (*integer*) – The length of time (in minutes) that the grant is valid. When the grant expires at the end of this period, the user will no longer be able to use the credentials to log in. If the user is logged in at the time, he or she automatically will be logged out.

Return type `dict`

Returns

Response Syntax

```
{
  'TemporaryCredential': {
    'Username': 'string',
    'Password': 'string',
    'ValidForInMinutes': 123,
    'InstanceId': 'string'
  }
}
```

Response Structure

- (*dict*) –

Contains the response to a `GrantAccess` request.

- **TemporaryCredential** (*dict*) –

A `TemporaryCredential` object that contains the data needed to log in to the instance by RDP clients, such as the Microsoft Remote Desktop Connection.

- * **Username** (*string*) –
The user name.
- * **Password** (*string*) –
The password.
- * **ValidForInMinutes** (*integer*) –
The length of time (in minutes) that the grant is valid. When the grant expires, at the end of this period, the user will no longer be able to use the credentials to log in. If they are logged in at the time, they will be automatically logged out.
- * **InstanceId** (*string*) –
The instance's AWS OpsWorks ID.

reboot_instance (**kwargs)

Reboots a specified instance. For more information, see [Starting, Stopping, and Rebooting Instances](#) .

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.reboot_instance(
    InstanceId='string'
)
```

Parameters InstanceId (*string*) – [REQUIRED]

The instance ID.

Returns None

register_ecs_cluster (**kwargs)

Registers a specified Amazon ECS cluster with a stack. You can register only one cluster with a stack. A cluster can be registered with only one stack. For more information, see [Resource Management](#) .

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.register_ecs_cluster(
    EcsClusterArn='string',
    StackId='string'
)
```

Parameters

- **EcsClusterArn** (*string*) – [REQUIRED]

The cluster's ARN.

- **StackId** (*string*) – [REQUIRED]

The stack ID.

Return type dict

Returns

Response Syntax

```
{
    'EcsClusterArn': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a `RegisterEcsCluster` request.

- **EcsClusterArn** (*string*) –

The cluster's ARN.

register_elastic_ip (***kwargs*)

Registers an Elastic IP address with a specified stack. An address can be registered with only one stack at a time. If the address is already registered, you must first deregister it by calling `DeregisterElasticIp`. For more information, see [Resource Management](#).

Required Permissions : To use this action, an IAM user must have a `Manage` permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.register_elastic_ip(
    ElasticIp='string',
    StackId='string'
)
```

Parameters

- **ElasticIp** (*string*) – [REQUIRED]

The Elastic IP address.

- **StackId** (*string*) – [REQUIRED]

The stack ID.

Return type dict

Returns**Response Syntax**

```
{
    'ElasticIp': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a `RegisterElasticIp` request.

- **ElasticIp** (*string*) –

The Elastic IP address.

register_instance (***kwargs*)

Registers instances with a specified stack that were created outside of AWS OpsWorks.

Note: We do not recommend using this action to register instances. The complete registration operation has two primary steps, installing the AWS OpsWorks agent on the instance and registering the instance with the stack. `RegisterInstance` handles only the second step. You should instead use the AWS CLI `register` command, which performs the entire registration operation. For more information, see [Registering an Instance with an AWS OpsWorks Stack](#).

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.register_instance(
    StackId='string',
    Hostname='string',
    PublicIp='string',
    PrivateIp='string',
    RsaPublicKey='string',
    RsaPublicKeyFingerprint='string',
    InstanceIdentity={
        'Document': 'string',
        'Signature': 'string'
    }
)
```

Parameters

- **StackId** (*string*) – [REQUIRED]
The ID of the stack that the instance is to be registered with.
- **Hostname** (*string*) – The instance’s hostname.
- **PublicIp** (*string*) – The instance’s public IP address.
- **PrivateIp** (*string*) – The instance’s private IP address.
- **RsaPublicKey** (*string*) – The instances public RSA key. This key is used to encrypt communication between the instance and the service.
- **RsaPublicKeyFingerprint** (*string*) – The instances public RSA key fingerprint.
- **InstanceIdentity** (*dict*) – An InstanceIdentity object that contains the instance’s identity.
 - **Document** (*string*) –
A JSON document that contains the metadata.
 - **Signature** (*string*) –
A signature that can be used to verify the document’s accuracy and authenticity.

Return type dict

Returns

Response Syntax

```
{
    'InstanceId': 'string'
}
```

Response Structure

- (*dict*) –
Contains the response to a RegisterInstanceResult request.
 - **InstanceId** (*string*) –
The registered instance’s AWS OpsWorks ID.

register_rds_db_instance (***kwargs*)

Registers an Amazon RDS instance with a stack.

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.register_rds_db_instance(  
    StackId='string',  
    RdsDbInstanceArn='string',  
    DbUser='string',  
    DbPassword='string'  
)
```

Parameters

- **StackId** (*string*) – [REQUIRED]
The stack ID.
- **RdsDbInstanceArn** (*string*) – [REQUIRED]
The Amazon RDS instance’s ARN.
- **DbUser** (*string*) – [REQUIRED]
The database’s master user name.
- **DbPassword** (*string*) – [REQUIRED]
The database password.

Returns None

register_volume (**kwargs)

Registers an Amazon EBS volume with a specified stack. A volume can be registered with only one stack at a time. If the volume is already registered, you must first deregister it by calling `DeregisterVolume` . For more information, see [Resource Management](#) .

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.register_volume(  
    Ec2VolumeId='string',  
    StackId='string'  
)
```

Parameters

- **Ec2VolumeId** (*string*) – The Amazon EBS volume ID.
- **StackId** (*string*) – [REQUIRED]
The stack ID.

Return type dict

Returns

Response Syntax

```
{  
    'VolumeId': 'string'  
}
```

Response Structure

- (*dict*) –

Contains the response to a `RegisterVolume` request.

- **VolumeId** (*string*) –

The volume ID.

set_load_based_auto_scaling (***kwargs*)

Specify the load-based auto scaling configuration for a specified layer. For more information, see [Managing Load with Time-based and Load-based Instances](#) .

Note: To use load-based auto scaling, you must create a set of load-based auto scaling instances. Load-based auto scaling operates only on the instances from that set, so you must ensure that you have created enough instances to handle the maximum anticipated load.

Required Permissions : To use this action, an IAM user must have a `Manage` permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.set_load_based_auto_scaling(
    LayerId='string',
    Enable=True|False,
    UpScaling={
        'InstanceCount': 123,
        'ThresholdsWaitTime': 123,
        'IgnoreMetricsTime': 123,
        'CpuThreshold': 123.0,
        'MemoryThreshold': 123.0,
        'LoadThreshold': 123.0,
        'Alarms': [
            'string',
        ]
    },
    DownScaling={
        'InstanceCount': 123,
        'ThresholdsWaitTime': 123,
        'IgnoreMetricsTime': 123,
        'CpuThreshold': 123.0,
        'MemoryThreshold': 123.0,
        'LoadThreshold': 123.0,
        'Alarms': [
            'string',
        ]
    }
)
```

Parameters

- **LayerId** (*string*) – **[REQUIRED]**

The layer ID.

- **Enable** (*boolean*) – Enables load-based auto scaling for the layer.
- **UpScaling** (*dict*) – An `AutoScalingThresholds` object with the upscaling threshold configuration. If the load exceeds these thresholds for a specified amount of time, AWS OpsWorks starts a specified number of instances.
 - **InstanceCount** (*integer*) –

The number of instances to add or remove when the load exceeds a threshold.

- **ThresholdsWaitTime** (*integer*) –

The amount of time, in minutes, that the load must exceed a threshold before more instances are added or removed.

- **IgnoreMetricsTime** (*integer*) –

The amount of time (in minutes) after a scaling event occurs that AWS OpsWorks should ignore metrics and suppress additional scaling events. For example, AWS OpsWorks adds new instances following an upscaling event but the instances won't start reducing the load until they have been booted and configured. There is no point in raising additional scaling events during that operation, which typically takes several minutes. `IgnoreMetricsTime` allows you to direct AWS OpsWorks to suppress scaling events long enough to get the new instances online.

- **CpuThreshold** (*float*) –

The CPU utilization threshold, as a percent of the available CPU.

- **MemoryThreshold** (*float*) –

The memory utilization threshold, as a percent of the available memory.

- **LoadThreshold** (*float*) –

The load threshold. For more information about how load is computed, see [Load \(computing\)](#).

- **Alarms** (*list*) –

Custom Cloudwatch auto scaling alarms, to be used as thresholds. This parameter takes a list of up to five alarm names, which are case sensitive and must be in the same region as the stack.

Note: To use custom alarms, you must update your service role to allow `cloudwatch:DescribeAlarms`. You can either have AWS OpsWorks update the role for you when you first use this feature or you can edit the role manually. For more information, see [Allowing AWS OpsWorks to Act on Your Behalf](#).

* (*string*) –

- **DownScaling** (*dict*) – An `AutoScalingThresholds` object with the down-scaling threshold configuration. If the load falls below these thresholds for a specified amount of time, AWS OpsWorks stops a specified number of instances.

- **InstanceCount** (*integer*) –

The number of instances to add or remove when the load exceeds a threshold.

- **ThresholdsWaitTime** (*integer*) –

The amount of time, in minutes, that the load must exceed a threshold before more instances are added or removed.

- **IgnoreMetricsTime** (*integer*) –

The amount of time (in minutes) after a scaling event occurs that AWS OpsWorks should ignore metrics and suppress additional scaling events. For example, AWS OpsWorks adds new instances following an upscaling event but the instances won't start reducing the load until they have been booted and configured. There is no point in raising additional scaling events during that operation, which typically takes several minutes.

`IgnoreMetricsTime` allows you to direct AWS OpsWorks to suppress scaling events long enough to get the new instances online.

– **CpuThreshold** (*float*) –

The CPU utilization threshold, as a percent of the available CPU.

– **MemoryThreshold** (*float*) –

The memory utilization threshold, as a percent of the available memory.

– **LoadThreshold** (*float*) –

The load threshold. For more information about how load is computed, see [Load \(computing\)](#).

– **Alarms** (*list*) –

Custom Cloudwatch auto scaling alarms, to be used as thresholds. This parameter takes a list of up to five alarm names, which are case sensitive and must be in the same region as the stack.

Note: To use custom alarms, you must update your service role to allow `cloudwatch:DescribeAlarms`. You can either have AWS OpsWorks update the role for you when you first use this feature or you can edit the role manually. For more information, see [Allowing AWS OpsWorks to Act on Your Behalf](#).

* (*string*) –

Returns None

set_permission (***kwargs*)

Specifies a user's permissions. For more information, see [Security and Permissions](#).

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.set_permission(
    StackId='string',
    IamUserArn='string',
    AllowSsh=True|False,
    AllowSudo=True|False,
    Level='string'
)
```

Parameters

- **StackId** (*string*) – [REQUIRED]

The stack ID.

- **IamUserArn** (*string*) – [REQUIRED]

The user's IAM ARN.

- **AllowSsh** (*boolean*) – The user is allowed to use SSH to communicate with the instance.
- **AllowSudo** (*boolean*) – The user is allowed to use **sudo** to elevate privileges.
- **Level** (*string*) – The user's permission level, which must be set to one of the following strings. You cannot set your own permissions level.
 - deny
 - show
 - deploy

- manage
- iam_only

For more information on the permissions associated with these levels, see [Managing User Permissions](#).

Returns None

set_time_based_auto_scaling (**kwargs)

Specify the time-based auto scaling configuration for a specified instance. For more information, see [Managing Load with Time-based and Load-based Instances](#).

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.set_time_based_auto_scaling(
    InstanceId='string',
    AutoScalingSchedule={
        'Monday': {
            'string': 'string'
        },
        'Tuesday': {
            'string': 'string'
        },
        'Wednesday': {
            'string': 'string'
        },
        'Thursday': {
            'string': 'string'
        },
        'Friday': {
            'string': 'string'
        },
        'Saturday': {
            'string': 'string'
        },
        'Sunday': {
            'string': 'string'
        }
    }
)
```

Parameters

- **InstanceId** (*string*) – [REQUIRED]

The instance ID.

- **AutoScalingSchedule** (*dict*) – An AutoScalingSchedule with the instance schedule.

- **Monday** (*dict*) –

The schedule for Monday.

* (*string*) –
· (*string*) –

- **Tuesday** (*dict*) –

The schedule for Tuesday.

* (*string*) –
· (*string*) –

- **Wednesday** (*dict*) –
The schedule for Wednesday.
* (*string*) –
 · (*string*) –
- **Thursday** (*dict*) –
The schedule for Thursday.
* (*string*) –
 · (*string*) –
- **Friday** (*dict*) –
The schedule for Friday.
* (*string*) –
 · (*string*) –
- **Saturday** (*dict*) –
The schedule for Saturday.
* (*string*) –
 · (*string*) –
- **Sunday** (*dict*) –
The schedule for Sunday.
* (*string*) –
 · (*string*) –

Returns None

start_instance (***kwargs*)

Starts a specified instance. For more information, see [Starting, Stopping, and Rebooting Instances](#) .

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.start_instance(
    InstanceId='string'
)
```

Parameters **InstanceId** (*string*) – [REQUIRED]

The instance ID.

Returns None

start_stack (***kwargs*)

Starts a stack's instances.

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.start_stack(
    StackId='string'
)
```

Parameters **StackId** (*string*) – [REQUIRED]

The stack ID.

Returns None

stop_instance (***kwargs*)

Stops a specified instance. When you stop a standard instance, the data disappears and must be reinstalled when you restart the instance. You can stop an Amazon EBS-backed instance without losing data. For more information, see [Starting, Stopping, and Rebooting Instances](#) .

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.stop_instance(  
    InstanceId='string'  
)
```

Parameters **InstanceId** (*string*) – [REQUIRED]

The instance ID.

Returns None

stop_stack (***kwargs*)

Stops a specified stack.

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.stop_stack(  
    StackId='string'  
)
```

Parameters **StackId** (*string*) – [REQUIRED]

The stack ID.

Returns None

unassign_instance (***kwargs*)

Unassigns a registered instance from all of it's layers. The instance remains in the stack as an unassigned instance and can be assigned to another layer, as needed. You cannot use this action with instances that were created with AWS OpsWorks.

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.unassign_instance(  
    InstanceId='string'  
)
```

Parameters **InstanceId** (*string*) – [REQUIRED]

The instance ID.

Returns None

unassign_volume (**kwargs)

Unassigns an assigned Amazon EBS volume. The volume remains registered with the stack. For more information, see [Resource Management](#).

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.unassign_volume(
    VolumeId='string'
)
```

Parameters *VolumeId* (string) – [REQUIRED]

The volume ID.

Returns None

update_app (**kwargs)

Updates a specified app.

Required Permissions : To use this action, an IAM user must have a Deploy or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.update_app(
    AppId='string',
    Name='string',
    Description='string',
    DataSources=[
        {
            'Type': 'string',
            'Arn': 'string',
            'DatabaseName': 'string'
        },
    ],
    Type='aws-flow-ruby'|'java'|'rails'|'php'|'nodejs'|'static'|'other',
    AppSource={
        'Type': 'git'|'svn'|'archive'|'s3',
        'Url': 'string',
        'Username': 'string',
        'Password': 'string',
        'SshKey': 'string',
        'Revision': 'string'
    },
    Domains=[
        'string',
    ],
    EnableSsl=True|False,
    SslConfiguration={
        'Certificate': 'string',
        'PrivateKey': 'string',
        'Chain': 'string'
    },
    Attributes={
        'string': 'string'
    }
```

```
    },
    Environment=[
        {
            'Key': 'string',
            'Value': 'string',
            'Secure': True|False
        },
    ]
)
```

Parameters

- **AppId** (*string*) – [REQUIRED]

The app ID.

- **Name** (*string*) – The app name.
- **Description** (*string*) – A description of the app.
- **DataSources** (*list*) – The app’s data sources.

– (*dict*) –

Describes an app’s data source.

- * **Type** (*string*) –

The data source’s type, `AutoSelectOpsworksMysqlInstance`, `OpsworksMysqlInstance`, or `RdsDbInstance`.

- * **Arn** (*string*) –

The data source’s ARN.

- * **DatabaseName** (*string*) –

The database name.

- **Type** (*string*) – The app type.
- **AppSource** (*dict*) – A Source object that specifies the app repository.

– **Type** (*string*) –

The repository type.

– **Url** (*string*) –

The source URL.

– **Username** (*string*) –

This parameter depends on the repository type.

- * For Amazon S3 bundles, set `Username` to the appropriate IAM access key ID.

- * For HTTP bundles, Git repositories, and Subversion repositories, set `Username` to the user name.

– **Password** (*string*) –

When included in a request, the parameter depends on the repository type.

- * For Amazon S3 bundles, set `Password` to the appropriate IAM secret access key.

- * For HTTP bundles and Subversion repositories, set `Password` to the password.

For more information on how to safely handle IAM credentials, see <http://docs.aws.amazon.com/general/latest/gr/aws-access-keys-best-practices.html>.

In responses, AWS OpsWorks returns `*****FILTERED*****` instead of the actual value.

– **SshKey** (*string*) –

In requests, the repository's SSH key.

In responses, AWS OpsWorks returns *****FILTERED***** instead of the actual value.

– **Revision** (*string*) –

The application's version. AWS OpsWorks enables you to easily deploy new versions of an application. One of the simplest approaches is to have branches or revisions in your repository that represent different versions that can potentially be deployed.

- **Domains** (*list*) – The app's virtual host settings, with multiple domains separated by commas. For example: 'www.example.com, example.com'

– (*string*) –

- **EnableSsl** (*boolean*) – Whether SSL is enabled for the app.

- **SslConfiguration** (*dict*) – An `SslConfiguration` object with the SSL configuration.

– **Certificate** (*string*) – [REQUIRED]

The contents of the certificate's domain.crt file.

– **PrivateKey** (*string*) – [REQUIRED]

The private key; the contents of the certificate's domain.kex file.

– **Chain** (*string*) –

Optional. Can be used to specify an intermediate certificate authority key or client authentication.

- **Attributes** (*dict*) – One or more user-defined key/value pairs to be added to the stack attributes.

– (*string*) –

* (*string*) –

- **Environment** (*list*) – An array of `EnvironmentVariable` objects that specify environment variables to be associated with the app. After you deploy the app, these variables are defined on the associated app server instances. For more information, see [Environment Variables](#).

There is no specific limit on the number of environment variables. However, the size of the associated data structure - which includes the variables' names, values, and protected flag values - cannot exceed 10 KB (10240 Bytes). This limit should accommodate most if not all use cases. Exceeding it will cause an exception with the message, "Environment: is too large (maximum is 10KB)."

Note: This parameter is supported only by Chef 11.10 stacks. If you have specified one or more environment variables, you cannot modify the stack's Chef version.

– (*dict*) –

Represents an app's environment variable.

* **Key** (*string*) – [REQUIRED]

(Required) The environment variable's name, which can consist of up to 64 characters and must be specified. The name can contain upper- and lowercase letters, numbers, and underscores (_), but it must start with a letter or underscore.

* **Value** (*string*) – [REQUIRED]

(Optional) The environment variable's value, which can be left empty. If you specify a value, it can contain up to 256 characters,

which must all be printable.

* **Secure** (*boolean*) –

(Optional) Whether the variable's value will be returned by the DescribeApps action. To conceal an environment variable's value, set Secure to true. DescribeApps then returns *****FILTERED***** instead of the actual value. The default value for Secure is false.

Returns None

update_elastic_ip (***kwargs*)

Updates a registered Elastic IP address's name. For more information, see [Resource Management](#).

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.update_elastic_ip(
    ElasticIp='string',
    Name='string'
)
```

Parameters

- **ElasticIp** (*string*) – [REQUIRED]

The address.

- **Name** (*string*) – The new name.

Returns None

update_instance (***kwargs*)

Updates a specified instance.

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.update_instance(
    InstanceId='string',
    LayerIds=[
        'string',
    ],
    InstanceType='string',
    AutoScalingType='load'|'timer',
    Hostname='string',
    Os='string',
    AmiId='string',
    SshKeyName='string',
    Architecture='x86_64'|'i386',
    InstallUpdatesOnBoot=True|False,
    EbsOptimized=True|False,
    AgentVersion='string'
)
```

Parameters

- **InstanceId** (*string*) – [REQUIRED]

The instance ID.

- **LayerIds** (*list*) – The instance’s layer IDs.
 - (*string*) –
- **InstanceType** (*string*) – The instance type, such as `t2.micro`. For a list of supported instance types, open the stack in the console, choose **Instances**, and choose **+ Instance**. The **Size** list contains the currently supported types. For more information, see [Instance Families and Types](#). The parameter values that you use to specify the various types are in the **API Name** column of the **Available Instance Types** table.
- **AutoScalingType** (*string*) – For load-based or time-based instances, the type. Windows stacks can use only time-based instances.
- **Hostname** (*string*) – The instance host name.
- **Os** (*string*) – The instance’s operating system, which must be set to one of the following.
 - A supported Linux operating system: An Amazon Linux version, such as Amazon Linux 2015.03, Red Hat Enterprise Linux 7, Ubuntu 12.04 LTS, or Ubuntu 14.04 LTS.
 - Microsoft Windows Server 2012 R2 Base.
 - A custom AMI: `Custom`.

For more information on the supported operating systems, see [AWS OpsWorks Operating Systems](#).

The default option is the current Amazon Linux version. If you set this parameter to `Custom`, you must use the `AmiId` parameter to specify the custom AMI that you want to use. For more information on the supported operating systems, see [Operating Systems](#). For more information on how to use custom AMIs with OpsWorks, see [Using Custom AMIs](#).

Note: You can specify a different Linux operating system for the updated stack, but you cannot change from Linux to Windows or Windows to Linux.

-
- **AmiId** (*string*) – A custom AMI ID to be used to create the instance. The AMI must be based on one of the supported operating systems. For more information, see [Instances](#)

Note: If you specify a custom AMI, you must set `Os` to `Custom`.

-
- **SshKeyName** (*string*) – The instance’s Amazon EC2 key name.
 - **Architecture** (*string*) – The instance architecture. Instance types do not necessarily support both architectures. For a list of the architectures that are supported by the different instance types, see [Instance Families and Types](#).
 - **InstallUpdatesOnBoot** (*boolean*) – Whether to install operating system and package updates when the instance boots. The default value is `true`. To control when updates are installed, set this value to `false`. You must then update your instances manually by using `CreateDeployment` to run the `update_dependencies` stack command or by manually running `yum` (Amazon Linux) or `apt-get` (Ubuntu) on the instances.

Note: We strongly recommend using the default value of `true`, to ensure that your instances have the latest security updates.

-
- **EbsOptimized** (*boolean*) – This property cannot be updated.
 - **AgentVersion** (*string*) – The default AWS OpsWorks agent version. You have the following options:

- INHERIT - Use the stack's default agent version setting.
- *version_number* - Use the specified agent version. This value overrides the stack's default setting. To update the agent version, you must edit the instance configuration and specify a new version. AWS OpsWorks then automatically installs that version on the instance.

The default setting is INHERIT . To specify an agent version, you must use the complete version number, not the abbreviated number shown on the console. For a list of available agent version numbers, call DescribeAgentVersions .

Returns None

update_layer (***kwargs*)

Updates a specified layer.

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.update_layer(
    LayerId='string',
    Name='string',
    Shortname='string',
    Attributes={
        'string': 'string'
    },
    CustomInstanceProfileArn='string',
    CustomJson='string',
    CustomSecurityGroupIds=[
        'string',
    ],
    Packages=[
        'string',
    ],
    VolumeConfigurations=[
        {
            'MountPoint': 'string',
            'RaidLevel': 123,
            'NumberOfDisks': 123,
            'Size': 123,
            'VolumeType': 'string',
            'Iops': 123
        },
    ],
    EnableAutoHealing=True|False,
    AutoAssignElasticIps=True|False,
    AutoAssignPublicIps=True|False,
    CustomRecipes={
        'Setup': [
            'string',
        ],
        'Configure': [
            'string',
        ],
        'Deploy': [
            'string',
        ],
        'Undeploy': [
```

```

        'string',
    ],
    'Shutdown': [
        'string',
    ]
},
InstallUpdatesOnBoot=True|False,
UseEbsOptimizedInstances=True|False,
LifecycleEventConfiguration={
    'Shutdown': {
        'ExecutionTimeout': 123,
        'DelayUntilElbConnectionsDrained': True|False
    }
}
)

```

Parameters

- **LayerId** (*string*) – [REQUIRED]

The layer ID.

- **Name** (*string*) – The layer name, which is used by the console.
- **Shortname** (*string*) – For custom layers only, use this parameter to specify the layer’s short name, which is used internally by AWS OpsWorks and by Chef. The short name is also used as the name for the directory where your app files are installed. It can have a maximum of 200 characters and must be in the following format: /A[a-z0-9-_.]+Z/.

The built-in layers’ short names are defined by AWS OpsWorks. For more information, see the [Layer Reference](#)

- **Attributes** (*dict*) – One or more user-defined key/value pairs to be added to the stack attributes.
 - (*string*) –
 - * (*string*) –
- **CustomInstanceProfileArn** (*string*) – The ARN of an IAM profile to be used for all of the layer’s EC2 instances. For more information about IAM ARNs, see [Using Identifiers](#).
- **CustomJson** (*string*) – A JSON-formatted string containing custom stack configuration and deployment attributes to be installed on the layer’s instances. For more information, see [Using Custom JSON](#).
- **CustomSecurityGroupIds** (*list*) – An array containing the layer’s custom security group IDs.
 - (*string*) –
- **Packages** (*list*) – An array of `Package` objects that describe the layer’s packages.
 - (*string*) –
- **VolumeConfigurations** (*list*) – A `VolumeConfigurations` object that describes the layer’s Amazon EBS volumes.
 - (*dict*) –

Describes an Amazon EBS volume configuration.

- * **MountPoint** (*string*) – [REQUIRED]

The volume mount point. For example “/dev/sdh”.

- * **RaidLevel** (*integer*) –

The volume [RAID level](#).

- * **NumberOfDisks** (*integer*) – [REQUIRED]

The number of disks in the volume.

* **Size** (*integer*) – [REQUIRED]

The volume size.

* **VolumeType** (*string*) –

The volume type:

- `standard` - Magnetic
- `io1` - Provisioned IOPS (SSD)
- `gp2` - General Purpose (SSD)

* **Iops** (*integer*) –

For PIOPS volumes, the IOPS per disk.

- **EnableAutoHealing** (*boolean*) – Whether to disable auto healing for the layer.
- **AutoAssignElasticIps** (*boolean*) – Whether to automatically assign an [Elastic IP address](#) to the layer’s instances. For more information, see [How to Edit a Layer](#) .
- **AutoAssignPublicIps** (*boolean*) – For stacks that are running in a VPC, whether to automatically assign a public IP address to the layer’s instances. For more information, see [How to Edit a Layer](#) .
- **CustomRecipes** (*dict*) – A `LayerCustomRecipes` object that specifies the layer’s custom recipes.
 - **Setup** (*list*) –
An array of custom recipe names to be run following a `setup` event.
* (*string*) –
 - **Configure** (*list*) –
An array of custom recipe names to be run following a `configure` event.
* (*string*) –
 - **Deploy** (*list*) –
An array of custom recipe names to be run following a `deploy` event.
* (*string*) –
 - **Undeploy** (*list*) –
An array of custom recipe names to be run following a `undeploy` event.
* (*string*) –
 - **Shutdown** (*list*) –
An array of custom recipe names to be run following a `shutdown` event.
* (*string*) –

- **InstallUpdatesOnBoot** (*boolean*) – Whether to install operating system and package updates when the instance boots. The default value is `true` . To control when updates are installed, set this value to `false` . You must then update your instances manually by using `CreateDeployment` to run the `update_dependencies` stack command or manually running `yum` (Amazon Linux) or `apt-get` (Ubuntu) on the instances.

Note: We strongly recommend using the default value of `true` , to ensure that your instances have the latest security updates.

- **UseEbsOptimizedInstances** (*boolean*) – Whether to use Amazon EBS-optimized instances.
- **LifecycleEventConfiguration** (*dict*) –
 - **Shutdown** (*dict*) –
A `ShutdownEventConfiguration` object that specifies the Shutdown event configuration.

* **ExecutionTimeout** (*integer*) –

The time, in seconds, that AWS OpsWorks will wait after triggering a Shutdown event before shutting down an instance.

* **DelayUntilElbConnectionsDrained** (*boolean*) –

Whether to enable Elastic Load Balancing connection draining. For more information, see [Connection Draining](#)

Returns None

update_my_user_profile (***kwargs*)

Updates a user's SSH public key.

Required Permissions : To use this action, an IAM user must have self-management enabled or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.update_my_user_profile(
    SshPublicKey='string'
)
```

Parameters **SshPublicKey** (*string*) – The user's SSH public key.

Returns None

update_rds_db_instance (***kwargs*)

Updates an Amazon RDS instance.

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.update_rds_db_instance(
    RdsDbInstanceArn='string',
    DbUser='string',
    DbPassword='string'
)
```

Parameters

- **RdsDbInstanceArn** (*string*) – **[REQUIRED]**

The Amazon RDS instance's ARN.

- **DbUser** (*string*) – The master user name.

- **DbPassword** (*string*) – The database password.

Returns None

update_stack (***kwargs*)

Updates a specified stack.

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = client.update_stack(
    StackId='string',
    Name='string',
)
```

```
Attributes={
    'string': 'string'
},
ServiceRoleArn='string',
DefaultInstanceProfileArn='string',
DefaultOs='string',
HostnameTheme='string',
DefaultAvailabilityZone='string',
DefaultSubnetId='string',
CustomJson='string',
ConfigurationManager={
    'Name': 'string',
    'Version': 'string'
},
ChefConfiguration={
    'ManageBerkshelf': True|False,
    'BerkshelfVersion': 'string'
},
UseCustomCookbooks=True|False,
CustomCookbooksSource={
    'Type': 'git'|'svn'|'archive'|'s3',
    'Url': 'string',
    'Username': 'string',
    'Password': 'string',
    'SshKey': 'string',
    'Revision': 'string'
},
DefaultSshKeyName='string',
DefaultRootDeviceType='ebs'|'instance-store',
UseOpsworksSecurityGroups=True|False,
AgentVersion='string'
)
```

Parameters

- **StackId** (*string*) – [REQUIRED]

The stack ID.

- **Name** (*string*) – The stack’s new name.
- **Attributes** (*dict*) – One or more user-defined key-value pairs to be added to the stack attributes.
 - (*string*) –
 - * (*string*) –
- **ServiceRoleArn** (*string*) – Do not use this parameter. You cannot update a stack’s service role.
- **DefaultInstanceProfileArn** (*string*) – The ARN of an IAM profile that is the default profile for all of the stack’s EC2 instances. For more information about IAM ARNs, see [Using Identifiers](#).
- **DefaultOs** (*string*) – The stack’s operating system, which must be set to one of the following:
 - A supported Linux operating system: An Amazon Linux version, such as Amazon Linux 2015.03, Red Hat Enterprise Linux 7, Ubuntu 12.04 LTS, or Ubuntu 14.04 LTS.
 - Microsoft Windows Server 2012 R2 Base.
 - A custom AMI: Custom. You specify the custom AMI you want to use when you create instances. For more information on how to use custom AMIs with OpsWorks, see [Using Custom AMIs](#).

The default option is the stack’s current operating system. For more information

on the supported operating systems, see [AWS OpsWorks Operating Systems](#) .

- **HostnameTheme** (*string*) – The stack’s new host name theme, with spaces replaced by underscores. The theme is used to generate host names for the stack’s instances. By default, `HostnameTheme` is set to `Layer_Dependent` , which creates host names by appending integers to the layer’s short name. The other themes are:

- `Baked_Goods`
- `Clouds`
- `Europe_Cities`
- `Fruits`
- `Greek_Deities`
- `Legendary_creatures_from_Japan`
- `Planets_and_Moons`
- `Roman_Deities`
- `Scottish_Islands`
- `US_Cities`
- `Wild_Cats`

To obtain a generated host name, call `GetHostNameSuggestion` , which returns a host name based on the current theme.

- **DefaultAvailabilityZone** (*string*) – The stack’s default Availability Zone, which must be in the stack’s region. For more information, see [Regions and Endpoints](#) . If you also specify a value for `DefaultSubnetId` , the subnet must be in the same zone. For more information, see `CreateStack` .
- **DefaultSubnetId** (*string*) – The stack’s default VPC subnet ID. This parameter is required if you specify a value for the `VpcId` parameter. All instances are launched into this subnet unless you specify otherwise when you create the instance. If you also specify a value for `DefaultAvailabilityZone` , the subnet must be in that zone. For information on default values and when this parameter is required, see the `VpcId` parameter description.
- **CustomJson** (*string*) – A string that contains user-defined, custom JSON. It can be used to override the corresponding default stack configuration JSON values or to pass data to recipes. The string should be in the following format and escape characters such as “”:

```
{\"key1\": \"value1\", \"key2\":
 \"value2\", ...}
```

For more information on custom JSON, see [Use Custom JSON to Modify the Stack Configuration Attributes](#) .

- **ConfigurationManager** (*dict*) – The configuration manager. When you clone a stack, we recommend that you use the configuration manager to specify the Chef version: 0.9, 11.4, or 11.10. The default value is currently 11.4.
 - **Name** (*string*) –
The name. This parameter must be set to “Chef”.
 - **Version** (*string*) –
The Chef version. This parameter must be set to 0.9, 11.4, or 11.10. The default value is 11.4.
- **ChefConfiguration** (*dict*) – A `ChefConfiguration` object that specifies whether to enable Berkshelf and the Berkshelf version on Chef 11.10 stacks. For more information, see [Create a New Stack](#) .
 - **ManageBerkshelf** (*boolean*) –
Whether to enable Berkshelf.
 - **BerkshelfVersion** (*string*) –
The Berkshelf version.

- **UseCustomCookbooks** (*boolean*) – Whether the stack uses custom cookbooks.
- **CustomCookbooksSource** (*dict*) – Contains the information required to retrieve an app or cookbook from a repository. For more information, see [Creating Apps](#) or [Custom Recipes and Cookbooks](#).

- **Type** (*string*) –

The repository type.

- **Url** (*string*) –

The source URL.

- **Username** (*string*) –

This parameter depends on the repository type.

- * For Amazon S3 bundles, set `Username` to the appropriate IAM access key ID.

- * For HTTP bundles, Git repositories, and Subversion repositories, set `Username` to the user name.

- **Password** (*string*) –

When included in a request, the parameter depends on the repository type.

- * For Amazon S3 bundles, set `Password` to the appropriate IAM secret access key.

- * For HTTP bundles and Subversion repositories, set `Password` to the password.

For more information on how to safely handle IAM credentials, see <http://docs.aws.amazon.com/general/latest/gr/aws-access-keys-best-practices.html>.

In responses, AWS OpsWorks returns `*****FILTERED*****` instead of the actual value.

- **SshKey** (*string*) –

In requests, the repository’s SSH key.

In responses, AWS OpsWorks returns `*****FILTERED*****` instead of the actual value.

- **Revision** (*string*) –

The application’s version. AWS OpsWorks enables you to easily deploy new versions of an application. One of the simplest approaches is to have branches or revisions in your repository that represent different versions that can potentially be deployed.

- **DefaultSshKeyName** (*string*) – A default Amazon EC2 key-pair name. The default value is `none`. If you specify a key-pair name, AWS OpsWorks installs the public key on the instance and you can use the private key with an SSH client to log in to the instance. For more information, see [Using SSH to Communicate with an Instance](#) and [Managing SSH Access](#). You can override this setting by specifying a different key pair, or no key pair, when you [create an instance](#).
- **DefaultRootDeviceType** (*string*) – The default root device type. This value is used by default for all instances in the stack, but you can override it when you create an instance. For more information, see [Storage for the Root Device](#).
- **UseOpsworksSecurityGroups** (*boolean*) – Whether to associate the AWS OpsWorks built-in security groups with the stack’s layers.

AWS OpsWorks provides a standard set of built-in security groups, one for each layer, which are associated with layers by default. `UseOpsworksSecurityGroups` allows you to provide your own custom security groups instead of using the built-in groups.

UseOpsworksSecurityGroups has the following settings:

- True - AWS OpsWorks automatically associates the appropriate built-in security group with each layer (default setting). You can associate additional security groups with a layer after you create it, but you cannot delete the built-in security group.
- False - AWS OpsWorks does not associate built-in security groups with layers. You must create appropriate EC2 security groups and associate a security group with each layer that you create. However, you can still manually associate a built-in security group with a layer on. Custom security groups are required only for those layers that need custom settings.

For more information, see [Create a New Stack](#).

- **AgentVersion** (*string*) – The default AWS OpsWorks agent version. You have the following options:
 - Auto-update - Set this parameter to LATEST . AWS OpsWorks automatically installs new agent versions on the stack’s instances as soon as they are available.
 - Fixed version - Set this parameter to your preferred agent version. To update the agent version, you must edit the stack configuration and specify a new version. AWS OpsWorks then automatically installs that version on the stack’s instances.

The default setting is LATEST . To specify an agent version, you must use the complete version number, not the abbreviated number shown on the console. For a list of available agent version numbers, call DescribeAgentVersions .

Note: You can also specify an agent version when you create or update an instance, which overrides the stack’s default setting.

Returns None

update_user_profile (***kwargs*)

Updates a specified user profile.

Required Permissions : To use this action, an IAM user must have an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.update_user_profile(
    IamUserArn='string',
    SshUsername='string',
    SshPublicKey='string',
    AllowSelfManagement=True|False
)
```

Parameters

- **IamUserArn** (*string*) – [REQUIRED]

The user IAM ARN.

- **SshUsername** (*string*) – The user’s SSH user name. The allowable characters are [a-z], [A-Z], [0-9], ‘-’, and ‘_’. If the specified name includes other punctuation marks, AWS OpsWorks removes them. For example, my . name will be changed to myname . If you do not specify an SSH user name, AWS OpsWorks generates one from the IAM user name.
- **SshPublicKey** (*string*) – The user’s new SSH public key.
- **AllowSelfManagement** (*boolean*) – Whether users can specify their own SSH public key through the My Settings page. For more information, see [Managing User Permissions](#).

Returns None

update_volume (***kwargs*)

Updates an Amazon EBS volume's name or mount point. For more information, see [Resource Management](#).

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = client.update_volume(
    VolumeId='string',
    Name='string',
    MountPoint='string'
)
```

Parameters

- **VolumeId** (*string*) – [REQUIRED]
The volume ID.
- **Name** (*string*) – The new name.
- **MountPoint** (*string*) – The new mount point.

Returns None

Service Resource

class `OpsWorks.ServiceResource`

A resource representing AWS OpsWorks:

```
import boto3

opsworks = boto3.resource('opsworks')
```

These are the resource's available actions:

- `create_stack()`

These are the resource's available sub-resources:

- `Layer()`
- `Stack()`
- `StackSummary()`

These are the resource's available collections:

- `stacks`

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

create_stack (***kwargs*)

Creates a new stack. For more information, see [Create a New Stack](#).

Required Permissions : To use this action, an IAM user must have an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```

stack = opsworks.create_stack(
    Name='string',
    Region='string',
    VpcId='string',
    Attributes={
        'string': 'string'
    },
    ServiceRoleArn='string',
    DefaultInstanceProfileArn='string',
    DefaultOs='string',
    HostnameTheme='string',
    DefaultAvailabilityZone='string',
    DefaultSubnetId='string',
    CustomJson='string',
    ConfigurationManager={
        'Name': 'string',
        'Version': 'string'
    },
    ChefConfiguration={
        'ManageBerkshelf': True|False,
        'BerkshelfVersion': 'string'
    },
    UseCustomCookbooks=True|False,
    UseOpsworksSecurityGroups=True|False,
    CustomCookbooksSource={
        'Type': 'git'|'svn'|'archive'|'s3',
        'Url': 'string',
        'Username': 'string',
        'Password': 'string',
        'SshKey': 'string',
        'Revision': 'string'
    },
    DefaultSshKeyName='string',
    DefaultRootDeviceType='ebs'|'instance-store',
    AgentVersion='string'
)

```

Parameters

- **Name** (*string*) – [REQUIRED]

The stack name.

- **Region** (*string*) – [REQUIRED]

The stack’s AWS region, such as “us-east-1”. For more information about Amazon regions, see [Regions and Endpoints](#) .

- **VpcId** (*string*) – The ID of the VPC that the stack is to be launched into. The VPC must be in the stack’s region. All instances are launched into this VPC. You cannot change the ID later.
 - If your account supports EC2-Classic, the default value is `no VPC` .
 - If your account does not support EC2-Classic, the default value is the default VPC for the specified region.

If the VPC ID corresponds to a default VPC and you have specified either the `DefaultAvailabilityZone` or the `DefaultSubnetId` parameter only, AWS OpsWorks infers the value of the other parameter. If you specify neither parameter, AWS OpsWorks sets these parameters to the first valid Availability Zone for the specified region and the corresponding default VPC subnet ID, respectively.

If you specify a nondefault VPC ID, note the following:

- It must belong to a VPC in your account that is in the specified region.
- You must specify a value for `DefaultSubnetId`.

For more information on how to use AWS OpsWorks with a VPC, see [Running a Stack in a VPC](#). For more information on default VPC and EC2-Classical, see [Supported Platforms](#).

- **Attributes** (*dict*) – One or more user-defined key-value pairs to be added to the stack attributes.
 - (*string*) –
 - * (*string*) –

- **ServiceRoleArn** (*string*) – [REQUIRED]

The stack's AWS Identity and Access Management (IAM) role, which allows AWS OpsWorks to work with AWS resources on your behalf. You must set this parameter to the Amazon Resource Name (ARN) for an existing IAM role. For more information about IAM ARNs, see [Using Identifiers](#).

- **DefaultInstanceProfileArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of an IAM profile that is the default profile for all of the stack's EC2 instances. For more information about IAM ARNs, see [Using Identifiers](#).

- **DefaultOs** (*string*) – The stack's default operating system, which is installed on every instance unless you specify a different operating system when you create the instance. You can specify one of the following.
 - A supported Linux operating system: An Amazon Linux version, such as Amazon Linux 2015.03, Red Hat Enterprise Linux 7, Ubuntu 12.04 LTS, or Ubuntu 14.04 LTS.
 - Microsoft Windows Server 2012 R2 Base.
 - A custom AMI: `Custom`. You specify the custom AMI you want to use when you create instances. For more information, see [Using Custom AMIs](#).

The default option is the current Amazon Linux version. For more information on the supported operating systems, see [AWS OpsWorks Operating Systems](#).

- **HostnameTheme** (*string*) – The stack's host name theme, with spaces replaced by underscores. The theme is used to generate host names for the stack's instances. By default, `HostnameTheme` is set to `Layer_Dependent`, which creates host names by appending integers to the layer's short name. The other themes are:
 - `Baked_Goods`
 - `Clouds`
 - `Europe_Cities`
 - `Fruits`
 - `Greek_Deities`
 - `Legendary_creatures_from_Japan`
 - `Planets_and_Moons`
 - `Roman_Deities`
 - `Scottish_Islands`
 - `US_Cities`
 - `Wild_Cats`

To obtain a generated host name, call `GetHostNameSuggestion`, which returns a host name based on the current theme.

- **DefaultAvailabilityZone** (*string*) – The stack's default Availability Zone, which must be in the specified region. For more information, see [Regions and End-points](#). If you also specify a value for `DefaultSubnetId`, the subnet must be in the same zone. For more information, see the `VpcId` parameter descrip-

tion.

- **DefaultSubnetId** (*string*) – The stack’s default VPC subnet ID. This parameter is required if you specify a value for the `VpcId` parameter. All instances are launched into this subnet unless you specify otherwise when you create the instance. If you also specify a value for `DefaultAvailabilityZone`, the subnet must be in that zone. For information on default values and when this parameter is required, see the `VpcId` parameter description.
- **CustomJson** (*string*) – A string that contains user-defined, custom JSON. It can be used to override the corresponding default stack configuration attribute values or to pass data to recipes. The string should be in the following escape characters such as `""`:

```
{\"key1\": \"value1\", \"key2\":
 \"value2\", ...}
```

For more information on custom JSON, see [Use Custom JSON to Modify the Stack Configuration Attributes](#).

- **ConfigurationManager** (*dict*) – The configuration manager. When you clone a stack we recommend that you use the configuration manager to specify the Chef version: 0.9, 11.4, or 11.10. The default value is currently 11.4.
 - **Name** (*string*) –

The name. This parameter must be set to “Chef”.
 - **Version** (*string*) –

The Chef version. This parameter must be set to 0.9, 11.4, or 11.10. The default value is 11.4.
- **ChefConfiguration** (*dict*) – A `ChefConfiguration` object that specifies whether to enable Berkshelf and the Berkshelf version on Chef 11.10 stacks. For more information, see [Create a New Stack](#).
 - **ManageBerkshelf** (*boolean*) –

Whether to enable Berkshelf.
 - **BerkshelfVersion** (*string*) –

The Berkshelf version.
- **UseCustomCookbooks** (*boolean*) – Whether the stack uses custom cookbooks.
- **UseOpsworksSecurityGroups** (*boolean*) – Whether to associate the AWS OpsWorks built-in security groups with the stack’s layers.

AWS OpsWorks provides a standard set of built-in security groups, one for each layer, which are associated with layers by default. With `UseOpsworksSecurityGroups` you can instead provide your own custom security groups. `UseOpsworksSecurityGroups` has the following settings:

- True - AWS OpsWorks automatically associates the appropriate built-in security group with each layer (default setting). You can associate additional security groups with a layer after you create it, but you cannot delete the built-in security group.
- False - AWS OpsWorks does not associate built-in security groups with layers. You must create appropriate EC2 security groups and associate a security group with each layer that you create. However, you can still manually associate a built-in security group with a layer on creation; custom security groups are required only for those layers that need custom settings.

For more information, see [Create a New Stack](#).

- **CustomCookbooksSource** (*dict*) – Contains the information required to retrieve an app or cookbook from a repository. For more information, see [Creating Apps](#)

or [Custom Recipes and Cookbooks](#) .

- **Type** (*string*) –

The repository type.

- **Url** (*string*) –

The source URL.

- **Username** (*string*) –

This parameter depends on the repository type.

- * For Amazon S3 bundles, set `Username` to the appropriate IAM access key ID.

- * For HTTP bundles, Git repositories, and Subversion repositories, set `Username` to the user name.

- **Password** (*string*) –

When included in a request, the parameter depends on the repository type.

- * For Amazon S3 bundles, set `Password` to the appropriate IAM secret access key.

- * For HTTP bundles and Subversion repositories, set `Password` to the password.

For more information on how to safely handle IAM credentials, see <http://docs.aws.amazon.com/general/latest/gr/aws-access-keys-best-practices.html> .

In responses, AWS OpsWorks returns `*****FILTERED*****` instead of the actual value.

- **SshKey** (*string*) –

In requests, the repository’s SSH key.

In responses, AWS OpsWorks returns `*****FILTERED*****` instead of the actual value.

- **Revision** (*string*) –

The application’s version. AWS OpsWorks enables you to easily deploy new versions of an application. One of the simplest approaches is to have branches or revisions in your repository that represent different versions that can potentially be deployed.

- **DefaultSshKeyName** (*string*) – A default Amazon EC2 key pair name. The default value is none. If you specify a key pair name, AWS OpsWorks installs the public key on the instance and you can use the private key with an SSH client to log in to the instance. For more information, see [Using SSH to Communicate with an Instance](#) and [Managing SSH Access](#) . You can override this setting by specifying a different key pair, or no key pair, when you [create an instance](#) .
- **DefaultRootDeviceType** (*string*) – The default root device type. This value is the default for all instances in the stack, but you can override it when you create an instance. The default option is `instance-store` . For more information, see [Storage for the Root Device](#) .
- **AgentVersion** (*string*) – The default AWS OpsWorks agent version. You have the following options:
 - Auto-update - Set this parameter to `LATEST` . AWS OpsWorks automatically installs new agent versions on the stack’s instances as soon as they are available.
 - Fixed version - Set this parameter to your preferred agent version. To update the agent version, you must edit the stack configuration and specify a new version. AWS OpsWorks then automatically installs that version on the stack’s instances.

The default setting is `LATEST` . To specify an agent version, you must use the complete version number, not the abbreviated number shown on the console. For a list of available agent version numbers, call `DescribeAgentVersions` .

Note: You can also specify an agent version when you create or update an instance, which overrides the stack's default setting.

Return type `opsworks.Stack`

Returns Stack resource

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Layer (*id*)

Creates a Layer resource.:

```
layer = opsworks.Layer('id')
```

Parameters *id* (*string*) – The Layer's id identifier. This **must** be set.

Return type `OpsWorks.Layer`

Returns A Layer resource

Stack (*id*)

Creates a Stack resource.:

```
stack = opsworks.Stack('id')
```

Parameters *id* (*string*) – The Stack's id identifier. This **must** be set.

Return type `OpsWorks.Stack`

Returns A Stack resource

StackSummary (*stack_id*)

Creates a StackSummary resource.:

```
stack_summary = opsworks.StackSummary('stack_id')
```

Parameters *stack_id* (*string*) – The StackSummary's stack_id identifier. This **must** be set.

Return type `OpsWorks.StackSummary`

Returns A StackSummary resource

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

stacks

all ()

Creates an iterable of all Stack resources in the collection.

Request Syntax

```
stack_iterator = opsworks.stacks.all()
```

Return type `list(opsworks.Stack)`

Returns A list of Stack resources

filter (***kwargs*)

Creates an iterable of all Stack resources in the collection filtered by kwargs passed to method.

Request Syntax

```
stack_iterator = opsworks.stacks.filter(
    StackIds=[
        'string',
    ]
)
```

Parameters **StackIds** (*list*) – An array of stack IDs that specify the stacks to be described. If you omit this parameter, `DescribeStacks` returns a description of every stack.

- (*string*) –

Return type `list(opsworks.Stack)`

Returns A list of Stack resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Stack resources in the collection.

Request Syntax

```
stack_iterator = opsworks.stacks.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(opsworks.Stack)`

Returns A list of Stack resources

page_size (***kwargs*)

Creates an iterable of all Stack resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
stack_iterator = opsworks.stacks.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(opsworks.Stack)`

Returns A list of Stack resources

Layer

class `OpsWorks.Layer` (*id*)

A resource representing an AWS OpsWorks Layer:

```
import boto3
```

```
opsworks = boto3.resource('opsworks')
layer = opsworks.Layer('id')
```

Parameters **id** (*string*) – The Layer's id identifier. This **must** be set.

These are the resource's available identifiers:

- `id`

These are the resource's available attributes:

- `attributes`
- `auto_assign_elastic_ips`
- `auto_assign_public_ips`
- `created_at`
- `custom_instance_profile_arn`
- `custom_json`
- `custom_recipes`
- `custom_security_group_ids`
- `default_recipes`
- `default_security_group_names`
- `enable_auto_healing`
- `install_updates_on_boot`
- `layer_id`
- `lifecycle_event_configuration`
- `name`
- `packages`
- `shortname`
- `stack_id`
- `type`
- `use_ebs_optimized_instances`
- `volume_configurations`

These are the resource's available references:

- `stack`

These are the resource's available actions:

- `delete()`
- `load()`
- `reload()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

id

(*string*) The Layer's id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

attributes

(*dict*)

The layer attributes.

For the `HaproxyStatsPassword`, `MysqlRootPassword`, and `GangliaPassword` attributes, AWS OpsWorks returns `*****FILTERED*****` instead of the actual value

For an ECS Cluster layer, AWS OpsWorks the `EcsClusterArn` attribute is set to the cluster's ARN.

auto_assign_elastic_ips

(*boolean*)

Whether to automatically assign an [Elastic IP address](#) to the layer's instances. For more information, see [How to Edit a Layer](#).

auto_assign_public_ips*(boolean)*

For stacks that are running in a VPC, whether to automatically assign a public IP address to the layer's instances. For more information, see [How to Edit a Layer](#) .

created_at*(string)*

Date when the layer was created.

custom_instance_profile_arn*(string)*

The ARN of the default IAM profile to be used for the layer's EC2 instances. For more information about IAM ARNs, see [Using Identifiers](#) .

custom_json*(string)*

A JSON formatted string containing the layer's custom stack configuration and deployment attributes.

custom_recipes*(dict)*

A `LayerCustomRecipes` object that specifies the layer's custom recipes.

custom_security_group_ids*(list)*

An array containing the layer's custom security group IDs.

default_recipes*(dict)*

AWS OpsWorks supports five lifecycle events: **setup** , **configuration** , **deploy** , **undeploy** , and **shutdown** . For each layer, AWS OpsWorks runs a set of standard recipes for each event. In addition, you can provide custom recipes for any or all layers and events. AWS OpsWorks runs custom event recipes after the standard recipes. `LayerCustomRecipes` specifies the custom recipes for a particular layer to be run in response to each of the five events.

To specify a recipe, use the cookbook's directory name in the repository followed by two colons and the recipe name, which is the recipe's file name without the `.rb` extension. For example: `phpapp2::dbsetup` specifies the `dbsetup.rb` recipe in the repository's `phpapp2` folder.

default_security_group_names*(list)*

An array containing the layer's security group names.

enable_auto_healing*(boolean)*

Whether auto healing is disabled for the layer.

install_updates_on_boot*(boolean)*

Whether to install operating system and package updates when the instance boots. The default value is `true` . If this value is set to `false` , you must then update your instances manually by using `CreateDeployment` to run the `update_dependencies` stack command or manually running `yum` (Amazon Linux) or `apt-get` (Ubuntu) on the instances.

Note: We strongly recommend using the default value of `true` , to ensure that your instances have the

latest security updates.

layer_id

(string)

The layer ID.

lifecycle_event_configuration

(dict)

A `LifeCycleEventConfiguration` object that specifies the Shutdown event configuration.

name

(string)

The layer name.

packages

(list)

An array of `Package` objects that describe the layer's packages.

shortname

(string)

The layer short name.

stack_id

(string)

The layer stack ID.

type

(string)

The layer type.

use_ebs_optimized_instances

(boolean)

Whether the layer uses Amazon EBS-optimized instances.

volume_configurations

(list)

A `VolumeConfigurations` object that describes the layer's Amazon EBS volumes.

References

References are related resource instances that have a belongs-to relationship. For more information about references refer to the [Resources Introduction Guide](#).

stack

(Stack) The related stack if set, otherwise None.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

delete()

Deletes a specified layer. You must first stop and then delete all associated instances or unassign registered instances. For more information, see [How to Delete a Layer](#).

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
response = layer.delete()
```

Returns None

load()

Calls `opsworks.Client.describe_layers()` to update the attributes of the Layer resource

Request Syntax

```
layer.load()
```

Returns None

reload()

Calls `opsworks.Client.describe_layers()` to update the attributes of the Layer resource

Request Syntax

```
layer.reload()
```

Returns None

Stack

class `OpsWorks.Stack(id)`

A resource representing an AWS OpsWorks Stack:

```
import boto3
```

```
opsworks = boto3.resource('opsworks')
stack = opsworks.Stack('id')
```

Parameters `id` (*string*) – The Stack’s id identifier. This **must** be set.

These are the resource’s available identifiers:

- `id`

These are the resource’s available attributes:

- `agent_version`
- `arn`
- `attributes`
- `chef_configuration`
- `configuration_manager`
- `created_at`
- `custom_cookbooks_source`
- `custom_json`
- `default_availability_zone`
- `default_instance_profile_arn`
- `default_os`
- `default_root_device_type`
- `default_ssh_key_name`
- `default_subnet_id`

- hostname_theme
- name
- region
- service_role_arn
- stack_id
- use_custom_cookbooks
- use_opsworks_security_groups
- vpc_id

These are the resource's available actions:

- create_layer()
- delete()
- load()
- reload()

These are the resource's available sub-resources:

- Summary()

These are the resource's available collections:

- layers

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

id

(string) The Stack's id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

agent_version

(string)

The agent version. This parameter is set to LATEST for auto-update, or a version number for a fixed agent version.

arn

(string)

The stack's ARN.

attributes

(dict)

The stack's attributes.

chef_configuration

(dict)

A `ChefConfiguration` object that specifies whether to enable Berkshelf and the Berkshelf version. For more information, see [Create a New Stack](#).

configuration_manager

(dict)

The configuration manager.

created_at

(string)

The date when the stack was created.

custom_cookbooks_source*(dict)*

Contains the information required to retrieve an app or cookbook from a repository. For more information, see [Creating Apps](#) or [Custom Recipes and Cookbooks](#) .

custom_json*(string)*

A JSON object that contains user-defined attributes to be added to the stack configuration and deployment attributes. You can use custom JSON to override the corresponding default stack configuration attribute values or to pass data to recipes. The string should be in the following format and must escape characters such as “”:

```
"{\"key1\": \"value1\", \"key2\": \"value2\", ...}"
```

For more information on custom JSON, see [Use Custom JSON to Modify the Stack Configuration Attributes](#) .

default_availability_zone*(string)*

The stack’s default Availability Zone. For more information, see [Regions and Endpoints](#) .

default_instance_profile_arn*(string)*

The ARN of an IAM profile that is the default profile for all of the stack’s EC2 instances. For more information about IAM ARNs, see [Using Identifiers](#) .

default_os*(string)*

The stack’s default operating system.

default_root_device_type*(string)*

The default root device type. This value is used by default for all instances in the stack, but you can override it when you create an instance. For more information, see [Storage for the Root Device](#) .

default_ssh_key_name*(string)*

A default Amazon EC2 key pair for the stack’s instances. You can override this value when you create or update an instance.

default_subnet_id*(string)*

The default subnet ID; applicable only if the stack is running in a VPC.

hostname_theme*(string)*

The stack host name theme, with spaces replaced by underscores.

name*(string)*

The stack name.

region*(string)*

The stack AWS region, such as “us-east-1”. For more information about AWS regions, see [Regions and Endpoints](#) .

service_role_arn

(string)

The stack AWS Identity and Access Management (IAM) role.

stack_id

(string)

The stack ID.

use_custom_cookbooks

(boolean)

Whether the stack uses custom cookbooks.

use_opsworks_security_groups

(boolean)

Whether the stack automatically associates the AWS OpsWorks built-in security groups with the stack’s layers.

vpc_id

(string)

The VPC ID; applicable only if the stack is running in a VPC.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

create_layer (**kwargs)

Creates a layer. For more information, see [How to Create a Layer](#) .

Note: You should use **CreateLayer** for noncustom layer types such as PHP App Server only if the stack does not have an existing layer of that type. A stack can have at most one instance of each noncustom layer; if you attempt to create a second instance, **CreateLayer** fails. A stack can have an arbitrary number of custom layers, so you can call **CreateLayer** as many times as you like for that layer type.

Required Permissions : To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#) .

Request Syntax

```
layer = stack.create_layer(
    Type='aws-flow-ruby'|'ecs-cluster'|'java-app'|'lb'|'web'|'php-app'|'rails-app'|'nodejs-
    Name='string',
    Shortname='string',
    Attributes={
        'string': 'string'
    },
    CustomInstanceProfileArn='string',
    CustomJson='string',
    CustomSecurityGroupIds=[
        'string',
    ],
    Packages=[
        'string',
```

```
],
VolumeConfigurations=[
    {
        'MountPoint': 'string',
        'RaidLevel': 123,
        'NumberOfDisks': 123,
        'Size': 123,
        'VolumeType': 'string',
        'Iops': 123
    },
],
EnableAutoHealing=True|False,
AutoAssignElasticIps=True|False,
AutoAssignPublicIps=True|False,
CustomRecipes={
    'Setup': [
        'string',
    ],
    'Configure': [
        'string',
    ],
    'Deploy': [
        'string',
    ],
    'Undeploy': [
        'string',
    ],
    'Shutdown': [
        'string',
    ]
},
InstallUpdatesOnBoot=True|False,
UseEbsOptimizedInstances=True|False,
LifecycleEventConfiguration={
    'Shutdown': {
        'ExecutionTimeout': 123,
        'DelayUntilElbConnectionsDrained': True|False
    }
}
)
```

Parameters

- **Type** (*string*) – [REQUIRED]

The layer type. A stack cannot have more than one built-in layer of the same type. It can have any number of custom layers.

- **Name** (*string*) – [REQUIRED]

The layer name, which is used by the console.

- **Shortname** (*string*) – [REQUIRED]

For custom layers only, use this parameter to specify the layer's short name, which is used internally by AWS OpsWorks and by Chef recipes. The short name is also used as the name for the directory where your app files are installed. It can have a maximum of 200 characters, which are limited to the alphanumeric characters, '-', '_', and '.'.

The built-in layers' short names are defined by AWS OpsWorks. For more information, see the [Layer Reference](#).

- **Attributes** (*dict*) – One or more user-defined key-value pairs to be added to the stack attributes.

To create a cluster layer, set the `EcsClusterArn` attribute to the cluster's ARN.

- (*string*) –
- * (*string*) –
- **CustomInstanceProfileArn** (*string*) – The ARN of an IAM profile to be used for the layer's EC2 instances. For more information about IAM ARNs, see [Using Identifiers](#).
- **CustomJson** (*string*) – A JSON-formatted string containing custom stack configuration and deployment attributes to be installed on the layer's instances. For more information, see [Using Custom JSON](#).
- **CustomSecurityGroupIds** (*list*) – An array containing the layer custom security group IDs.
 - (*string*) –
- **Packages** (*list*) – An array of `Package` objects that describes the layer packages.
 - (*string*) –
- **VolumeConfigurations** (*list*) – A `VolumeConfigurations` object that describes the layer's Amazon EBS volumes.
 - (*dict*) –

Describes an Amazon EBS volume configuration.

- * **MountPoint** (*string*) – **[REQUIRED]**

The volume mount point. For example “/dev/sdh”.

- * **RaidLevel** (*integer*) –

The volume [RAID level](#).

- * **NumberOfDisks** (*integer*) – **[REQUIRED]**

The number of disks in the volume.

- * **Size** (*integer*) – **[REQUIRED]**

The volume size.

- * **VolumeType** (*string*) –

The volume type:

- `standard - Magnetic`
- `io1 - Provisioned IOPS (SSD)`
- `gp2 - General Purpose (SSD)`

- * **Iops** (*integer*) –

For PIOPS volumes, the IOPS per disk.

- **EnableAutoHealing** (*boolean*) – Whether to disable auto healing for the layer.
- **AutoAssignElasticIps** (*boolean*) – Whether to automatically assign an [Elastic IP address](#) to the layer's instances. For more information, see [How to Edit a Layer](#).
- **AutoAssignPublicIps** (*boolean*) – For stacks that are running in a VPC, whether to automatically assign a public IP address to the layer's instances. For more information, see [How to Edit a Layer](#).
- **CustomRecipes** (*dict*) – A `LayerCustomRecipes` object that specifies the layer custom recipes.
 - **Setup** (*list*) –

An array of custom recipe names to be run following a `setup` event.

 - * (*string*) –

- **Configure** (*list*) –

An array of custom recipe names to be run following a `configure` event.

* (*string*) –

- **Deploy** (*list*) –

An array of custom recipe names to be run following a `deploy` event.

* (*string*) –

- **Undeploy** (*list*) –

An array of custom recipe names to be run following a `undeploy` event.

* (*string*) –

- **Shutdown** (*list*) –

An array of custom recipe names to be run following a `shutdown` event.

* (*string*) –

- **InstallUpdatesOnBoot** (*boolean*) – Whether to install operating system and package updates when the instance boots. The default value is `true`. To control when updates are installed, set this value to `false`. You must then update your instances manually by using `CreateDeployment` to run the `update_dependencies` stack command or by manually running `yum` (Amazon Linux) or `apt-get` (Ubuntu) on the instances.

Note: To ensure that your instances have the latest security updates, we strongly recommend using the default value of `true`.

- **UseEbsOptimizedInstances** (*boolean*) – Whether to use Amazon EBS-optimized instances.
- **LifecycleEventConfiguration** (*dict*) – A `LifeCycleEventConfiguration` object that you can use to configure the `Shutdown` event to specify an execution timeout and enable or disable Elastic Load Balancer connection draining.

- **Shutdown** (*dict*) –

A `ShutdownEventConfiguration` object that specifies the `Shutdown` event configuration.

* **ExecutionTimeout** (*integer*) –

The time, in seconds, that AWS OpsWorks will wait after triggering a `Shutdown` event before shutting down an instance.

* **DelayUntilElbConnectionsDrained** (*boolean*) –

Whether to enable Elastic Load Balancing connection draining. For more information, see [Connection Draining](#)

Return type `opsworks.Layer`

Returns Layer resource

delete()

Deletes a specified stack. You must first delete all instances, layers, and apps or deregister registered instances. For more information, see [Shut Down a Stack](#).

Required Permissions : To use this action, an IAM user must have a `Manage` permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [Managing User Permissions](#).

Request Syntax

```
response = stack.delete()
```

Returns None

load()

Calls `opsworks.Client.describe_stacks()` to update the attributes of the Stack resource

Request Syntax

```
stack.load()
```

Returns None

reload()

Calls `opsworks.Client.describe_stacks()` to update the attributes of the Stack resource

Request Syntax

```
stack.reload()
```

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Summary()

Creates a StackSummary resource.:

```
stack_summary = stack.Summary()
```

Return type `OpsWorks.StackSummary`

Returns A StackSummary resource

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

layers**all()**

Creates an iterable of all Layer resources in the collection.

Request Syntax

```
layer_iterator = stack.layers.all()
```

Return type `list(opsworks.Layer)`

Returns A list of Layer resources

filter(kwargs)**

Creates an iterable of all Layer resources in the collection filtered by kwargs passed to method.

Request Syntax

```
layer_iterator = stack.layers.filter(
    LayerIds=[
        'string',
    ]
)
```

Parameters **LayerIds** (*list*) – An array of layer IDs that specify the layers to be described. If you omit this parameter, `DescribeLayers` returns a description of every layer in the specified stack.

- (*string*) –

Return type `list(opsworks.Layer)`

Returns A list of Layer resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Layer resources in the collection.

Request Syntax

```
layer_iterator = stack.layers.limit(  
    count=123  
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(opsworks.Layer)`

Returns A list of Layer resources

page_size (***kwargs*)

Creates an iterable of all Layer resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
layer_iterator = stack.layers.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(opsworks.Layer)`

Returns A list of Layer resources

StackSummary

class `OpsWorks.StackSummary` (*stack_id*)

A resource representing an AWS OpsWorks StackSummary:

```
import boto3
```

```
opsworks = boto3.resource('opsworks')  
stack_summary = opsworks.StackSummary('stack_id')
```

Parameters **stack_id** (*string*) – The StackSummary’s `stack_id` identifier. This **must** be set.

These are the resource’s available identifiers:

- `stack_id`

These are the resource’s available attributes:

- `apps_count`
- `arn`
- `instances_count`
- `layers_count`
- `name`

These are the resource’s available sub-resources:

- `Stack()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

stack_id

(*string*) The StackSummary's stack_id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

apps_count

(*integer*)

The number of apps.

arn

(*string*)

The stack's ARN.

instances_count

(*dict*)

An `InstancesCount` object with the number of instances in each status.

layers_count

(*integer*)

The number of layers.

name

(*string*)

The stack name.

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Stack()

Creates a Stack resource.:

```
stack = stack_summary.Stack()
```

Return type `OpsWorks.Stack`

Returns A Stack resource

RDS

Table of Contents

- [RDS](#)
 - [Client](#)
 - [Paginators](#)
 - [Waiters](#)

Client

`class RDS.Client`

A low-level client representing Amazon Relational Database Service (RDS):

```
import boto3
```

```
client = boto3.client('rds')
```

These are the available methods:

- `add_source_identifier_to_subscription()`
- `add_tags_to_resource()`
- `apply_pending_maintenance_action()`
- `authorize_db_security_group_ingress()`
- `can_paginate()`
- `copy_db_cluster_snapshot()`
- `copy_db_parameter_group()`
- `copy_db_snapshot()`
- `copy_option_group()`
- `create_db_cluster()`
- `create_db_cluster_parameter_group()`
- `create_db_cluster_snapshot()`
- `create_db_instance()`
- `create_db_instance_read_replica()`
- `create_db_parameter_group()`
- `create_db_security_group()`
- `create_db_snapshot()`
- `create_db_subnet_group()`
- `create_event_subscription()`
- `create_option_group()`
- `delete_db_cluster()`
- `delete_db_cluster_parameter_group()`
- `delete_db_cluster_snapshot()`
- `delete_db_instance()`
- `delete_db_parameter_group()`
- `delete_db_security_group()`
- `delete_db_snapshot()`
- `delete_db_subnet_group()`
- `delete_event_subscription()`
- `delete_option_group()`
- `describe_account_attributes()`
- `describe_certificates()`
- `describe_db_cluster_parameter_groups()`
- `describe_db_cluster_parameters()`
- `describe_db_cluster_snapshots()`
- `describe_db_clusters()`
- `describe_db_engine_versions()`
- `describe_db_instances()`
- `describe_db_log_files()`
- `describe_db_parameter_groups()`
- `describe_db_parameters()`
- `describe_db_security_groups()`
- `describe_db_snapshots()`
- `describe_db_subnet_groups()`

- `describe_engine_default_cluster_parameters()`
- `describe_engine_default_parameters()`
- `describe_event_categories()`
- `describe_event_subscriptions()`
- `describe_events()`
- `describe_option_group_options()`
- `describe_option_groups()`
- `describe_orderable_db_instance_options()`
- `describe_pending_maintenance_actions()`
- `describe_reserved_db_instances()`
- `describe_reserved_db_instances_offerings()`
- `download_db_log_file_portion()`
- `failover_db_cluster()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_waiter()`
- `list_tags_for_resource()`
- `modify_db_cluster()`
- `modify_db_cluster_parameter_group()`
- `modify_db_instance()`
- `modify_db_parameter_group()`
- `modify_db_subnet_group()`
- `modify_event_subscription()`
- `modify_option_group()`
- `promote_read_replica()`
- `purchase_reserved_db_instances_offering()`
- `reboot_db_instance()`
- `remove_source_identifier_from_subscription()`
- `remove_tags_from_resource()`
- `reset_db_cluster_parameter_group()`
- `reset_db_parameter_group()`
- `restore_db_cluster_from_snapshot()`
- `restore_db_cluster_to_point_in_time()`
- `restore_db_instance_from_db_snapshot()`
- `restore_db_instance_to_point_in_time()`
- `revoke_db_security_group_ingress()`

`add_source_identifier_to_subscription(**kwargs)`
 Adds a source identifier to an existing RDS event notification subscription.

Request Syntax

```
response = client.add_source_identifier_to_subscription(
    SubscriptionName='string',
    SourceIdentifier='string'
)
```

Parameters

- **SubscriptionName** (*string*) – [REQUIRED]

The name of the RDS event notification subscription you want to add a source identifier to.

- **SourceIdentifier** (*string*) – [REQUIRED]

The identifier of the event source to be added. An identifier must begin with a letter and must contain only ASCII letters, digits, and hyphens; it cannot end with a hyphen or contain two consecutive hyphens.

Constraints:

- If the source type is a DB instance, then a `DBInstanceIdentifier` must be supplied.
- If the source type is a DB security group, a `DBSecurityGroupName` must be supplied.
- If the source type is a DB parameter group, a `DBParameterGroupName` must be supplied.
- If the source type is a DB snapshot, a `DBSnapshotIdentifier` must be supplied.

Return type dict

Returns

Response Syntax

```
{
  'EventSubscription': {
    'CustomerAwsId': 'string',
    'CustSubscriptionId': 'string',
    'SnsTopicArn': 'string',
    'Status': 'string',
    'SubscriptionCreationTime': 'string',
    'SourceType': 'string',
    'SourceIdsList': [
      'string',
    ],
    'EventCategoriesList': [
      'string',
    ],
    'Enabled': True|False
  }
}
```

Response Structure

- (*dict*) –

Contains the results of a successful invocation of the `DescribeEventSubscriptions` action.

- **EventSubscription** (*dict*) –

Contains the results of a successful invocation of the `DescribeEventSubscriptions` action.

- * **CustomerAwsId** (*string*) –

The AWS customer account associated with the RDS event notification subscription.

- * **CustSubscriptionId** (*string*) –

The RDS event notification subscription Id.

- * **SnsTopicArn** (*string*) –

The topic ARN of the RDS event notification subscription.

- * **Status** (*string*) –

The status of the RDS event notification subscription.

Constraints:

Can be one of the following: creating | modifying | deleting | active | no-permission | topic-not-exist

The status “no-permission” indicates that RDS no longer has permission to post to the SNS topic. The status “topic-not-exist” indicates that the topic was deleted after the subscription was created.

* **SubscriptionCreationTime** (*string*) –

The time the RDS event notification subscription was created.

* **SourceType** (*string*) –

The source type for the RDS event notification subscription.

* **SourceIdsList** (*list*) –

A list of source IDs for the RDS event notification subscription.
· (*string*) –

* **EventCategoriesList** (*list*) –

A list of event categories for the RDS event notification subscription.
· (*string*) –

* **Enabled** (*boolean*) –

A Boolean value indicating if the subscription is enabled. True indicates the subscription is enabled.

add_tags_to_resource (***kwargs*)

Adds metadata tags to an Amazon RDS resource. These tags can also be used with cost allocation reporting to track cost associated with Amazon RDS resources, or used in a Condition statement in an IAM policy for Amazon RDS.

For an overview on tagging Amazon RDS resources, see [Tagging Amazon RDS Resources](#).

Request Syntax

```
response = client.add_tags_to_resource(
    ResourceName='string',
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **ResourceName** (*string*) – [REQUIRED]

The Amazon RDS resource the tags will be added to. This value is an Amazon Resource Name (ARN). For information about creating an ARN, see [Constructing an RDS Amazon Resource Name \(ARN\)](#).

- **Tags** (*list*) – [REQUIRED]

The tags to be assigned to the Amazon RDS resource.

– (*dict*) –

Metadata assigned to an Amazon RDS resource consisting of a key-value pair.

* **Key** (*string*) –

A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with

“aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

* **Value** (*string*) –

A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

Returns None

apply_pending_maintenance_action (***kwargs*)

Applies a pending maintenance action to a resource (for example, to a DB instance).

Request Syntax

```
response = client.apply_pending_maintenance_action(
    ResourceIdentifier='string',
    ApplyAction='string',
    OptInType='string'
)
```

Parameters

• **ResourceIdentifier** (*string*) – **[REQUIRED]**

The RDS Amazon Resource Name (ARN) of the resource that the pending maintenance action applies to. For information about creating an ARN, see [Constructing an RDS Amazon Resource Name \(ARN\)](#).

• **ApplyAction** (*string*) – **[REQUIRED]**

The pending maintenance action to apply to this resource.

• **OptInType** (*string*) – **[REQUIRED]**

A value that specifies the type of opt-in request, or undoes an opt-in request. An opt-in request of type `immediate` cannot be undone.

Valid values:

- `immediate` - Apply the maintenance action immediately.
- `next-maintenance` - Apply the maintenance action during the next maintenance window for the resource.
- `undo-opt-in` - Cancel any existing `next-maintenance` opt-in requests.

Return type dict

Returns

Response Syntax

```
{
  'ResourcePendingMaintenanceActions': {
    'ResourceIdentifier': 'string',
    'PendingMaintenanceActionDetails': [
      {
        'Action': 'string',
        'AutoAppliedAfterDate': datetime(2015, 1, 1),
        'ForcedApplyDate': datetime(2015, 1, 1),
        'OptInStatus': 'string',
        'CurrentApplyDate': datetime(2015, 1, 1),
        'Description': 'string'
      ]
    ]
  }
}
```

```

    },
  ]
}

```

Response Structure

- (*dict*) –

Describes the pending maintenance actions for a resource.

- **ResourcePendingMaintenanceActions** (*dict*) –

Describes the pending maintenance actions for a resource.

- * **ResourceIdentifier** (*string*) –

The ARN of the resource that has pending maintenance actions.

- * **PendingMaintenanceActionDetails** (*list*) –

A list that provides details about the pending maintenance actions for the resource.

- (*dict*) –

Provides information about a pending maintenance action for a resource.

- **Action** (*string*) –

The type of pending maintenance action that is available for the resource.

- **AutoAppliedAfterDate** (*datetime*) –

The date of the maintenance window when the action will be applied. The maintenance action will be applied to the resource during its first maintenance window after this date. If this date is specified, any `next-maintenance` opt-in requests are ignored.

- **ForcedApplyDate** (*datetime*) –

The date when the maintenance action will be automatically applied. The maintenance action will be applied to the resource on this date regardless of the maintenance window for the resource. If this date is specified, any `immediate` opt-in requests are ignored.

- **OptInStatus** (*string*) –

Indicates the type of opt-in request that has been received for the resource.

- **CurrentApplyDate** (*datetime*) –

The effective date when the pending maintenance action will be applied to the resource. This date takes into account opt-in requests received from the `ApplyPendingMaintenanceAction` API, the `AutoAppliedAfterDate`, and the `ForcedApplyDate`. This value is blank if an opt-in request has not been received and nothing has been specified as `AutoAppliedAfterDate` or `ForcedApplyDate`.

- **Description** (*string*) –

A description providing more detail about the maintenance action.

authorize_db_security_group_ingress (**kwargs)

Enables ingress to a DBSecurityGroup using one of two forms of authorization. First, EC2 or VPC security groups can be added to the DBSecurityGroup if the application using the database is running on EC2 or VPC instances. Second, IP ranges are available if the application accessing your database is running on the Internet. Required parameters for this API are one of CIDR range, EC2SecurityGroupId for VPC, or (EC2SecurityGroupOwnerId and either EC2SecurityGroupName or EC2SecurityGroupId for non-VPC).

Note: You cannot authorize ingress from an EC2 security group in one region to an Amazon RDS DB instance in another. You cannot authorize ingress from a VPC security group in one VPC to an Amazon RDS DB instance in another.

For an overview of CIDR ranges, go to the [Wikipedia Tutorial](#) .

Request Syntax

```
response = client.authorize_db_security_group_ingress(
    DBSecurityGroupName='string',
    CIDRIP='string',
    EC2SecurityGroupName='string',
    EC2SecurityGroupId='string',
    EC2SecurityGroupOwnerId='string'
)
```

Parameters

- **DBSecurityGroupName** (*string*) – [REQUIRED]
The name of the DB security group to add authorization to.
- **CIDRIP** (*string*) – The IP range to authorize.
- **EC2SecurityGroupName** (*string*) – Name of the EC2 security group to authorize. For VPC DB security groups, EC2SecurityGroupId must be provided. Otherwise, EC2SecurityGroupOwnerId and either EC2SecurityGroupName or EC2SecurityGroupId must be provided.
- **EC2SecurityGroupId** (*string*) – Id of the EC2 security group to authorize. For VPC DB security groups, EC2SecurityGroupId must be provided. Otherwise, EC2SecurityGroupOwnerId and either EC2SecurityGroupName or EC2SecurityGroupId must be provided.
- **EC2SecurityGroupOwnerId** (*string*) – AWS account number of the owner of the EC2 security group specified in the EC2SecurityGroupName parameter. The AWS Access Key ID is not an acceptable value. For VPC DB security groups, EC2SecurityGroupId must be provided. Otherwise, EC2SecurityGroupOwnerId and either EC2SecurityGroupName or EC2SecurityGroupId must be provided.

Return type dict

Returns**Response Syntax**

```
{
  'DBSecurityGroup': {
    'OwnerId': 'string',
    'DBSecurityGroupName': 'string',
    'DBSecurityGroupDescription': 'string',
    'VpcId': 'string',
    'EC2SecurityGroups': [
      {
```

```

        'Status': 'string',
        'EC2SecurityGroupName': 'string',
        'EC2SecurityGroupId': 'string',
        'EC2SecurityGroupOwnerId': 'string'
    },
],
'IPRanges': [
    {
        'Status': 'string',
        'CIDRIP': 'string'
    },
]
}
}

```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the following actions:

- DescribeDBSecurityGroups
- AuthorizeDBSecurityGroupIngress
- CreateDBSecurityGroup
- RevokeDBSecurityGroupIngress

This data type is used as a response element in the DescribeDBSecurityGroups action.

- **DBSecurityGroup** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * DescribeDBSecurityGroups
- * AuthorizeDBSecurityGroupIngress
- * CreateDBSecurityGroup
- * RevokeDBSecurityGroupIngress

This data type is used as a response element in the DescribeDBSecurityGroups action.

- * **OwnerId** (*string*) –

Provides the AWS ID of the owner of a specific DB security group.

- * **DBSecurityGroupName** (*string*) –

Specifies the name of the DB security group.

- * **DBSecurityGroupDescription** (*string*) –

Provides the description of the DB security group.

- * **VpcId** (*string*) –

Provides the VpcId of the DB security group.

- * **EC2SecurityGroups** (*list*) –

Contains a list of EC2SecurityGroup elements.

- (*dict*) –

This data type is used as a response element in the following actions:

- AuthorizeDBSecurityGroupIngress
- DescribeDBSecurityGroups
- RevokeDBSecurityGroupIngress
- **Status** (*string*) –

Provides the status of the EC2 security group. Status can be

- “authorizing”, “authorized”, “revoking”, and “revoked”.
- **EC2SecurityGroupName** (*string*) –
Specifies the name of the EC2 security group.
- **EC2SecurityGroupId** (*string*) –
Specifies the id of the EC2 security group.
- **EC2SecurityGroupOwnerId** (*string*) –
Specifies the AWS ID of the owner of the EC2 security group specified in the EC2SecurityGroupName field.
- * **IPRanges** (*list*) –
Contains a list of IPRange elements.
 - (*dict*) –
This data type is used as a response element in the DescribeDBSecurityGroups action.
 - **Status** (*string*) –
Specifies the status of the IP range. Status can be “authorizing”, “authorized”, “revoking”, and “revoked”.
 - **CIDRIP** (*string*) –
Specifies the IP range.

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you’d normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

copy_db_cluster_snapshot (***kwargs*)

Creates a snapshot of a DB cluster. For more information on Amazon Aurora, see [Aurora on Amazon RDS](#) in the *Amazon RDS User Guide*.

Request Syntax

```
response = client.copy_db_cluster_snapshot(
    SourceDBClusterSnapshotIdentifier='string',
    TargetDBClusterSnapshotIdentifier='string',
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **SourceDBClusterSnapshotIdentifier** (*string*) – [REQUIRED]

The identifier of the DB cluster snapshot to copy. This parameter is not case-sensitive.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens.
- First character must be a letter.

- Cannot end with a hyphen or contain two consecutive hyphens.

Example: `my-cluster-snapshot1`

- **TargetDBClusterSnapshotIdentifier** (*string*) – [REQUIRED]

The identifier of the new DB cluster snapshot to create from the source DB cluster snapshot. This parameter is not case-sensitive.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.

Example: `my-cluster-snapshot2`

- **Tags** (*list*) – A list of tags.
- (*dict*) –

Metadata assigned to an Amazon RDS resource consisting of a key-value pair.

- * **Key** (*string*) –

A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “`^([\p{L}\p{Z}\p{N}_./=+\\-]*)$`”).

- * **Value** (*string*) –

A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “`^([\p{L}\p{Z}\p{N}_./=+\\-]*)$`”).

Return type dict

Returns

Response Syntax

```
{
  'DBClusterSnapshot': {
    'AvailabilityZones': [
      'string',
    ],
    'DBClusterSnapshotIdentifier': 'string',
    'DBClusterIdentifier': 'string',
    'SnapshotCreateTime': datetime(2015, 1, 1),
    'Engine': 'string',
    'AllocatedStorage': 123,
    'Status': 'string',
    'Port': 123,
    'VpcId': 'string',
    'ClusterCreateTime': datetime(2015, 1, 1),
    'MasterUsername': 'string',
    'EngineVersion': 'string',
    'LicenseModel': 'string',
    'SnapshotType': 'string',
    'PercentProgress': 123
  }
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the following actions:

- CreateDBClusterSnapshot
- DeleteDBClusterSnapshot

This data type is used as a response element in the DescribeDBClusterSnapshots action.

- **DBClusterSnapshot** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * CreateDBClusterSnapshot
- * DeleteDBClusterSnapshot

This data type is used as a response element in the DescribeDBClusterSnapshots action.

- * **AvailabilityZones** (*list*) –

Provides the list of EC2 Availability Zones that instances in the DB cluster snapshot can be restored in.

· (*string*) –

- * **DBClusterSnapshotIdentifier** (*string*) –

Specifies the identifier for the DB cluster snapshot.

- * **DBClusterIdentifier** (*string*) –

Specifies the DB cluster identifier of the DB cluster that this DB cluster snapshot was created from.

- * **SnapshotCreateTime** (*datetime*) –

Provides the time when the snapshot was taken, in Universal Coordinated Time (UTC).

- * **Engine** (*string*) –

Specifies the name of the database engine.

- * **AllocatedStorage** (*integer*) –

Specifies the allocated storage size in gigabytes (GB).

- * **Status** (*string*) –

Specifies the status of this DB cluster snapshot.

- * **Port** (*integer*) –

Specifies the port that the DB cluster was listening on at the time of the snapshot.

- * **VpcId** (*string*) –

Provides the VPC ID associated with the DB cluster snapshot.

- * **ClusterCreateTime** (*datetime*) –

Specifies the time when the DB cluster was created, in Universal Coordinated Time (UTC).

- * **MasterUsername** (*string*) –

Provides the master username for the DB cluster snapshot.

- * **EngineVersion** (*string*) –

Provides the version of the database engine for this DB cluster snapshot.

- * **LicenseModel** (*string*) –

Provides the license model information for this DB cluster snapshot.

* **SnapshotType** (*string*) –

Provides the type of the DB cluster snapshot.

* **PercentProgress** (*integer*) –

Specifies the percentage of the estimated data that has been transferred.

copy_db_parameter_group (**kwargs)

Copies the specified DB parameter group.

Request Syntax

```
response = client.copy_db_parameter_group(
    SourceDBParameterGroupIdentifier='string',
    TargetDBParameterGroupIdentifier='string',
    TargetDBParameterGroupDescription='string',
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

• **SourceDBParameterGroupIdentifier** (*string*) – [REQUIRED]

The identifier or ARN for the source DB parameter group. For information about creating an ARN, see [Constructing an RDS Amazon Resource Name \(ARN\)](#).

Constraints:

- Must specify a valid DB parameter group.
- If the source DB parameter group is in the same region as the copy, specify a valid DB parameter group identifier, for example `my-db-param-group`, or a valid ARN.
- If the source DB parameter group is in a different region than the copy, specify a valid DB parameter group ARN, for example `arn:aws:rds:us-west-2:123456789012:pg:special-parameters`.

• **TargetDBParameterGroupIdentifier** (*string*) – [REQUIRED]

The identifier for the copied DB parameter group.

Constraints:

- Cannot be null, empty, or blank
- Must contain from 1 to 255 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Example: `my-db-parameter-group`

• **TargetDBParameterGroupDescription** (*string*) – [REQUIRED]

A description for the copied DB parameter group.

• **Tags** (*list*) – A list of tags.

– (*dict*) –

Metadata assigned to an Amazon RDS resource consisting of a key-value pair.

* **Key** (*string*) –

A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

* **Value** (*string*) –

A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

Return type dict

Returns

Response Syntax

```
{
  'DBParameterGroup': {
    'DBParameterGroupName': 'string',
    'DBParameterGroupFamily': 'string',
    'Description': 'string'
  }
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the CreateDBParameterGroup action.

This data type is used as a request parameter in the DeleteDBParameterGroup action, and as a response element in the DescribeDBParameterGroups action.

- **DBParameterGroup** (*dict*) –

Contains the result of a successful invocation of the CreateDBParameterGroup action.

This data type is used as a request parameter in the DeleteDBParameterGroup action, and as a response element in the DescribeDBParameterGroups action.

- * **DBParameterGroupName** (*string*) –

Provides the name of the DB parameter group.

- * **DBParameterGroupFamily** (*string*) –

Provides the name of the DB parameter group family that this DB parameter group is compatible with.

- * **Description** (*string*) –

Provides the customer-specified description for this DB parameter group.

copy_db_snapshot (***kwargs*)

Copies the specified DBSnapshot. The source DBSnapshot must be in the “available” state.

Request Syntax

```
response = client.copy_db_snapshot(
    SourceDBSnapshotIdentifier='string',
```

```

TargetDBSnapshotIdentifier='string',
Tags=[
    {
        'Key': 'string',
        'Value': 'string'
    },
],
CopyTags=True|False
)

```

Parameters

- **SourceDBSnapshotIdentifier** (*string*) – [REQUIRED]

The identifier for the source DB snapshot.

Constraints:

- Must specify a valid system snapshot in the “available” state.
- If the source snapshot is in the same region as the copy, specify a valid DB snapshot identifier.
- If the source snapshot is in a different region than the copy, specify a valid DB snapshot ARN. For more information, go to [Copying a DB Snapshot](#).

Example: rds:mydb-2012-04-02-00-01

Example: arn:aws:rds:rr-regn-1:123456789012:snapshot:mysql-instance1-sna

- **TargetDBSnapshotIdentifier** (*string*) – [REQUIRED]

The identifier for the copied snapshot.

Constraints:

- Cannot be null, empty, or blank
- Must contain from 1 to 255 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Example: my-db-snapshot

- **Tags** (*list*) – A list of tags.

– (*dict*) –

Metadata assigned to an Amazon RDS resource consisting of a key-value pair.

* **Key** (*string*) –

A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

* **Value** (*string*) –

A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

- **CopyTags** (*boolean*) – This property is not currently implemented.

Return type dict

Returns

Response Syntax

```
{
  'DBSnapshot': {
    'DBSnapshotIdentifier': 'string',
    'DBInstanceIdentifier': 'string',
    'SnapshotCreateTime': datetime(2015, 1, 1),
    'Engine': 'string',
    'AllocatedStorage': 123,
    'Status': 'string',
    'Port': 123,
    'AvailabilityZone': 'string',
    'VpcId': 'string',
    'InstanceCreateTime': datetime(2015, 1, 1),
    'MasterUsername': 'string',
    'EngineVersion': 'string',
    'LicenseModel': 'string',
    'SnapshotType': 'string',
    'Iops': 123,
    'OptionGroupName': 'string',
    'PercentProgress': 123,
    'SourceRegion': 'string',
    'SourceDBSnapshotIdentifier': 'string',
    'StorageType': 'string',
    'TdeCredentialArn': 'string',
    'Encrypted': True|False,
    'KmsKeyId': 'string'
  }
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the following actions:

- CreateDBSnapshot
- DeleteDBSnapshot

This data type is used as a response element in the DescribeDBSnapshots action.

- **DBSnapshot** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * CreateDBSnapshot
- * DeleteDBSnapshot

This data type is used as a response element in the DescribeDBSnapshots action.

- * **DBSnapshotIdentifier** (*string*) –

Specifies the identifier for the DB snapshot.

- * **DBInstanceIdentifier** (*string*) –

Specifies the DB instance identifier of the DB instance this DB snapshot was created from.

- * **SnapshotCreateTime** (*datetime*) –

Provides the time when the snapshot was taken, in Universal Coordinated Time (UTC).

- * **Engine** (*string*) –

Specifies the name of the database engine.

- * **AllocatedStorage** (*integer*) –

Specifies the allocated storage size in gigabytes (GB).

- * **Status** (*string*) –
Specifies the status of this DB snapshot.
- * **Port** (*integer*) –
Specifies the port that the database engine was listening on at the time of the snapshot.
- * **AvailabilityZone** (*string*) –
Specifies the name of the Availability Zone the DB instance was located in at the time of the DB snapshot.
- * **VpcId** (*string*) –
Provides the VPC ID associated with the DB snapshot.
- * **InstanceCreateTime** (*datetime*) –
Specifies the time when the snapshot was taken, in Universal Coordinated Time (UTC).
- * **MasterUsername** (*string*) –
Provides the master username for the DB snapshot.
- * **EngineVersion** (*string*) –
Specifies the version of the database engine.
- * **LicenseModel** (*string*) –
License model information for the restored DB instance.
- * **SnapshotType** (*string*) –
Provides the type of the DB snapshot.
- * **Iops** (*integer*) –
Specifies the Provisioned IOPS (I/O operations per second) value of the DB instance at the time of the snapshot.
- * **OptionGroupName** (*string*) –
Provides the option group name for the DB snapshot.
- * **PercentProgress** (*integer*) –
The percentage of the estimated data that has been transferred.
- * **SourceRegion** (*string*) –
The region that the DB snapshot was created in or copied from.
- * **SourceDBSnapshotIdentifier** (*string*) –
The DB snapshot Arn that the DB snapshot was copied from. It only has value in case of cross customer or cross region copy.
- * **StorageType** (*string*) –
Specifies the storage type associated with DB Snapshot.
- * **TdeCredentialArn** (*string*) –
The ARN from the Key Store with which to associate the instance for TDE encryption.
- * **Encrypted** (*boolean*) –
Specifies whether the DB snapshot is encrypted.
- * **KmsKeyId** (*string*) –
If `Encrypted` is true, the KMS key identifier for the encrypted DB snapshot.

copy_option_group (**kwargs)
Copies the specified option group.

Request Syntax

```
response = client.copy_option_group(
    SourceOptionGroupIdentifier='string',
    TargetOptionGroupIdentifier='string',
    TargetOptionGroupDescription='string',
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **SourceOptionGroupIdentifier** (*string*) – [REQUIRED]

The identifier or ARN for the source option group. For information about creating an ARN, see [Constructing an RDS Amazon Resource Name \(ARN\)](#).

Constraints:

- Must specify a valid option group.
- If the source option group is in the same region as the copy, specify a valid option group identifier, for example `my-option-group`, or a valid ARN.
- If the source option group is in a different region than the copy, specify a valid option group ARN, for example `arn:aws:rds:us-west-2:123456789012:og:special-options`.

- **TargetOptionGroupIdentifier** (*string*) – [REQUIRED]

The identifier for the copied option group.

Constraints:

- Cannot be null, empty, or blank
- Must contain from 1 to 255 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Example: `my-option-group`

- **TargetOptionGroupDescription** (*string*) – [REQUIRED]

The description for the copied option group.

- **Tags** (*list*) – A list of tags.
 - (*dict*) –

Metadata assigned to an Amazon RDS resource consisting of a key-value pair.

- * **Key** (*string*) –

A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “`^([\\p{L}\\p{Z}\\p{N}]_./=+\\-]*)$`”).

- * **Value** (*string*) –

A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

Return type dict

Returns

Response Syntax

```
{
  'OptionGroup': {
    'OptionGroupName': 'string',
    'OptionGroupDescription': 'string',
    'EngineName': 'string',
    'MajorEngineVersion': 'string',
    'Options': [
      {
        'OptionName': 'string',
        'OptionDescription': 'string',
        'Persistent': True|False,
        'Permanent': True|False,
        'Port': 123,
        'OptionSettings': [
          {
            'Name': 'string',
            'Value': 'string',
            'DefaultValue': 'string',
            'Description': 'string',
            'ApplyType': 'string',
            'DataType': 'string',
            'AllowedValues': 'string',
            'IsModifiable': True|False,
            'IsCollection': True|False
          },
        ],
        'DBSecurityGroupMemberships': [
          {
            'DBSecurityGroupName': 'string',
            'Status': 'string'
          },
        ],
        'VpcSecurityGroupMemberships': [
          {
            'VpcSecurityGroupId': 'string',
            'Status': 'string'
          },
        ],
      },
    ],
    'AllowsVpcAndNonVpcInstanceMemberships': True|False,
    'VpcId': 'string'
  }
}
```

Response Structure

- (dict) –
- OptionGroup (dict) –

- * **OptionGroupName** (*string*) –
Specifies the name of the option group.
- * **OptionGroupDescription** (*string*) –
Provides a description of the option group.
- * **EngineName** (*string*) –
Indicates the name of the engine that this option group can be applied to.
- * **MajorEngineVersion** (*string*) –
Indicates the major engine version associated with this option group.
- * **Options** (*list*) –
Indicates what options are available in the option group.
 - (*dict*) –
Option details.
 - **OptionName** (*string*) –
The name of the option.
 - **OptionDescription** (*string*) –
The description of the option.
 - **Persistent** (*boolean*) –
Indicate if this option is persistent.
 - **Permanent** (*boolean*) –
Indicate if this option is permanent.
 - **Port** (*integer*) –
If required, the port configured for this option to use.
 - **OptionSettings** (*list*) –
The option settings for this option.
 - (*dict*) –
Option settings are the actual settings being applied or configured for that option. It is used when you modify an option group or describe option groups. For example, the `NATIVE_NETWORK_ENCRYPTION` option has a setting called `SQLNET.ENCRYPTION_SERVER` that can have several different values.
 - **Name** (*string*) –
The name of the option that has settings that you can set.
 - **Value** (*string*) –
The current value of the option setting.
 - **DefaultValue** (*string*) –
The default value of the option setting.
 - **Description** (*string*) –
The description of the option setting.
 - **ApplyType** (*string*) –
The DB engine specific parameter type.
 - **DataType** (*string*) –
The data type of the option setting.

- **AllowedValues** (*string*) –
The allowed values of the option setting.
- **IsModifiable** (*boolean*) –
A Boolean value that, when true, indicates the option setting can be modified from the default.
- **IsCollection** (*boolean*) –
Indicates if the option setting is part of a collection.
- **DBSecurityGroupMemberships** (*list*) –
If the option requires access to a port, then this DB security group allows access to the port.
- (*dict*) –
This data type is used as a response element in the following actions:
 - **ModifyDBInstance**
 - **RebootDBInstance**
 - **RestoreDBInstanceFromDBSnapshot**
 - **RestoreDBInstanceToPointInTime**
- **DBSecurityGroupName** (*string*) –
The name of the DB security group.
- **Status** (*string*) –
The status of the DB security group.
- **VpcSecurityGroupMemberships** (*list*) –
If the option requires access to a port, then this VPC security group allows access to the port.
- (*dict*) –
This data type is used as a response element for queries on VPC security group membership.
- **VpcSecurityGroupId** (*string*) –
The name of the VPC security group.
- **Status** (*string*) –
The status of the VPC security group.
- * **AllowsVpcAndNonVpcInstanceMemberships** (*boolean*) –
Indicates whether this option group can be applied to both VPC and non-VPC instances. The value `true` indicates the option group can be applied to both VPC and non-VPC instances.
- * **VpcId** (*string*) –
If **AllowsVpcAndNonVpcInstanceMemberships** is `false`, this field is blank. If **AllowsVpcAndNonVpcInstanceMemberships** is `true` and this field is blank, then this option group can be applied to both VPC and non-VPC instances. If this field contains a value, then this option group can only be applied to instances that are in the VPC indicated by this field.

create_db_cluster (***kwargs*)

Creates a new Amazon Aurora DB cluster. For more information on Amazon Aurora, see [Aurora on Amazon RDS](#) in the *Amazon RDS User Guide*.

Request Syntax

```
response = client.create_db_cluster(  
    AvailabilityZones=[  
        'string',  
    ],  
    BackupRetentionPeriod=123,  
    CharacterSetName='string',  
    DatabaseName='string',  
    DBClusterIdentifier='string',  
    DBClusterParameterGroupName='string',  
    VpcSecurityGroupIds=[  
        'string',  
    ],  
    DBSubnetGroupName='string',  
    Engine='string',  
    EngineVersion='string',  
    Port=123,  
    MasterUsername='string',  
    MasterUserPassword='string',  
    OptionGroupName='string',  
    PreferredBackupWindow='string',  
    PreferredMaintenanceWindow='string',  
    Tags=[  
        {  
            'Key': 'string',  
            'Value': 'string'  
        },  
    ],  
)
```

Parameters

- **AvailabilityZones** (*list*) – A list of EC2 Availability Zones that instances in the DB cluster can be created in. For information on regions and Availability Zones, see [Regions and Availability Zones](#).
 - (*string*) –
- **BackupRetentionPeriod** (*integer*) – The number of days for which automated backups are retained. Setting this parameter to a positive number enables backups. Setting this parameter to 0 disables automated backups.

Default: 1

Constraints:

- Must be a value from 0 to 35
- **CharacterSetName** (*string*) – A value that indicates that the DB cluster should be associated with the specified CharacterSet.
- **DatabaseName** (*string*) – The name for your database of up to 8 alpha-numeric characters. If you do not provide a name, Amazon RDS will not create a database in the DB cluster you are creating.
- **DBClusterIdentifier** (*string*) – The DB cluster identifier. This parameter is stored as a lowercase string.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.

Example: my-cluster1

- **DBClusterParameterGroupName** (*string*) – The name of the DB cluster parameter group to associate with this DB cluster. If this argument is omitted,

default.aurora5.6 for the specified engine will be used.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens
- **VpcSecurityGroupIds** (*list*) – A list of EC2 VPC security groups to associate with this DB cluster.
 - (*string*) –
- **DBSubnetGroupName** (*string*) – A DB subnet group to associate with this DB cluster.
- **Engine** (*string*) – The name of the database engine to be used for this DB cluster.

Valid Values: MySQL

- **EngineVersion** (*string*) – The version number of the database engine to use.

Aurora

Example: 5.6.0

- **Port** (*integer*) – The port number on which the instances in the DB cluster accept connections.

Default: 3306

- **MasterUsername** (*string*) – The name of the master user for the client DB cluster.

Constraints:

- Must be 1 to 16 alphanumeric characters.
- First character must be a letter.
- Cannot be a reserved word for the chosen database engine.
- **MasterUserPassword** (*string*) – The password for the master database user. This password can contain any printable ASCII character except “/”, “””, or “@”.

Constraints: Must contain from 8 to 41 characters.

- **OptionGroupName** (*string*) – A value that indicates that the DB cluster should be associated with the specified option group.

Permanent options cannot be removed from an option group. The option group cannot be removed from a DB cluster once it is associated with a DB cluster.

- **PreferredBackupWindow** (*string*) – The daily time range during which automated backups are created if automated backups are enabled using the BackupRetentionPeriod parameter.

Default: A 30-minute window selected at random from an 8-hour block of time per region. To see the time blocks available, see [Adjusting the Preferred Maintenance Window](#) in the *Amazon RDS User Guide*.

Constraints:

- Must be in the format hh24:mi-hh24:mi .
- Times should be in Universal Coordinated Time (UTC).
- Must not conflict with the preferred maintenance window.
- Must be at least 30 minutes.
- **PreferredMaintenanceWindow** (*string*) – The weekly time range during which system maintenance can occur, in Universal Coordinated Time (UTC).

Format: ddd:hh24:mi-ddd:hh24:mi

Default: A 30-minute window selected at random from an 8-hour block of time per region, occurring on a random day of the week. To see the time blocks

available, see [Adjusting the Preferred Maintenance Window](#) in the *Amazon RDS User Guide*.

Valid Days: Mon, Tue, Wed, Thu, Fri, Sat, Sun

Constraints: Minimum 30-minute window.

- **Tags** (*list*) – A list of tags.
 - (*dict*) –

Metadata assigned to an Amazon RDS resource consisting of a key-value pair.

- * **Key** (*string*) –

A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

- * **Value** (*string*) –

A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

Return type dict

Returns

Response Syntax

```
{
  'DBCluster': {
    'AllocatedStorage': 123,
    'AvailabilityZones': [
      'string',
    ],
    'BackupRetentionPeriod': 123,
    'CharacterSetName': 'string',
    'DatabaseName': 'string',
    'DBClusterIdentifier': 'string',
    'DBClusterParameterGroup': 'string',
    'DBSubnetGroup': 'string',
    'Status': 'string',
    'PercentProgress': 'string',
    'EarliestRestorableTime': datetime(2015, 1, 1),
    'Endpoint': 'string',
    'Engine': 'string',
    'EngineVersion': 'string',
    'LatestRestorableTime': datetime(2015, 1, 1),
    'Port': 123,
    'MasterUsername': 'string',
    'DBClusterOptionGroupMemberships': [
      {
        'DBClusterOptionGroupName': 'string',
        'Status': 'string'
      },
    ],
    'PreferredBackupWindow': 'string',
    'PreferredMaintenanceWindow': 'string',
  },
}
```

```

'DBClusterMembers': [
    {
        'DBInstanceIdentifier': 'string',
        'IsClusterWriter': True|False,
        'DBClusterParameterGroupStatus': 'string'
    },
],
'VpcSecurityGroups': [
    {
        'VpcSecurityGroupId': 'string',
        'Status': 'string'
    },
]
}
}

```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the following actions:

- CreateDBCluster
- DeleteDBCluster
- FailoverDBCluster
- ModifyDBCluster
- RestoreDBClusterFromSnapshot

This data type is used as a response element in the DescribeDBClusters action.

- **DBCluster** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * CreateDBCluster
- * DeleteDBCluster
- * FailoverDBCluster
- * ModifyDBCluster
- * RestoreDBClusterFromSnapshot

This data type is used as a response element in the DescribeDBClusters action.

- * **AllocatedStorage** (*integer*) –

Specifies the allocated storage size in gigabytes (GB).

- * **AvailabilityZones** (*list*) –

Provides the list of EC2 Availability Zones that instances in the DB cluster can be created in.

· (*string*) –

- * **BackupRetentionPeriod** (*integer*) –

Specifies the number of days for which automatic DB snapshots are retained.

- * **CharacterSetName** (*string*) –

If present, specifies the name of the character set that this cluster is associated with.

- * **DatabaseName** (*string*) –

Contains the name of the initial database of this DB cluster that was provided at create time, if one was specified when the DB cluster was created. This same name is returned for the life of the DB cluster.

- * **DBClusterIdentifier** (*string*) –
Contains a user-supplied DB cluster identifier. This identifier is the unique key that identifies a DB cluster.
- * **DBClusterParameterGroup** (*string*) –
Specifies the name of the DB cluster parameter group for the DB cluster.
- * **DBSubnetGroup** (*string*) –
Specifies information on the subnet group associated with the DB cluster, including the name, description, and subnets in the subnet group.
- * **Status** (*string*) –
Specifies the current state of this DB cluster.
- * **PercentProgress** (*string*) –
Specifies the progress of the operation as a percentage.
- * **EarliestRestorableTime** (*datetime*) –
Specifies the earliest time to which a database can be restored with point-in-time restore.
- * **Endpoint** (*string*) –
Specifies the connection endpoint for the primary instance of the DB cluster.
- * **Engine** (*string*) –
Provides the name of the database engine to be used for this DB cluster.
- * **EngineVersion** (*string*) –
Indicates the database engine version.
- * **LatestRestorableTime** (*datetime*) –
Specifies the latest time to which a database can be restored with point-in-time restore.
- * **Port** (*integer*) –
Specifies the port that the database engine is listening on.
- * **MasterUsername** (*string*) –
Contains the master username for the DB cluster.
- * **DBClusterOptionGroupMemberships** (*list*) –
Provides the list of option group memberships for this DB cluster.
 - (*dict*) –
Contains status information for a DB cluster option group.
 - **DBClusterOptionGroupName** (*string*) –
Specifies the name of the DB cluster option group.
 - **Status** (*string*) –
Specifies the status of the DB cluster option group.
- * **PreferredBackupWindow** (*string*) –
Specifies the daily time range during which automated backups are created if automated backups are enabled, as determined by the `BackupRetentionPeriod`.

* **PreferredMaintenanceWindow** (*string*) –

Specifies the weekly time range during which system maintenance can occur, in Universal Coordinated Time (UTC).

* **DBClusterMembers** (*list*) –

Provides the list of instances that make up the DB cluster.

· (*dict*) –

Contains information about an instance that is part of a DB cluster.

· **DBInstanceIdentifier** (*string*) –

Specifies the instance identifier for this member of the DB cluster.

· **IsClusterWriter** (*boolean*) –

Value that is `true` if the cluster member is the primary instance for the DB cluster and `false` otherwise.

· **DBClusterParameterGroupStatus** (*string*) –

Specifies the status of the DB cluster parameter group for this member of the DB cluster.

* **VpcSecurityGroups** (*list*) –

Provides a list of VPC security groups that the DB cluster belongs to.

· (*dict*) –

This data type is used as a response element for queries on VPC security group membership.

· **VpcSecurityGroupId** (*string*) –

The name of the VPC security group.

· **Status** (*string*) –

The status of the VPC security group.

create_db_cluster_parameter_group (**kwargs)

Creates a new DB cluster parameter group.

Parameters in a DB cluster parameter group apply to all of the instances in a DB cluster.

A DB cluster parameter group is initially created with the default parameters for the database engine used by instances in the DB cluster. To provide custom values for any of the parameters, you must modify the group after creating it using `ModifyDBClusterParameterGroup`. Once you've created a DB cluster parameter group, you need to associate it with your DB cluster using `ModifyDBCluster`. When you associate a new DB cluster parameter group with a running DB cluster, you need to reboot the DB instances in the DB cluster without failover for the new DB cluster parameter group and associated settings to take effect.

Warning: After you create a DB cluster parameter group, you should wait at least 5 minutes before creating your first DB cluster that uses that DB cluster parameter group as the default parameter group. This allows Amazon RDS to fully complete the create action before the DB cluster parameter group is used as the default for a new DB cluster. This is especially important for parameters that are critical when creating the default database for a DB cluster, such as the character set for the default database defined by the `character_set_database` parameter. You can use the *Parameter Groups* option of the [Amazon RDS console](#) or the `DescribeDBClusterParameters` command to verify that your DB cluster parameter group has been created or modified.

For more information on Amazon Aurora, see [Aurora on Amazon RDS](#) in the *Amazon RDS User Guide*.

Request Syntax

```
response = client.create_db_cluster_parameter_group(
    DBClusterParameterGroupName='string',
    DBParameterGroupFamily='string',
    Description='string',
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **DBClusterParameterGroupName** (*string*) – [REQUIRED]

The name of the DB cluster parameter group.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Note: This value is stored as a lowercase string.

- **DBParameterGroupFamily** (*string*) – [REQUIRED]

The DB cluster parameter group family name. A DB cluster parameter group can be associated with one and only one DB cluster parameter group family, and can be applied only to a DB cluster running a database engine and engine version compatible with that DB cluster parameter group family.

- **Description** (*string*) – [REQUIRED]

The description for the DB cluster parameter group.

- **Tags** (*list*) – A list of tags.
 - (*dict*) –

Metadata assigned to an Amazon RDS resource consisting of a key-value pair.

* **Key** (*string*) –

A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

* **Value** (*string*) –

A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

Return type dict

Returns

Response Syntax


```
{
    'DBClusterParameterGroup': {
        'DBClusterParameterGroupName': 'string',
        'DBParameterGroupFamily': 'string',
        'Description': 'string'
    }
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the CreateDBClusterParameterGroup action.

This data type is used as a request parameter in the DeleteDBClusterParameterGroup action, and as a response element in the DescribeDBClusterParameterGroups action.

- **DBClusterParameterGroup** (*dict*) –

Contains the result of a successful invocation of the CreateDBClusterParameterGroup action.

This data type is used as a request parameter in the DeleteDBClusterParameterGroup action, and as a response element in the DescribeDBClusterParameterGroups action.

- * **DBClusterParameterGroupName** (*string*) –

Provides the name of the DB cluster parameter group.

- * **DBParameterGroupFamily** (*string*) –

Provides the name of the DB parameter group family that this DB cluster parameter group is compatible with.

- * **Description** (*string*) –

Provides the customer-specified description for this DB cluster parameter group.

create_db_cluster_snapshot (***kwargs*)

Creates a snapshot of a DB cluster. For more information on Amazon Aurora, see [Aurora on Amazon RDS](#) in the *Amazon RDS User Guide*.

Request Syntax

```
response = client.create_db_cluster_snapshot(
    DBClusterSnapshotIdentifier='string',
    DBClusterIdentifier='string',
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
)
```

Parameters

- **DBClusterSnapshotIdentifier** (*string*) – **[REQUIRED]**

The identifier of the DB cluster snapshot. This parameter is stored as a lowercase string.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.

Example: `my-cluster1-snapshot1`

- **DBClusterIdentifier** (*string*) – **[REQUIRED]**

The identifier of the DB cluster to create a snapshot for. This parameter is not case-sensitive.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.

Example: `my-cluster1`

- **Tags** (*list*) – The tags to be assigned to the DB cluster snapshot.
 - (*dict*) –

Metadata assigned to an Amazon RDS resource consisting of a key-value pair.

- * **Key** (*string*) –

A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

- * **Value** (*string*) –

A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

Return type dict

Returns

Response Syntax

```
{
  'DBClusterSnapshot': {
    'AvailabilityZones': [
      'string',
    ],
    'DBClusterSnapshotIdentifier': 'string',
    'DBClusterIdentifier': 'string',
    'SnapshotCreateTime': datetime(2015, 1, 1),
    'Engine': 'string',
    'AllocatedStorage': 123,
    'Status': 'string',
    'Port': 123,
    'VpcId': 'string',
    'ClusterCreateTime': datetime(2015, 1, 1),
    'MasterUsername': 'string',
    'EngineVersion': 'string',
    'LicenseModel': 'string',
    'SnapshotType': 'string',
    'PercentProgress': 123
  }
}
```

```

    }
}

```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the following actions:

- CreateDBClusterSnapshot
- DeleteDBClusterSnapshot

This data type is used as a response element in the DescribeDBClusterSnapshots action.

- **DBClusterSnapshot** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * CreateDBClusterSnapshot
- * DeleteDBClusterSnapshot

This data type is used as a response element in the DescribeDBClusterSnapshots action.

- * **AvailabilityZones** (*list*) –

Provides the list of EC2 Availability Zones that instances in the DB cluster snapshot can be restored in.

· (*string*) –

- * **DBClusterSnapshotIdentifier** (*string*) –

Specifies the identifier for the DB cluster snapshot.

- * **DBClusterIdentifier** (*string*) –

Specifies the DB cluster identifier of the DB cluster that this DB cluster snapshot was created from.

- * **SnapshotCreateTime** (*datetime*) –

Provides the time when the snapshot was taken, in Universal Coordinated Time (UTC).

- * **Engine** (*string*) –

Specifies the name of the database engine.

- * **AllocatedStorage** (*integer*) –

Specifies the allocated storage size in gigabytes (GB).

- * **Status** (*string*) –

Specifies the status of this DB cluster snapshot.

- * **Port** (*integer*) –

Specifies the port that the DB cluster was listening on at the time of the snapshot.

- * **VpcId** (*string*) –

Provides the VPC ID associated with the DB cluster snapshot.

- * **ClusterCreateTime** (*datetime*) –

Specifies the time when the DB cluster was created, in Universal Coordinated Time (UTC).

- * **MasterUsername** (*string*) –

Provides the master username for the DB cluster snapshot.

- * **EngineVersion** (*string*) –

Provides the version of the database engine for this DB cluster snapshot.

* **LicenseModel** (*string*) –

Provides the license model information for this DB cluster snapshot.

* **SnapshotType** (*string*) –

Provides the type of the DB cluster snapshot.

* **PercentProgress** (*integer*) –

Specifies the percentage of the estimated data that has been transferred.

create_db_instance (***kwargs*)

Creates a new DB instance.

Request Syntax

```
response = client.create_db_instance(
    DBName='string',
    DBInstanceIdentifier='string',
    AllocatedStorage=123,
    DBInstanceClass='string',
    Engine='string',
    MasterUsername='string',
    MasterUserPassword='string',
    DBSecurityGroups=[
        'string',
    ],
    VpcSecurityGroupIds=[
        'string',
    ],
    AvailabilityZone='string',
    DBSubnetGroupName='string',
    PreferredMaintenanceWindow='string',
    DBParameterGroupName='string',
    BackupRetentionPeriod=123,
    PreferredBackupWindow='string',
    Port=123,
    MultiAZ=True|False,
    EngineVersion='string',
    AutoMinorVersionUpgrade=True|False,
    LicenseModel='string',
    Iops=123,
    OptionGroupName='string',
    CharacterSetName='string',
    PubliclyAccessible=True|False,
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
    DBClusterIdentifier='string',
    StorageType='string',
    TdeCredentialArn='string',
    TdeCredentialPassword='string',
    StorageEncrypted=True|False,
    KmsKeyId='string',
    Domain='string',
    CopyTagsToSnapshot=True|False
)
```

Parameters

- **DBName** (*string*) – The meaning of this parameter differs according to the database engine you use.

Type: String

MySQL

The name of the database to create when the DB instance is created. If this parameter is not specified, no database is created in the DB instance.

Constraints:

- Must contain 1 to 64 alphanumeric characters
- Cannot be a word reserved by the specified database engine

PostgreSQL

The name of the database to create when the DB instance is created. If this parameter is not specified, the default “postgres” database is created in the DB instance.

Constraints:

- Must contain 1 to 63 alphanumeric characters
- Must begin with a letter or an underscore. Subsequent characters can be letters, underscores, or digits (0-9).
- Cannot be a word reserved by the specified database engine

Oracle

The Oracle System ID (SID) of the created DB instance.

Default: ORCL

Constraints:

- Cannot be longer than 8 characters

SQL Server

Not applicable. Must be null.

- **DBInstanceIdentifier** (*string*) – **[REQUIRED]**

The DB instance identifier. This parameter is stored as a lowercase string.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens (1 to 15 for SQL Server).
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.

Example: mydbinstance

- **AllocatedStorage** (*integer*) – The amount of storage (in gigabytes) to be initially allocated for the database instance.

Type: Integer

MySQL

Constraints: Must be an integer from 5 to 6144.

PostgreSQL

Constraints: Must be an integer from 5 to 6144.

Oracle

Constraints: Must be an integer from 10 to 6144.

SQL Server

Constraints: Must be an integer from 200 to 4096 (Standard Edition and Enterprise Edition) or from 20 to 4096 (Express Edition and Web Edition)

- **DBInstanceClass** (*string*) – [REQUIRED]

The compute and memory capacity of the DB instance.

Valid Values: db.t1.micro | db.m1.small | db.m1.medium
| db.m1.large | db.m1.xlarge | db.m2.xlarge
| db.m2.2xlarge | db.m2.4xlarge | db.m3.medium
| db.m3.large | db.m3.xlarge | db.m3.2xlarge |
db.r3.large | db.r3.xlarge | db.r3.2xlarge |
db.r3.4xlarge | db.r3.8xlarge | db.t2.micro |
db.t2.small | db.t2.medium

- **Engine** (*string*) – [REQUIRED]

The name of the database engine to be used for this instance.

Valid Values: MySQL | oracle-sel | oracle-se | oracle-ee |
sqlserver-ee | sqlserver-se | sqlserver-ex | sqlserver-web
| postgres

Not every database engine is available for every AWS region.

- **MasterUsername** (*string*) – The name of master user for the client DB instance.

MySQL

Constraints:

- Must be 1 to 16 alphanumeric characters.
- First character must be a letter.
- Cannot be a reserved word for the chosen database engine.

Type: String

Oracle

Constraints:

- Must be 1 to 30 alphanumeric characters.
- First character must be a letter.
- Cannot be a reserved word for the chosen database engine.

SQL Server

Constraints:

- Must be 1 to 128 alphanumeric characters.
- First character must be a letter.
- Cannot be a reserved word for the chosen database engine.

PostgreSQL

Constraints:

- Must be 1 to 63 alphanumeric characters.
- First character must be a letter.
- Cannot be a reserved word for the chosen database engine.

- **MasterUserPassword** (*string*) – The password for the master database user. Can be any printable ASCII character except “/”, “””, or “@”.

Type: String

MySQL

Constraints: Must contain from 8 to 41 characters.

Oracle

Constraints: Must contain from 8 to 30 characters.

SQL Server

Constraints: Must contain from 8 to 128 characters.

PostgreSQL

Constraints: Must contain from 8 to 128 characters.

- **DBSecurityGroups** (*list*) – A list of DB security groups to associate with this DB instance.

Default: The default DB security group for the database engine.

– (*string*) –

- **VpcSecurityGroupIds** (*list*) – A list of EC2 VPC security groups to associate with this DB instance.

Default: The default EC2 VPC security group for the DB subnet group's VPC.

– (*string*) –

- **AvailabilityZone** (*string*) – The EC2 Availability Zone that the database instance will be created in. For information on regions and Availability Zones, see [Regions and Availability Zones](#).

Default: A random, system-chosen Availability Zone in the endpoint's region.

Example: `us-east-1d`

Constraint: The `AvailabilityZone` parameter cannot be specified if the `MultiAZ` parameter is set to `true`. The specified Availability Zone must be in the same region as the current endpoint.

- **DBSubnetGroupName** (*string*) – A DB subnet group to associate with this DB instance.

If there is no DB subnet group, then it is a non-VPC DB instance.

- **PreferredMaintenanceWindow** (*string*) – The weekly time range during which system maintenance can occur, in Universal Coordinated Time (UTC). For more information, see [DB Instance Maintenance](#).

Format: `ddd:hh24:mi-ddd:hh24:mi`

Default: A 30-minute window selected at random from an 8-hour block of time per region, occurring on a random day of the week. To see the time blocks available, see [Adjusting the Preferred Maintenance Window](#) in the *Amazon RDS User Guide*.

Valid Days: Mon, Tue, Wed, Thu, Fri, Sat, Sun

Constraints: Minimum 30-minute window.

- **DBParameterGroupName** (*string*) – The name of the DB parameter group to associate with this DB instance. If this argument is omitted, the default `DBParameterGroup` for the specified engine will be used.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **BackupRetentionPeriod** (*integer*) – The number of days for which automated backups are retained. Setting this parameter to a positive number enables backups. Setting this parameter to 0 disables automated backups.

Default: 1

Constraints:

- Must be a value from 0 to 35
- Cannot be set to 0 if the DB instance is a source to Read Replicas

- **PreferredBackupWindow** (*string*) – The daily time range during which automated backups are created if automated backups are enabled, using the `BackupRetentionPeriod` parameter. For more information, see [DB Instance Backups](#).

Default: A 30-minute window selected at random from an 8-hour block of time per region. To see the time blocks available, see [Adjusting the Preferred Maintenance Window](#) in the *Amazon RDS User Guide*.

Constraints:

- Must be in the format `hh24:mi-hh24:mi`.
 - Times should be in Universal Coordinated Time (UTC).
 - Must not conflict with the preferred maintenance window.
 - Must be at least 30 minutes.
- **Port** (*integer*) – The port number on which the database accepts connections.

MySQL

Default: 3306

Valid Values: 1150–65535

Type: Integer

PostgreSQL

Default: 5432

Valid Values: 1150–65535

Type: Integer

Oracle

Default: 1521

Valid Values: 1150–65535

SQL Server

Default: 1433

Valid Values: 1150–65535 except for 1434, 3389, 47001, 49152, and 49152 through 49156.

- **MultiAZ** (*boolean*) – Specifies if the DB instance is a Multi-AZ deployment. You cannot set the AvailabilityZone parameter if the MultiAZ parameter is set to true. Do not set this value if you want a Multi-AZ deployment for a SQL Server DB instance. Multi-AZ for SQL Server is set using the Mirroring option in an option group.
- **EngineVersion** (*string*) – The version number of the database engine to use.

The following are the database engines and major and minor versions that are available with Amazon RDS. Not every database engine is available for every AWS region.

MySQL

- **Version 5.1** (Only available in the following regions: **ap-northeast-1**, **ap-southeast-1**, **ap-southeast-2**, **eu-west-1**, **sa-east-1**, **us-west-1**, **us-west-2**): 5.1.73a | 5.1.73b
- **Version 5.5** (Only available in the following regions: **ap-northeast-1**, **ap-southeast-1**, **ap-southeast-2**, **eu-west-1**, **sa-east-1**, **us-west-1**, **us-west-2**): 5.5.40 | 5.5.40a
- **Version 5.5** (Available in all regions): 5.5.40b | 5.5.41 | 5.5.42
- **Version 5.6** (Available in all regions): 5.6.19a | 5.6.19b | 5.6.21 | 5.6.21b | 5.6.22 | 5.6.23

MySQL

- **Version 5.1** (Only available in the following regions: **ap-northeast-1**, **ap-southeast-1**, **ap-southeast-2**, **eu-west-1**, **sa-east-1**, **us-west-1**, **us-west-2**): 5.1.73a | 5.1.73b

- **Version 5.5 (Only available in the following regions: ap-northeast-1, ap-southeast-1, ap-southeast-2, eu-west-1, sa-east-1, us-west-1, us-west-2):** 5.5.40 | 5.5.40a
- **Version 5.5 (Available in all regions):** 5.5.40b | 5.5.41 | 5.5.42
- **Version 5.6 (Available in all regions):** 5.6.19a | 5.6.19b | 5.6.21 | 5.6.21b | 5.6.22 | 5.6.23

MySQL

- **Version 5.1 (Only available in the following regions: ap-northeast-1, ap-southeast-1, ap-southeast-2, eu-west-1, sa-east-1, us-west-1, us-west-2):** 5.1.73a | 5.1.73b
- **Version 5.5 (Only available in the following regions: ap-northeast-1, ap-southeast-1, ap-southeast-2, eu-west-1, sa-east-1, us-west-1, us-west-2):** 5.5.40 | 5.5.40a
- **Version 5.5 (Available in all regions):** 5.5.40b | 5.5.41 | 5.5.42
- **Version 5.6 (Available in all regions):** 5.6.19a | 5.6.19b | 5.6.21 | 5.6.21b | 5.6.22 | 5.6.23

MySQL

- **Version 5.1 (Only available in the following regions: ap-northeast-1, ap-southeast-1, ap-southeast-2, eu-west-1, sa-east-1, us-west-1, us-west-2):** 5.1.73a | 5.1.73b
- **Version 5.5 (Only available in the following regions: ap-northeast-1, ap-southeast-1, ap-southeast-2, eu-west-1, sa-east-1, us-west-1, us-west-2):** 5.5.40 | 5.5.40a
- **Version 5.5 (Available in all regions):** 5.5.40b | 5.5.41 | 5.5.42
- **Version 5.6 (Available in all regions):** 5.6.19a | 5.6.19b | 5.6.21 | 5.6.21b | 5.6.22 | 5.6.23

Oracle Database Enterprise Edition (oracle-ee)

- **Version 11.2 (Only available in the following regions: ap-northeast-1, ap-southeast-1, ap-southeast-2, eu-west-1, sa-east-1, us-west-1, us-west-2):** 11.2.0.2.v3 | 11.2.0.2.v4 | 11.2.0.2.v5 | 11.2.0.2.v6 | 11.2.0.2.v7
- **Version 11.2 (Available in all regions):** 11.2.0.3.v1 | 11.2.0.3.v2 | 11.2.0.4.v1 | 11.2.0.4.v3
- **Version 12.1 (Available in all regions):** 12.1.0.1.v1

Oracle Database Enterprise Edition (oracle-ee)

- **Version 11.2 (Only available in the following regions: ap-northeast-1, ap-southeast-1, ap-southeast-2, eu-west-1, sa-east-1, us-west-1, us-west-2):** 11.2.0.2.v3 | 11.2.0.2.v4 | 11.2.0.2.v5 | 11.2.0.2.v6 | 11.2.0.2.v7
- **Version 11.2 (Available in all regions):** 11.2.0.3.v1 | 11.2.0.3.v2 | 11.2.0.4.v1 | 11.2.0.4.v3
- **Version 12.1 (Available in all regions):** 12.1.0.1.v1

Oracle Database Enterprise Edition (oracle-ee)

- **Version 11.2 (Only available in the following regions: ap-northeast-1, ap-southeast-1, ap-southeast-2, eu-west-1, sa-east-1, us-west-1, us-west-2):** 11.2.0.2.v3 | 11.2.0.2.v4 | 11.2.0.2.v5 | 11.2.0.2.v6 | 11.2.0.2.v7
- **Version 11.2 (Available in all regions):** 11.2.0.3.v1 | 11.2.0.3.v2 | 11.2.0.4.v1 | 11.2.0.4.v3
- **Version 12.1 (Available in all regions):** 12.1.0.1.v1

Oracle Database Standard Edition (oracle-se)

- **Version 11.2 (Only available in the following regions: us-west-1):** 11.2.0.2.v3 | 11.2.0.2.v4 | 11.2.0.2.v5 | 11.2.0.2.v6 | 11.2.0.2.v7
- **Version 11.2 (Only available in the following regions: eu-central-1, us-west-1):** 11.2.0.3.v1 | 11.2.0.3.v2 | 11.2.0.4.v1 | 11.2.0.4.v3
- **Version 12.1 (Only available in the following regions: eu-central-1, us-west-1):** 12.1.0.1.v1

Oracle Database Standard Edition (oracle-se)

- **Version 11.2 (Only available in the following regions: us-west-1):** 11.2.0.2.v3 | 11.2.0.2.v4 | 11.2.0.2.v5 | 11.2.0.2.v6 | 11.2.0.2.v7
- **Version 11.2 (Only available in the following regions: eu-central-1, us-west-1):** 11.2.0.3.v1 | 11.2.0.3.v2 | 11.2.0.4.v1 | 11.2.0.4.v3
- **Version 12.1 (Only available in the following regions: eu-central-1, us-west-1):** 12.1.0.1.v1

Oracle Database Standard Edition (oracle-se)

- **Version 11.2 (Only available in the following regions: us-west-1):** 11.2.0.2.v3 | 11.2.0.2.v4 | 11.2.0.2.v5 | 11.2.0.2.v6 | 11.2.0.2.v7
- **Version 11.2 (Only available in the following regions: eu-central-1, us-west-1):** 11.2.0.3.v1 | 11.2.0.3.v2 | 11.2.0.4.v1 | 11.2.0.4.v3
- **Version 12.1 (Only available in the following regions: eu-central-1, us-west-1):** 12.1.0.1.v1

Oracle Database Standard Edition One (oracle-se1)

- **Version 11.2 (Only available in the following regions: us-west-1):** 11.2.0.2.v3 | 11.2.0.2.v4 | 11.2.0.2.v5 | 11.2.0.2.v6 | 11.2.0.2.v7
- **Version 11.2 (Only available in the following regions: eu-central-1, us-west-1):** 11.2.0.3.v1 | 11.2.0.3.v2 | 11.2.0.4.v1 | 11.2.0.4.v3
- **Version 12.1 (Only available in the following regions: eu-central-1, us-west-1):** 12.1.0.1.v1

Oracle Database Standard Edition One (oracle-se1)

- **Version 11.2 (Only available in the following regions: us-west-1):** 11.2.0.2.v3 | 11.2.0.2.v4 | 11.2.0.2.v5 | 11.2.0.2.v6 | 11.2.0.2.v7
- **Version 11.2 (Only available in the following regions: eu-central-1, us-west-1):** 11.2.0.3.v1 | 11.2.0.3.v2 | 11.2.0.4.v1 | 11.2.0.4.v3
- **Version 12.1 (Only available in the following regions: eu-central-1, us-west-1):** 12.1.0.1.v1

Oracle Database Standard Edition One (oracle-se1)

- **Version 11.2 (Only available in the following regions: us-west-1):** 11.2.0.2.v3 | 11.2.0.2.v4 | 11.2.0.2.v5 | 11.2.0.2.v6 | 11.2.0.2.v7
- **Version 11.2 (Only available in the following regions: eu-central-1, us-west-1):** 11.2.0.3.v1 | 11.2.0.3.v2 | 11.2.0.4.v1 | 11.2.0.4.v3
- **Version 12.1 (Only available in the following regions: eu-central-1, us-west-1):** 12.1.0.1.v1

PostgreSQL

- Version 9.3 (Only available in the following regions: ap-northeast-1, ap-southeast-1, ap-southeast-2, eu-west-1, sa-east-1, us-west-1, us-west-2): 9.3.1 | 9.3.2
- Version 9.3 (Available in all regions): 9.3.3 | 9.3.5 | 9.3.6
- Version 9.4 (Available in all regions): 9.4.1

PostgreSQL

- Version 9.3 (Only available in the following regions: ap-northeast-1, ap-southeast-1, ap-southeast-2, eu-west-1, sa-east-1, us-west-1, us-west-2): 9.3.1 | 9.3.2
- Version 9.3 (Available in all regions): 9.3.3 | 9.3.5 | 9.3.6
- Version 9.4 (Available in all regions): 9.4.1

PostgreSQL

- Version 9.3 (Only available in the following regions: ap-northeast-1, ap-southeast-1, ap-southeast-2, eu-west-1, sa-east-1, us-west-1, us-west-2): 9.3.1 | 9.3.2
- Version 9.3 (Available in all regions): 9.3.3 | 9.3.5 | 9.3.6
- Version 9.4 (Available in all regions): 9.4.1

Microsoft SQL Server Enterprise Edition (sqlserver-ee)

- Version 10.50 (Only available in the following regions: eu-central-1, us-west-1): 10.50.2789.0.v1
- Version 11.00 (Only available in the following regions: eu-central-1, us-west-1): 11.00.2100.60.v1

Microsoft SQL Server Enterprise Edition (sqlserver-ee)

- Version 10.50 (Only available in the following regions: eu-central-1, us-west-1): 10.50.2789.0.v1
- Version 11.00 (Only available in the following regions: eu-central-1, us-west-1): 11.00.2100.60.v1

Microsoft SQL Server Express Edition (sqlserver-ex)

- Version 10.50 (Available in all regions): 10.50.2789.0.v1
- Version 11.00 (Available in all regions): 11.00.2100.60.v1

Microsoft SQL Server Express Edition (sqlserver-ex)

- Version 10.50 (Available in all regions): 10.50.2789.0.v1
- Version 11.00 (Available in all regions): 11.00.2100.60.v1

Microsoft SQL Server Standard Edition (sqlserver-se)

- Version 10.50 (Available in all regions): 10.50.2789.0.v1
- Version 11.00 (Available in all regions): 11.00.2100.60.v1

Microsoft SQL Server Standard Edition (sqlserver-se)

- Version 10.50 (Available in all regions): 10.50.2789.0.v1
- Version 11.00 (Available in all regions): 11.00.2100.60.v1

Microsoft SQL Server Web Edition (sqlserver-web)

- Version 10.50 (Available in all regions): 10.50.2789.0.v1
- Version 11.00 (Available in all regions): 11.00.2100.60.v1

Microsoft SQL Server Web Edition (sqlserver-web)

- Version 10.50 (Available in all regions): 10.50.2789.0.v1
- Version 11.00 (Available in all regions): 11.00.2100.60.v1

- **AutoMinorVersionUpgrade** (*boolean*) – Indicates that minor engine upgrades will be applied automatically to the DB instance during the maintenance window.

Default: true

- **LicenseModel** (*string*) – License model information for this DB instance.

Valid values: license-included | bring-your-own-license | general-public-license

- **Iops** (*integer*) – The amount of Provisioned IOPS (input/output operations per second) to be initially allocated for the DB instance.

Constraints: To use PIOPS, this value must be an integer greater than 1000.

- **OptionGroupName** (*string*) – Indicates that the DB instance should be associated with the specified option group.

Permanent options, such as the TDE option for Oracle Advanced Security TDE, cannot be removed from an option group, and that option group cannot be removed from a DB instance once it is associated with a DB instance

- **CharacterSetName** (*string*) – For supported engines, indicates that the DB instance should be associated with the specified CharacterSet.
- **PubliclyAccessible** (*boolean*) – Specifies the accessibility options for the DB instance. A value of true specifies an Internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. A value of false specifies an internal instance with a DNS name that resolves to a private IP address.

Default: The default behavior varies depending on whether a VPC has been requested or not. The following list shows the default behavior in each case.

- **Default VPC:** true
- **VPC:** false

If no DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be publicly accessible. If a specific DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be private.

- **Tags** (*list*) – A list of tags.
 - (*dict*) –

Metadata assigned to an Amazon RDS resource consisting of a key-value pair.

- * **Key** (*string*) –

A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\-]*)\$”).

- * **Value** (*string*) –

A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\-]*)\$”).

- **DBClusterIdentifier** (*string*) – The identifier of the DB cluster that the instance will belong to.

For information on creating a DB cluster, see `CreateDBCluster` .

Type: String

- **StorageType** (*string*) – Specifies the storage type to be associated with the DB instance.

Valid values: `standard` | `gp2` | `io1`

If you specify `io1` , you must also include a value for the `Iops` parameter.

Default: `io1` if the `Iops` parameter is specified; otherwise `standard`

- **TdeCredentialArn** (*string*) – The ARN from the Key Store with which to associate the instance for TDE encryption.

- **TdeCredentialPassword** (*string*) – The password for the given ARN from the Key Store in order to access the device.
- **StorageEncrypted** (*boolean*) – Specifies whether the DB instance is encrypted.

Default: false

- **KmsKeyId** (*string*) – The KMS key identifier for an encrypted DB instance.

The KMS key identifier is the Amazon Resource Name (ARN) for the KMS encryption key. If you are creating a DB instance with the same AWS account that owns the KMS encryption key used to encrypt the new DB instance, then you can use the KMS key alias instead of the ARN for the KM encryption key.

If the `StorageEncrypted` parameter is true, and you do not specify a value for the `KmsKeyId` parameter, then Amazon RDS will use your default encryption key. AWS KMS creates the default encryption key for your AWS account. Your AWS account has a different default encryption key for each AWS region.

- **Domain** (*string*) – Specify the Active Directory Domain to create the instance in.
- **CopyTagsToSnapshot** (*boolean*) – This property is not currently implemented.

Return type dict

Returns

Response Syntax

```
{
  'DBInstance': {
    'DBInstanceIdentifier': 'string',
    'DBInstanceClass': 'string',
    'Engine': 'string',
    'DBInstanceStatus': 'string',
    'MasterUsername': 'string',
    'DBName': 'string',
    'Endpoint': {
      'Address': 'string',
      'Port': 123
    },
  },
  'AllocatedStorage': 123,
  'InstanceCreateTime': datetime(2015, 1, 1),
  'PreferredBackupWindow': 'string',
  'BackupRetentionPeriod': 123,
  'DBSecurityGroups': [
    {
      'DBSecurityGroupName': 'string',
      'Status': 'string'
    },
  ],
  'VpcSecurityGroups': [
    {
      'VpcSecurityGroupId': 'string',
      'Status': 'string'
    },
  ],
  'DBParameterGroups': [
    {
      'DBParameterGroupName': 'string',
      'ParameterApplyStatus': 'string'
    },
  ],
}
```

```
'AvailabilityZone': 'string',
'DBSubnetGroup': {
    'DBSubnetGroupName': 'string',
    'DBSubnetGroupDescription': 'string',
    'VpcId': 'string',
    'SubnetGroupStatus': 'string',
    'Subnets': [
        {
            'SubnetIdentifier': 'string',
            'SubnetAvailabilityZone': {
                'Name': 'string'
            },
            'SubnetStatus': 'string'
        },
    ]
},
'PreferredMaintenanceWindow': 'string',
'PendingModifiedValues': {
    'DBInstanceClass': 'string',
    'AllocatedStorage': 123,
    'MasterUserPassword': 'string',
    'Port': 123,
    'BackupRetentionPeriod': 123,
    'MultiAZ': True|False,
    'EngineVersion': 'string',
    'Iops': 123,
    'DBInstanceIdentifier': 'string',
    'StorageType': 'string',
    'CACertificateIdentifier': 'string'
},
'LatestRestorableTime': datetime(2015, 1, 1),
'MultiAZ': True|False,
'EngineVersion': 'string',
'AutoMinorVersionUpgrade': True|False,
'ReadReplicaSourceDBInstanceIdentifier': 'string',
'ReadReplicaDBInstanceIdentifiers': [
    'string',
],
'LicenseModel': 'string',
'Iops': 123,
'OptionGroupMemberships': [
    {
        'OptionGroupName': 'string',
        'Status': 'string'
    },
],
'CharacterSetName': 'string',
'SecondaryAvailabilityZone': 'string',
'PubliclyAccessible': True|False,
'StatusInfos': [
    {
        'StatusType': 'string',
        'Normal': True|False,
        'Status': 'string',
        'Message': 'string'
    },
],
'StorageType': 'string',
```

```

        'TdeCredentialArn': 'string',
        'DbInstancePort': 123,
        'DBClusterIdentifier': 'string',
        'StorageEncrypted': True|False,
        'KmsKeyId': 'string',
        'DbiResourceId': 'string',
        'CACertificateIdentifier': 'string',
        'DomainMemberships': [
            {
                'Domain': 'string',
                'Status': 'string',
                'Connectivity': 'string'
            },
        ],
        'CopyTagsToSnapshot': True|False
    }
}

```

Response Structure

- *(dict)* –

Contains the result of a successful invocation of the following actions:

- CreateDBInstance
- DeleteDBInstance
- ModifyDBInstance

This data type is used as a response element in the DescribeDBInstances action.

- **DBInstance** *(dict)* –

Contains the result of a successful invocation of the following actions:

- * CreateDBInstance
- * DeleteDBInstance
- * ModifyDBInstance

This data type is used as a response element in the DescribeDBInstances action.

- * **DBInstanceIdentifier** *(string)* –

Contains a user-supplied database identifier. This identifier is the unique key that identifies a DB instance.

- * **DBInstanceClass** *(string)* –

Contains the name of the compute and memory capacity class of the DB instance.

- * **Engine** *(string)* –

Provides the name of the database engine to be used for this DB instance.

- * **DBInstanceStatus** *(string)* –

Specifies the current state of this database.

- * **MasterUsername** *(string)* –

Contains the master username for the DB instance.

- * **DBName** *(string)* –

The meaning of this parameter differs according to the database engine you use. For example, this value returns either MySQL or PostgreSQL information when returning values from CreateDBInstanceReadReplica since Read Replicas are only supported for MySQL and PostgreSQL.

MySQL, SQL Server, PostgreSQL

Contains the name of the initial database of this instance that was provided at create time, if one was specified when the DB instance was created. This same name is returned for the life of the DB instance.

Type: String

Oracle

Contains the Oracle System ID (SID) of the created DB instance. Not shown when the returned parameters do not apply to an Oracle DB instance.

* **Endpoint** (*dict*) –

Specifies the connection endpoint.

- **Address** (*string*) –

Specifies the DNS address of the DB instance.

- **Port** (*integer*) –

Specifies the port that the database engine is listening on.

* **AllocatedStorage** (*integer*) –

Specifies the allocated storage size specified in gigabytes.

* **InstanceCreateTime** (*datetime*) –

Provides the date and time the DB instance was created.

* **PreferredBackupWindow** (*string*) –

Specifies the daily time range during which automated backups are created if automated backups are enabled, as determined by the `BackupRetentionPeriod`.

* **BackupRetentionPeriod** (*integer*) –

Specifies the number of days for which automatic DB snapshots are retained.

* **DBSecurityGroups** (*list*) –

Provides List of DB security group elements containing only `DBSecurityGroup.Name` and `DBSecurityGroup.Status` subelements.

- (*dict*) –

This data type is used as a response element in the following actions:

- `ModifyDBInstance`
- `RebootDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceToPointInTime`
- **DBSecurityGroupName** (*string*) –

The name of the DB security group.

- **Status** (*string*) –

The status of the DB security group.

* **VpcSecurityGroups** (*list*) –

Provides List of VPC security group elements that the DB instance belongs to.

- (*dict*) –

This data type is used as a response element for queries on VPC security group membership.

- **VpcSecurityGroupId** (*string*) –

The name of the VPC security group.

- **Status** (*string*) –

The status of the VPC security group.

- * **DBParameterGroups** (*list*) –

Provides the list of DB parameter groups applied to this DB instance.

- (*dict*) –

The status of the DB parameter group.

This data type is used as a response element in the following actions:

- CreateDBInstance
- CreateDBInstanceReadReplica
- DeleteDBInstance
- ModifyDBInstance
- RebootDBInstance
- RestoreDBInstanceFromDBSnapshot
- **DBParameterGroupName** (*string*) –

The name of the DP parameter group.

- **ParameterApplyStatus** (*string*) –

The status of parameter updates.

- * **AvailabilityZone** (*string*) –

Specifies the name of the Availability Zone the DB instance is located in.

- * **DBSubnetGroup** (*dict*) –

Specifies information on the subnet group associated with the DB instance, including the name, description, and subnets in the subnet group.

- **DBSubnetGroupName** (*string*) –

Specifies the name of the DB subnet group.

- **DBSubnetGroupDescription** (*string*) –

Provides the description of the DB subnet group.

- **VpcId** (*string*) –

Provides the VpcId of the DB subnet group.

- **SubnetGroupStatus** (*string*) –

Provides the status of the DB subnet group.

- **Subnets** (*list*) –

Contains a list of Subnet elements.

- (*dict*) –

This data type is used as a response element in the DescribeDBSubnetGroups action.

- **SubnetIdentifier** (*string*) –

Specifies the identifier of the subnet.

- **SubnetAvailabilityZone** (*dict*) –
Contains Availability Zone information.
This data type is used as an element in the following data type:
 - OrderableDBInstanceOption
- **Name** (*string*) –
The name of the availability zone.
- **SubnetStatus** (*string*) –
Specifies the status of the subnet.
- * **PreferredMaintenanceWindow** (*string*) –
Specifies the weekly time range during which system maintenance can occur, in Universal Coordinated Time (UTC).
- * **PendingModifiedValues** (*dict*) –
Specifies that changes to the DB instance are pending. This element is only included when changes are pending. Specific changes are identified by subelements.
 - **DBInstanceClass** (*string*) –
Contains the new `DBInstanceClass` for the DB instance that will be applied or is in progress.
 - **AllocatedStorage** (*integer*) –
Contains the new `AllocatedStorage` size for the DB instance that will be applied or is in progress.
 - **MasterUserPassword** (*string*) –
Contains the pending or in-progress change of the master credentials for the DB instance.
 - **Port** (*integer*) –
Specifies the pending port for the DB instance.
 - **BackupRetentionPeriod** (*integer*) –
Specifies the pending number of days for which automated backups are retained.
 - **MultiAZ** (*boolean*) –
Indicates that the Single-AZ DB instance is to change to a Multi-AZ deployment.
 - **EngineVersion** (*string*) –
Indicates the database engine version.
 - **Iops** (*integer*) –
Specifies the new Provisioned IOPS value for the DB instance that will be applied or is being applied.
 - **DBInstanceIdentifier** (*string*) –
Contains the new `DBInstanceIdentifier` for the DB instance that will be applied or is in progress.
 - **StorageType** (*string*) –
Specifies the storage type to be associated with the DB instance.
 - **CACertificateIdentifier** (*string*) –

Specifies the identifier of the CA certificate for the DB instance.

* **LatestRestorableTime** (*datetime*) –

Specifies the latest time to which a database can be restored with point-in-time restore.

* **MultiAZ** (*boolean*) –

Specifies if the DB instance is a Multi-AZ deployment.

* **EngineVersion** (*string*) –

Indicates the database engine version.

* **AutoMinorVersionUpgrade** (*boolean*) –

Indicates that minor version patches are applied automatically.

* **ReadReplicaSourceDBInstanceIdentifier** (*string*) –

Contains the identifier of the source DB instance if this DB instance is a Read Replica.

* **ReadReplicaDBInstanceIdentifiers** (*list*) –

Contains one or more identifiers of the Read Replicas associated with this DB instance.

· (*string*) –

* **LicenseModel** (*string*) –

License model information for this DB instance.

* **Iops** (*integer*) –

Specifies the Provisioned IOPS (I/O operations per second) value.

* **OptionGroupMemberships** (*list*) –

Provides the list of option group memberships for this DB instance.

· (*dict*) –

Provides information on the option groups the DB instance is a member of.

· **OptionGroupName** (*string*) –

The name of the option group that the instance belongs to.

· **Status** (*string*) –

The status of the DB instance's option group membership (e.g. in-sync, pending, pending-maintenance, applying).

* **CharacterSetName** (*string*) –

If present, specifies the name of the character set that this instance is associated with.

* **SecondaryAvailabilityZone** (*string*) –

If present, specifies the name of the secondary Availability Zone for a DB instance with multi-AZ support.

* **PubliclyAccessible** (*boolean*) –

Specifies the accessibility options for the DB instance. A value of true specifies an Internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. A value of false specifies an internal instance with a DNS name that resolves to a private IP address.

Default: The default behavior varies depending on whether a VPC has been requested or not. The following list shows the default behavior in each case.

- **Default VPC:** true
- **VPC:** false

If no DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be publicly accessible. If a specific DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be private.

* **StatusInfos** (*list*) –

The status of a Read Replica. If the instance is not a Read Replica, this will be blank.

- (*dict*) –

Provides a list of status information for a DB instance.

- **StatusType** (*string*) –

This value is currently “read replication.”

- **Normal** (*boolean*) –

Boolean value that is true if the instance is operating normally, or false if the instance is in an error state.

- **Status** (*string*) –

Status of the DB instance. For a StatusType of read replica, the values can be replicating, error, stopped, or terminated.

- **Message** (*string*) –

Details of the error if there is an error for the instance. If the instance is not in an error state, this value is blank.

* **StorageType** (*string*) –

Specifies the storage type associated with DB instance.

* **TdeCredentialArn** (*string*) –

The ARN from the Key Store with which the instance is associated for TDE encryption.

* **DbInstancePort** (*integer*) –

Specifies the port that the DB instance listens on. If the DB instance is part of a DB cluster, this can be a different port than the DB cluster port.

* **DBClusterIdentifier** (*string*) –

If the DB instance is a member of a DB cluster, contains the name of the DB cluster that the DB instance is a member of.

* **StorageEncrypted** (*boolean*) –

Specifies whether the DB instance is encrypted.

* **KmsKeyId** (*string*) –

If StorageEncrypted is true, the KMS key identifier for the encrypted DB instance.

* **DbiResourceId** (*string*) –

If StorageEncrypted is true, the region-unique, immutable identifier for the encrypted DB instance. This identifier is found

in AWS CloudTrail log entries whenever the KMS key for the DB instance is accessed.

* **CACertificateIdentifier** (*string*) –

The identifier of the CA certificate for this DB instance.

* **DomainMemberships** (*list*) –

The Active Directory Domain membership records associated with the DB instance.

· (*dict*) –

An Active Directory Domain membership record associated with the DB instance.

· **Domain** (*string*) –

The identifier of the Active Directory Domain.

· **Status** (*string*) –

The status of the DB instance's Active Directory Domain membership (e.g. joined, pending-join, failed etc).

· **Connectivity** (*string*) –

The observed connectivity of the Active Directory Domain.

* **CopyTagsToSnapshot** (*boolean*) –

This property is not currently implemented.

create_db_instance_read_replica (***kwargs*)

Creates a DB instance for a DB instance running MySQL or PostgreSQL that acts as a Read Replica of a source DB instance.

All Read Replica DB instances are created as Single-AZ deployments with backups disabled. All other DB instance attributes (including DB security groups and DB parameter groups) are inherited from the source DB instance, except as specified below.

Warning: The source DB instance must have backup retention enabled.

Request Syntax

```
response = client.create_db_instance_read_replica(
    DBInstanceIdentifier='string',
    SourceDBInstanceIdentifier='string',
    DBInstanceClass='string',
    AvailabilityZone='string',
    Port=123,
    AutoMinorVersionUpgrade=True|False,
    Iops=123,
    OptionGroupName='string',
    PubliclyAccessible=True|False,
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
    DBSubnetGroupName='string',
    StorageType='string',
    CopyTagsToSnapshot=True|False
)
```

Parameters

- **DBInstanceIdentifier** (*string*) – [REQUIRED]

The DB instance identifier of the Read Replica. This identifier is the unique key that identifies a DB instance. This parameter is stored as a lowercase string.

- **SourceDBInstanceIdentifier** (*string*) – [REQUIRED]

The identifier of the DB instance that will act as the source for the Read Replica. Each DB instance can have up to five Read Replicas.

Constraints:

- Must be the identifier of an existing DB instance.
- Can specify a DB instance that is a MySQL Read Replica only if the source is running MySQL 5.6.
- Can specify a DB instance that is a PostgreSQL Read Replica only if the source is running PostgreSQL 9.3.5.
- The specified DB instance must have automatic backups enabled, its backup retention period must be greater than 0.
- If the source DB instance is in the same region as the Read Replica, specify a valid DB instance identifier.
- If the source DB instance is in a different region than the Read Replica, specify a valid DB instance ARN. For more information, go to [Constructing a Amazon RDS Amazon Resource Name \(ARN\)](#).

- **DBInstanceClass** (*string*) – The compute and memory capacity of the Read Replica.

Valid Values: db.m1.small | db.m1.medium | db.m1.large | db.m1.xlarge | db.m2.xlarge | db.m2.2xlarge | db.m2.4xlarge | db.m3.medium | db.m3.large | db.m3.xlarge | db.m3.2xlarge | db.r3.large | db.r3.xlarge | db.r3.2xlarge | db.r3.4xlarge | db.r3.8xlarge | db.t2.micro | db.t2.small | db.t2.medium

Default: Inherits from the source DB instance.

- **AvailabilityZone** (*string*) – The Amazon EC2 Availability Zone that the Read Replica will be created in.

Default: A random, system-chosen Availability Zone in the endpoint's region.

Example: us-east-1d

- **Port** (*integer*) – The port number that the DB instance uses for connections.

Default: Inherits from the source DB instance

Valid Values: 1150–65535

- **AutoMinorVersionUpgrade** (*boolean*) – Indicates that minor engine upgrades will be applied automatically to the Read Replica during the maintenance window.

Default: Inherits from the source DB instance

- **Iops** (*integer*) – The amount of Provisioned IOPS (input/output operations per second) to be initially allocated for the DB instance.
- **OptionGroupName** (*string*) – The option group the DB instance will be associated with. If omitted, the default option group for the engine specified will be used.
- **PubliclyAccessible** (*boolean*) – Specifies the accessibility options for the DB instance. A value of true specifies an Internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. A value of false

specifies an internal instance with a DNS name that resolves to a private IP address.

Default: The default behavior varies depending on whether a VPC has been requested or not. The following list shows the default behavior in each case.

- **Default VPC:** true
- **VPC:** false

If no DB subnet group has been specified as part of the request and the Publicly-Accessible value has not been set, the DB instance will be publicly accessible. If a specific DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be private.

- **Tags** (*list*) – A list of tags.
 - (*dict*) –

Metadata assigned to an Amazon RDS resource consisting of a key-value pair.

- * **Key** (*string*) –

A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=\+\\-]*)\$”).

- * **Value** (*string*) –

A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=\+\\-]*)\$”).

- **DBSubnetGroupName** (*string*) – Specifies a DB subnet group for the DB instance. The new DB instance will be created in the VPC associated with the DB subnet group. If no DB subnet group is specified, then the new DB instance is not created in a VPC.

Constraints:

- Can only be specified if the source DB instance identifier specifies a DB instance in another region.
- The specified DB subnet group must be in the same region in which the operation is running.
- All Read Replicas in one region that are created from the same source DB instance must either:
 - * Specify DB subnet groups from the same VPC. All these Read Replicas will be created in the same VPC.
- Not specify a DB subnet group. All these Read Replicas will be created outside of any VPC.

- **StorageType** (*string*) – Specifies the storage type to be associated with the Read Replica.

Valid values: `standard` | `gp2` | `io1`

If you specify `io1`, you must also include a value for the `Iops` parameter.

Default: `io1` if the `Iops` parameter is specified; otherwise `standard`

- **CopyTagsToSnapshot** (*boolean*) – This property is not currently implemented.

Return type dict

Returns

Response Syntax

```
{
  'DBInstance': {
    'DBInstanceIdentifier': 'string',
    'DBInstanceClass': 'string',
    'Engine': 'string',
    'DBInstanceStatus': 'string',
    'MasterUsername': 'string',
    'DBName': 'string',
    'Endpoint': {
      'Address': 'string',
      'Port': 123
    },
    'AllocatedStorage': 123,
    'InstanceCreateTime': datetime(2015, 1, 1),
    'PreferredBackupWindow': 'string',
    'BackupRetentionPeriod': 123,
    'DBSecurityGroups': [
      {
        'DBSecurityGroupName': 'string',
        'Status': 'string'
      },
    ],
    'VpcSecurityGroups': [
      {
        'VpcSecurityGroupId': 'string',
        'Status': 'string'
      },
    ],
    'DBParameterGroups': [
      {
        'DBParameterGroupName': 'string',
        'ParameterApplyStatus': 'string'
      },
    ],
    'AvailabilityZone': 'string',
    'DBSubnetGroup': {
      'DBSubnetGroupName': 'string',
      'DBSubnetGroupDescription': 'string',
      'VpcId': 'string',
      'SubnetGroupStatus': 'string',
      'Subnets': [
        {
          'SubnetIdentifier': 'string',
          'SubnetAvailabilityZone': {
            'Name': 'string'
          },
          'SubnetStatus': 'string'
        },
      ],
    },
    'PreferredMaintenanceWindow': 'string',
    'PendingModifiedValues': {
      'DBInstanceClass': 'string',
      'AllocatedStorage': 123,
      'MasterUserPassword': 'string',
      'Port': 123,
      'BackupRetentionPeriod': 123,
      'MultiAZ': True|False,
```



```

        'EngineVersion': 'string',
        'Iops': 123,
        'DBInstanceIdentifier': 'string',
        'StorageType': 'string',
        'CACertificateIdentifier': 'string'
    },
    'LatestRestorableTime': datetime(2015, 1, 1),
    'MultiAZ': True|False,
    'EngineVersion': 'string',
    'AutoMinorVersionUpgrade': True|False,
    'ReadReplicaSourceDBInstanceIdentifier': 'string',
    'ReadReplicaDBInstanceIdentifiers': [
        'string',
    ],
    'LicenseModel': 'string',
    'Iops': 123,
    'OptionGroupMemberships': [
        {
            'OptionGroupName': 'string',
            'Status': 'string'
        },
    ],
    'CharacterSetName': 'string',
    'SecondaryAvailabilityZone': 'string',
    'PubliclyAccessible': True|False,
    'StatusInfos': [
        {
            'StatusType': 'string',
            'Normal': True|False,
            'Status': 'string',
            'Message': 'string'
        },
    ],
    'StorageType': 'string',
    'TdeCredentialArn': 'string',
    'DbInstancePort': 123,
    'DBClusterIdentifier': 'string',
    'StorageEncrypted': True|False,
    'KmsKeyId': 'string',
    'DbiResourceId': 'string',
    'CACertificateIdentifier': 'string',
    'DomainMemberships': [
        {
            'Domain': 'string',
            'Status': 'string',
            'Connectivity': 'string'
        },
    ],
    'CopyTagsToSnapshot': True|False
}
}

```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the following actions:

- CreateDBInstance
- DeleteDBInstance

- ModifyDBInstance

This data type is used as a response element in the DescribeDBInstances action.

- **DBInstance** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * CreateDBInstance
- * DeleteDBInstance
- * ModifyDBInstance

This data type is used as a response element in the DescribeDBInstances action.

- * **DBInstanceIdentifier** (*string*) –

Contains a user-supplied database identifier. This identifier is the unique key that identifies a DB instance.

- * **DBInstanceClass** (*string*) –

Contains the name of the compute and memory capacity class of the DB instance.

- * **Engine** (*string*) –

Provides the name of the database engine to be used for this DB instance.

- * **DBInstanceStatus** (*string*) –

Specifies the current state of this database.

- * **MasterUsername** (*string*) –

Contains the master username for the DB instance.

- * **DBName** (*string*) –

The meaning of this parameter differs according to the database engine you use. For example, this value returns either MySQL or PostgreSQL information when returning values from CreateDBInstanceReadReplica since Read Replicas are only supported for MySQL and PostgreSQL.

MySQL, SQL Server, PostgreSQL

Contains the name of the initial database of this instance that was provided at create time, if one was specified when the DB instance was created. This same name is returned for the life of the DB instance.

Type: String

Oracle

Contains the Oracle System ID (SID) of the created DB instance. Not shown when the returned parameters do not apply to an Oracle DB instance.

- * **Endpoint** (*dict*) –

Specifies the connection endpoint.

- **Address** (*string*) –

Specifies the DNS address of the DB instance.

- **Port** (*integer*) –

Specifies the port that the database engine is listening on.

- * **AllocatedStorage** (*integer*) –

Specifies the allocated storage size specified in gigabytes.

- * **InstanceCreateTime** (*datetime*) –

Provides the date and time the DB instance was created.

* **PreferredBackupWindow** (*string*) –

Specifies the daily time range during which automated backups are created if automated backups are enabled, as determined by the `BackupRetentionPeriod`.

* **BackupRetentionPeriod** (*integer*) –

Specifies the number of days for which automatic DB snapshots are retained.

* **DBSecurityGroups** (*list*) –

Provides List of DB security group elements containing only `DBSecurityGroup.Name` and `DBSecurityGroup.Status` subelements.

· (*dict*) –

This data type is used as a response element in the following actions:

- `ModifyDBInstance`
- `RebootDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceToPointInTime`
- **DBSecurityGroupName** (*string*) –

The name of the DB security group.

- **Status** (*string*) –

The status of the DB security group.

* **VpcSecurityGroups** (*list*) –

Provides List of VPC security group elements that the DB instance belongs to.

· (*dict*) –

This data type is used as a response element for queries on VPC security group membership.

- **VpcSecurityGroupId** (*string*) –

The name of the VPC security group.

- **Status** (*string*) –

The status of the VPC security group.

* **DBParameterGroups** (*list*) –

Provides the list of DB parameter groups applied to this DB instance.

· (*dict*) –

The status of the DB parameter group.

This data type is used as a response element in the following actions:

- `CreateDBInstance`
- `CreateDBInstanceReadReplica`
- `DeleteDBInstance`
- `ModifyDBInstance`
- `RebootDBInstance`
- `RestoreDBInstanceFromDBSnapshot`

- **DBParameterGroupName** (*string*) –
The name of the DP parameter group.
- **ParameterApplyStatus** (*string*) –
The status of parameter updates.
- * **AvailabilityZone** (*string*) –
Specifies the name of the Availability Zone the DB instance is located in.
- * **DBSubnetGroup** (*dict*) –
Specifies information on the subnet group associated with the DB instance, including the name, description, and subnets in the subnet group.
 - **DBSubnetGroupName** (*string*) –
Specifies the name of the DB subnet group.
 - **DBSubnetGroupDescription** (*string*) –
Provides the description of the DB subnet group.
 - **VpcId** (*string*) –
Provides the VpcId of the DB subnet group.
 - **SubnetGroupStatus** (*string*) –
Provides the status of the DB subnet group.
 - **Subnets** (*list*) –
Contains a list of Subnet elements.
 - (*dict*) –
This data type is used as a response element in the DescribeDBSubnetGroups action.
 - **SubnetIdentifier** (*string*) –
Specifies the identifier of the subnet.
 - **SubnetAvailabilityZone** (*dict*) –
Contains Availability Zone information.
This data type is used as an element in the following data type:
 - **OrderableDBInstanceOption**
 - **Name** (*string*) –
The name of the availability zone.
 - **SubnetStatus** (*string*) –
Specifies the status of the subnet.
- * **PreferredMaintenanceWindow** (*string*) –
Specifies the weekly time range during which system maintenance can occur, in Universal Coordinated Time (UTC).
- * **PendingModifiedValues** (*dict*) –
Specifies that changes to the DB instance are pending. This element is only included when changes are pending. Specific changes are identified by subelements.
 - **DBInstanceClass** (*string*) –
Contains the new DBInstanceClass for the DB instance that will be applied or is in progress.

- **AllocatedStorage** (*integer*) –
Contains the new `AllocatedStorage` size for the DB instance that will be applied or is in progress.
- **MasterUserPassword** (*string*) –
Contains the pending or in-progress change of the master credentials for the DB instance.
- **Port** (*integer*) –
Specifies the pending port for the DB instance.
- **BackupRetentionPeriod** (*integer*) –
Specifies the pending number of days for which automated backups are retained.
- **MultiAZ** (*boolean*) –
Indicates that the Single-AZ DB instance is to change to a Multi-AZ deployment.
- **EngineVersion** (*string*) –
Indicates the database engine version.
- **Iops** (*integer*) –
Specifies the new Provisioned IOPS value for the DB instance that will be applied or is being applied.
- **DBInstanceIdentifier** (*string*) –
Contains the new `DBInstanceIdentifier` for the DB instance that will be applied or is in progress.
- **StorageType** (*string*) –
Specifies the storage type to be associated with the DB instance.
- **CACertificateIdentifier** (*string*) –
Specifies the identifier of the CA certificate for the DB instance.
- * **LatestRestorableTime** (*datetime*) –
Specifies the latest time to which a database can be restored with point-in-time restore.
- * **MultiAZ** (*boolean*) –
Specifies if the DB instance is a Multi-AZ deployment.
- * **EngineVersion** (*string*) –
Indicates the database engine version.
- * **AutoMinorVersionUpgrade** (*boolean*) –
Indicates that minor version patches are applied automatically.
- * **ReadReplicaSourceDBInstanceIdentifier** (*string*) –
Contains the identifier of the source DB instance if this DB instance is a Read Replica.
- * **ReadReplicaDBInstanceIdentifiers** (*list*) –
Contains one or more identifiers of the Read Replicas associated with this DB instance.
 - (*string*) –

* **LicenseModel** (*string*) –

License model information for this DB instance.

* **Iops** (*integer*) –

Specifies the Provisioned IOPS (I/O operations per second) value.

* **OptionGroupMemberships** (*list*) –

Provides the list of option group memberships for this DB instance.

· (*dict*) –

Provides information on the option groups the DB instance is a member of.

· **OptionGroupName** (*string*) –

The name of the option group that the instance belongs to.

· **Status** (*string*) –

The status of the DB instance's option group membership (e.g. in-sync, pending, pending-maintenance, applying).

* **CharacterSetName** (*string*) –

If present, specifies the name of the character set that this instance is associated with.

* **SecondaryAvailabilityZone** (*string*) –

If present, specifies the name of the secondary Availability Zone for a DB instance with multi-AZ support.

* **PubliclyAccessible** (*boolean*) –

Specifies the accessibility options for the DB instance. A value of true specifies an Internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. A value of false specifies an internal instance with a DNS name that resolves to a private IP address.

Default: The default behavior varies depending on whether a VPC has been requested or not. The following list shows the default behavior in each case.

· **Default VPC:** true

· **VPC:** false

If no DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be publicly accessible. If a specific DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be private.

* **StatusInfos** (*list*) –

The status of a Read Replica. If the instance is not a Read Replica, this will be blank.

· (*dict*) –

Provides a list of status information for a DB instance.

· **StatusType** (*string*) –

This value is currently “read replication.”

· **Normal** (*boolean*) –

Boolean value that is true if the instance is operating normally, or false if the instance is in an error state.

- **Status** (*string*) –
Status of the DB instance. For a `StatusType` of read replica, the values can be replicating, error, stopped, or terminated.
- **Message** (*string*) –
Details of the error if there is an error for the instance. If the instance is not in an error state, this value is blank.
- * **StorageType** (*string*) –
Specifies the storage type associated with DB instance.
- * **TdeCredentialArn** (*string*) –
The ARN from the Key Store with which the instance is associated for TDE encryption.
- * **DbInstancePort** (*integer*) –
Specifies the port that the DB instance listens on. If the DB instance is part of a DB cluster, this can be a different port than the DB cluster port.
- * **DBClusterIdentifier** (*string*) –
If the DB instance is a member of a DB cluster, contains the name of the DB cluster that the DB instance is a member of.
- * **StorageEncrypted** (*boolean*) –
Specifies whether the DB instance is encrypted.
- * **KmsKeyId** (*string*) –
If `StorageEncrypted` is true, the KMS key identifier for the encrypted DB instance.
- * **DbiResourceId** (*string*) –
If `StorageEncrypted` is true, the region-unique, immutable identifier for the encrypted DB instance. This identifier is found in AWS CloudTrail log entries whenever the KMS key for the DB instance is accessed.
- * **CACertificateIdentifier** (*string*) –
The identifier of the CA certificate for this DB instance.
- * **DomainMemberships** (*list*) –
The Active Directory Domain membership records associated with the DB instance.
 - (*dict*) –
An Active Directory Domain membership record associated with the DB instance.
 - **Domain** (*string*) –
The identifier of the Active Directory Domain.
 - **Status** (*string*) –
The status of the DB instance's Active Directory Domain membership (e.g. joined, pending-join, failed etc).
 - **Connectivity** (*string*) –
The observed connectivity of the Active Directory Domain.
- * **CopyTagsToSnapshot** (*boolean*) –
This property is not currently implemented.

create_db_parameter_group (**kwargs)

Creates a new DB parameter group.

A DB parameter group is initially created with the default parameters for the database engine used by the DB instance. To provide custom values for any of the parameters, you must modify the group after creating it using *ModifyDBParameterGroup*. Once you've created a DB parameter group, you need to associate it with your DB instance using *ModifyDBInstance*. When you associate a new DB parameter group with a running DB instance, you need to reboot the DB instance without failover for the new DB parameter group and associated settings to take effect.

Warning: After you create a DB parameter group, you should wait at least 5 minutes before creating your first DB instance that uses that DB parameter group as the default parameter group. This allows Amazon RDS to fully complete the create action before the parameter group is used as the default for a new DB instance. This is especially important for parameters that are critical when creating the default database for a DB instance, such as the character set for the default database defined by the `character_set_database` parameter. You can use the *Parameter Groups* option of the [Amazon RDS console](#) or the *DescribeDBParameters* command to verify that your DB parameter group has been created or modified.

Request Syntax

```
response = client.create_db_parameter_group(
    DBParameterGroupName='string',
    DBParameterGroupFamily='string',
    Description='string',
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **DBParameterGroupName** (*string*) – [REQUIRED]

The name of the DB parameter group.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Note: This value is stored as a lowercase string.

- **DBParameterGroupFamily** (*string*) – [REQUIRED]

The DB parameter group family name. A DB parameter group can be associated with one and only one DB parameter group family, and can be applied only to a DB instance running a database engine and engine version compatible with that DB parameter group family.

- **Description** (*string*) – [REQUIRED]

The description for the DB parameter group.

- **Tags** (*list*) – A list of tags.
 - (*dict*) –

Metadata assigned to an Amazon RDS resource consisting of a key-value pair.

* **Key** (*string*) –

A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

* **Value** (*string*) –

A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

Return type dict

Returns

Response Syntax

```
{
  'DBParameterGroup': {
    'DBParameterGroupName': 'string',
    'DBParameterGroupFamily': 'string',
    'Description': 'string'
  }
}
```

Response Structure

• (*dict*) –

Contains the result of a successful invocation of the CreateDBParameterGroup action.

This data type is used as a request parameter in the DeleteDBParameterGroup action, and as a response element in the DescribeDBParameterGroups action.

– **DBParameterGroup** (*dict*) –

Contains the result of a successful invocation of the CreateDBParameterGroup action.

This data type is used as a request parameter in the DeleteDBParameterGroup action, and as a response element in the DescribeDBParameterGroups action.

* **DBParameterGroupName** (*string*) –

Provides the name of the DB parameter group.

* **DBParameterGroupFamily** (*string*) –

Provides the name of the DB parameter group family that this DB parameter group is compatible with.

* **Description** (*string*) –

Provides the customer-specified description for this DB parameter group.

create_db_security_group (***kwargs*)

Creates a new DB security group. DB security groups control access to a DB instance.

Request Syntax

```
response = client.create_db_security_group(
    DBSecurityGroupName='string',
    DBSecurityGroupDescription='string',
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **DBSecurityGroupName** (*string*) – [REQUIRED]

The name for the DB security group. This value is stored as a lowercase string.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens
- Must not be “Default”
- Cannot contain spaces

Example: mysecuritygroup

- **DBSecurityGroupDescription** (*string*) – [REQUIRED]

The description for the DB security group.

- **Tags** (*list*) – A list of tags.

– (*dict*) –

Metadata assigned to an Amazon RDS resource consisting of a key-value pair.

* **Key** (*string*) –

A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

* **Value** (*string*) –

A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

Return type dict

Returns

Response Syntax

```
{
    'DBSecurityGroup': {
        'OwnerId': 'string',
        'DBSecurityGroupName': 'string',
        'DBSecurityGroupDescription': 'string',
        'VpcId': 'string',
        'EC2SecurityGroups': [
            {
```

```

        'Status': 'string',
        'EC2SecurityGroupName': 'string',
        'EC2SecurityGroupId': 'string',
        'EC2SecurityGroupOwnerId': 'string'
    },
],
'IPRanges': [
    {
        'Status': 'string',
        'CIDRIP': 'string'
    },
]
}
}

```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the following actions:

- DescribeDBSecurityGroups
- AuthorizeDBSecurityGroupIngress
- CreateDBSecurityGroup
- RevokeDBSecurityGroupIngress

This data type is used as a response element in the DescribeDBSecurityGroups action.

- **DBSecurityGroup** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * DescribeDBSecurityGroups
- * AuthorizeDBSecurityGroupIngress
- * CreateDBSecurityGroup
- * RevokeDBSecurityGroupIngress

This data type is used as a response element in the DescribeDBSecurityGroups action.

- * **OwnerId** (*string*) –

Provides the AWS ID of the owner of a specific DB security group.

- * **DBSecurityGroupName** (*string*) –

Specifies the name of the DB security group.

- * **DBSecurityGroupDescription** (*string*) –

Provides the description of the DB security group.

- * **VpcId** (*string*) –

Provides the VpcId of the DB security group.

- * **EC2SecurityGroups** (*list*) –

Contains a list of EC2SecurityGroup elements.

- (*dict*) –

This data type is used as a response element in the following actions:

- AuthorizeDBSecurityGroupIngress
- DescribeDBSecurityGroups
- RevokeDBSecurityGroupIngress
- **Status** (*string*) –

Provides the status of the EC2 security group. Status can be

- “authorizing”, “authorized”, “revoking”, and “revoked”.
- **EC2SecurityGroupName** (*string*) –
Specifies the name of the EC2 security group.
- **EC2SecurityGroupId** (*string*) –
Specifies the id of the EC2 security group.
- **EC2SecurityGroupOwnerId** (*string*) –
Specifies the AWS ID of the owner of the EC2 security group specified in the EC2SecurityGroupName field.
- * **IPRanges** (*list*) –
Contains a list of IPRange elements.
 - (*dict*) –
This data type is used as a response element in the DescribeDBSecurityGroups action.
 - **Status** (*string*) –
Specifies the status of the IP range. Status can be “authorizing”, “authorized”, “revoking”, and “revoked”.
 - **CIDRIP** (*string*) –
Specifies the IP range.

create_db_snapshot (***kwargs*)

Creates a DBSnapshot. The source DBInstance must be in “available” state.

Request Syntax

```
response = client.create_db_snapshot(  
    DBSnapshotIdentifier='string',  
    DBInstanceIdentifier='string',  
    Tags=[  
        {  
            'Key': 'string',  
            'Value': 'string'  
        },  
    ],  
)
```

Parameters

- **DBSnapshotIdentifier** (*string*) – [REQUIRED]

The identifier for the DB snapshot.

Constraints:

- Cannot be null, empty, or blank
- Must contain from 1 to 255 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Example: my-snapshot-id

- **DBInstanceIdentifier** (*string*) – [REQUIRED]

The DB instance identifier. This is the unique key that identifies a DB instance.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **Tags** (*list*) – A list of tags.
- (*dict*) –

Metadata assigned to an Amazon RDS resource consisting of a key-value pair.

- * **Key** (*string*) –

A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

- * **Value** (*string*) –

A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

Return type dict

Returns

Response Syntax

```
{
  'DBSnapshot': {
    'DBSnapshotIdentifier': 'string',
    'DBInstanceIdentifier': 'string',
    'SnapshotCreateTime': datetime(2015, 1, 1),
    'Engine': 'string',
    'AllocatedStorage': 123,
    'Status': 'string',
    'Port': 123,
    'AvailabilityZone': 'string',
    'VpcId': 'string',
    'InstanceCreateTime': datetime(2015, 1, 1),
    'MasterUsername': 'string',
    'EngineVersion': 'string',
    'LicenseModel': 'string',
    'SnapshotType': 'string',
    'Iops': 123,
    'OptionGroupName': 'string',
    'PercentProgress': 123,
    'SourceRegion': 'string',
    'SourceDBSnapshotIdentifier': 'string',
    'StorageType': 'string',
    'TdeCredentialArn': 'string',
    'Encrypted': True|False,
    'KmsKeyId': 'string'
  }
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the following actions:

- CreateDBSnapshot
- DeleteDBSnapshot

This data type is used as a response element in the DescribeDBSnapshots action.

– **DBSnapshot** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * **CreateDBSnapshot**
- * **DeleteDBSnapshot**

This data type is used as a response element in the DescribeDBSnapshots action.

* **DBSnapshotIdentifier** (*string*) –

Specifies the identifier for the DB snapshot.

* **DBInstanceIdentifier** (*string*) –

Specifies the DB instance identifier of the DB instance this DB snapshot was created from.

* **SnapshotCreateTime** (*datetime*) –

Provides the time when the snapshot was taken, in Universal Coordinated Time (UTC).

* **Engine** (*string*) –

Specifies the name of the database engine.

* **AllocatedStorage** (*integer*) –

Specifies the allocated storage size in gigabytes (GB).

* **Status** (*string*) –

Specifies the status of this DB snapshot.

* **Port** (*integer*) –

Specifies the port that the database engine was listening on at the time of the snapshot.

* **AvailabilityZone** (*string*) –

Specifies the name of the Availability Zone the DB instance was located in at the time of the DB snapshot.

* **VpcId** (*string*) –

Provides the VPC ID associated with the DB snapshot.

* **InstanceCreateTime** (*datetime*) –

Specifies the time when the snapshot was taken, in Universal Coordinated Time (UTC).

* **MasterUsername** (*string*) –

Provides the master username for the DB snapshot.

* **EngineVersion** (*string*) –

Specifies the version of the database engine.

* **LicenseModel** (*string*) –

License model information for the restored DB instance.

* **SnapshotType** (*string*) –

Provides the type of the DB snapshot.

* **Iops** (*integer*) –

Specifies the Provisioned IOPS (I/O operations per second) value of the DB instance at the time of the snapshot.

* **OptionGroupName** (*string*) –

Provides the option group name for the DB snapshot.

- * **PercentProgress** (*integer*) –
The percentage of the estimated data that has been transferred.
- * **SourceRegion** (*string*) –
The region that the DB snapshot was created in or copied from.
- * **SourceDBSnapshotIdentifier** (*string*) –
The DB snapshot Arn that the DB snapshot was copied from. It only has value in case of cross customer or cross region copy.
- * **StorageType** (*string*) –
Specifies the storage type associated with DB Snapshot.
- * **TdeCredentialArn** (*string*) –
The ARN from the Key Store with which to associate the instance for TDE encryption.
- * **Encrypted** (*boolean*) –
Specifies whether the DB snapshot is encrypted.
- * **KmsKeyId** (*string*) –
If `Encrypted` is true, the KMS key identifier for the encrypted DB snapshot.

create_db_subnet_group (***kwargs*)

Creates a new DB subnet group. DB subnet groups must contain at least one subnet in at least two AZs in the region.

Request Syntax

```
response = client.create_db_subnet_group(
    DBSubnetGroupName='string',
    DBSubnetGroupDescription='string',
    SubnetIds=[
        'string',
    ],
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **DBSubnetGroupName** (*string*) – **[REQUIRED]**
The name for the DB subnet group. This value is stored as a lowercase string.

Constraints: Must contain no more than 255 alphanumeric characters or hyphens. Must not be “Default”.

Example: mySubnetgroup
- **DBSubnetGroupDescription** (*string*) – **[REQUIRED]**
The description for the DB subnet group.
- **SubnetIds** (*list*) – **[REQUIRED]**
The EC2 Subnet IDs for the DB subnet group.
– (*string*) –
- **Tags** (*list*) – A list of tags.

– (*dict*) –

Metadata assigned to an Amazon RDS resource consisting of a key-value pair.

* **Key** (*string*) –

A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

* **Value** (*string*) –

A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

Return type dict

Returns

Response Syntax

```
{
  'DBSubnetGroup': {
    'DBSubnetGroupName': 'string',
    'DBSubnetGroupDescription': 'string',
    'VpcId': 'string',
    'SubnetGroupStatus': 'string',
    'Subnets': [
      {
        'SubnetIdentifier': 'string',
        'SubnetAvailabilityZone': {
          'Name': 'string'
        },
        'SubnetStatus': 'string'
      },
    ]
  }
}
```

Response Structure

• (*dict*) –

Contains the result of a successful invocation of the following actions:

- CreateDBSubnetGroup
- ModifyDBSubnetGroup
- DescribeDBSubnetGroups
- DeleteDBSubnetGroup

This data type is used as a response element in the DescribeDBSubnetGroups action.

– **DBSubnetGroup** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * CreateDBSubnetGroup
- * ModifyDBSubnetGroup
- * DescribeDBSubnetGroups
- * DeleteDBSubnetGroup

This data type is used as a response element in the DescribeDBSubnetGroups action.

- * **DBSubnetGroupName** (*string*) –
Specifies the name of the DB subnet group.
- * **DBSubnetGroupDescription** (*string*) –
Provides the description of the DB subnet group.
- * **VpcId** (*string*) –
Provides the VpcId of the DB subnet group.
- * **SubnetGroupStatus** (*string*) –
Provides the status of the DB subnet group.
- * **Subnets** (*list*) –
Contains a list of Subnet elements.
 - (*dict*) –
This data type is used as a response element in the DescribeDBSubnetGroups action.
 - **SubnetIdentifier** (*string*) –
Specifies the identifier of the subnet.
 - **SubnetAvailabilityZone** (*dict*) –
Contains Availability Zone information.
This data type is used as an element in the following data type:
 - OrderableDBInstanceOption
 - **Name** (*string*) –
The name of the availability zone.
 - **SubnetStatus** (*string*) –
Specifies the status of the subnet.

create_event_subscription (**kwargs)

Creates an RDS event notification subscription. This action requires a topic ARN (Amazon Resource Name) created by either the RDS console, the SNS console, or the SNS API. To obtain an ARN with SNS, you must create a topic in Amazon SNS and subscribe to the topic. The ARN is displayed in the SNS console.

You can specify the type of source (SourceType) you want to be notified of, provide a list of RDS sources (SourceIds) that triggers the events, and provide a list of event categories (EventCategories) for events you want to be notified of. For example, you can specify SourceType = db-instance, SourceIds = mydbinstance1, mydbinstance2 and EventCategories = Availability, Backup.

If you specify both the SourceType and SourceIds, such as SourceType = db-instance and SourceIdentifier = myDBInstance1, you will be notified of all the db-instance events for the specified source. If you specify a SourceType but do not specify a SourceIdentifier, you will receive notice of the events for that source type for all your RDS sources. If you do not specify either the SourceType nor the SourceIdentifier, you will be notified of events generated from all RDS sources belonging to your customer account.

Request Syntax

```
response = client.create_event_subscription(
    SubscriptionName='string',
    SnsTopicArn='string',
    SourceType='string',
    EventCategories=[
        'string',
```

```
],
SourceIds=[
    'string',
],
Enabled=True|False,
Tags=[
    {
        'Key': 'string',
        'Value': 'string'
    },
]
)
```

Parameters

- **SubscriptionName** (*string*) – [REQUIRED]

The name of the subscription.

Constraints: The name must be less than 255 characters.

- **SnsTopicArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the SNS topic created for event notification. The ARN is created by Amazon SNS when you create a topic and subscribe to it.

- **SourceType** (*string*) – The type of source that will be generating the events. For example, if you want to be notified of events generated by a DB instance, you would set this parameter to `db-instance`. If this value is not specified, all events are returned.

Valid values: `db-instance` | `db-parameter-group` | `db-security-group` | `db-snapshot`

- **EventCategories** (*list*) – A list of event categories for a `SourceType` that you want to subscribe to. You can see a list of the categories for a given `SourceType` in the [Events](#) topic in the Amazon RDS User Guide or by using the **DescribeEventCategories** action.

– (*string*) –

- **SourceIds** (*list*) – The list of identifiers of the event sources for which events will be returned. If not specified, then all sources are included in the response. An identifier must begin with a letter and must contain only ASCII letters, digits, and hyphens; it cannot end with a hyphen or contain two consecutive hyphens.

Constraints:

- If `SourceIds` are supplied, `SourceType` must also be provided.
- If the source type is a DB instance, then a `DBInstanceIdentifier` must be supplied.
- If the source type is a DB security group, a `DBSecurityGroupName` must be supplied.
- If the source type is a DB parameter group, a `DBParameterGroupName` must be supplied.
- If the source type is a DB snapshot, a `DBSnapshotIdentifier` must be supplied.

– (*string*) –

- **Enabled** (*boolean*) – A Boolean value; set to **true** to activate the subscription, set to **false** to create the subscription but not active it.

- **Tags** (*list*) – A list of tags.

– (*dict*) –

Metadata assigned to an Amazon RDS resource consisting of a key-value

pair.

* **Key** (*string*) –

A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+~]*)\$”).

* **Value** (*string*) –

A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+~]*)\$”).

Return type dict

Returns

Response Syntax

```
{
  'EventSubscription': {
    'CustomerAwsId': 'string',
    'CustSubscriptionId': 'string',
    'SnsTopicArn': 'string',
    'Status': 'string',
    'SubscriptionCreationTime': 'string',
    'SourceType': 'string',
    'SourceIdsList': [
      'string',
    ],
    'EventCategoriesList': [
      'string',
    ],
    'Enabled': True|False
  }
}
```

Response Structure

• (*dict*) –

Contains the results of a successful invocation of the DescribeEventSubscriptions action.

– **EventSubscription** (*dict*) –

Contains the results of a successful invocation of the DescribeEventSubscriptions action.

* **CustomerAwsId** (*string*) –

The AWS customer account associated with the RDS event notification subscription.

* **CustSubscriptionId** (*string*) –

The RDS event notification subscription Id.

* **SnsTopicArn** (*string*) –

The topic ARN of the RDS event notification subscription.

* **Status** (*string*) –

The status of the RDS event notification subscription.

Constraints:

Can be one of the following: creating | modifying | deleting | active | no-permission | topic-not-exist

The status “no-permission” indicates that RDS no longer has permission to post to the SNS topic. The status “topic-not-exist” indicates that the topic was deleted after the subscription was created.

* **SubscriptionCreationTime** (*string*) –

The time the RDS event notification subscription was created.

* **SourceType** (*string*) –

The source type for the RDS event notification subscription.

* **SourceIdsList** (*list*) –

A list of source IDs for the RDS event notification subscription.
· (*string*) –

* **EventCategoriesList** (*list*) –

A list of event categories for the RDS event notification subscription.
· (*string*) –

* **Enabled** (*boolean*) –

A Boolean value indicating if the subscription is enabled. True indicates the subscription is enabled.

create_option_group (***kwargs*)

Creates a new option group. You can create up to 20 option groups.

Request Syntax

```
response = client.create_option_group(
    OptionGroupName='string',
    EngineName='string',
    MajorEngineVersion='string',
    OptionGroupDescription='string',
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **OptionGroupName** (*string*) – [REQUIRED]

Specifies the name of the option group to be created.

Constraints:

- Must be 1 to 255 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Example: myoptiongroup

- **EngineName** (*string*) – [REQUIRED]

Specifies the name of the engine that this option group should be associated with.

- **MajorEngineVersion** (*string*) – [REQUIRED]

Specifies the major version of the engine that this option group should be associated with.

- **OptionGroupDescription** (*string*) – [REQUIRED]

The description of the option group.

- **Tags** (*list*) – A list of tags.

– (*dict*) –

Metadata assigned to an Amazon RDS resource consisting of a key-value pair.

- * **Key** (*string*) –

A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

- * **Value** (*string*) –

A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

Return type dict

Returns

Response Syntax

```
{
  'OptionGroup': {
    'OptionGroupName': 'string',
    'OptionGroupDescription': 'string',
    'EngineName': 'string',
    'MajorEngineVersion': 'string',
    'Options': [
      {
        'OptionName': 'string',
        'OptionDescription': 'string',
        'Persistent': True|False,
        'Permanent': True|False,
        'Port': 123,
        'OptionSettings': [
          {
            'Name': 'string',
            'Value': 'string',
            'DefaultValue': 'string',
            'Description': 'string',
            'ApplyType': 'string',
            'DataType': 'string',
            'AllowedValues': 'string',
            'IsModifiable': True|False,
            'IsCollection': True|False
          }
        ],
        'DBSecurityGroupMemberships': [
          {
            'DBSecurityGroupName': 'string',
            'Status': 'string'
          }
        ]
      }
    ]
  }
}
```

```
        },
    ],
    'VpcSecurityGroupMemberships': [
        {
            'VpcSecurityGroupId': 'string',
            'Status': 'string'
        },
    ],
},
],
'AllowsVpcAndNonVpcInstanceMemberships': True|False,
'VpcId': 'string'
}
}
```

Response Structure

- *(dict)* –
 - **OptionGroup** *(dict)* –
 - * **OptionGroupName** *(string)* –
Specifies the name of the option group.
 - * **OptionGroupDescription** *(string)* –
Provides a description of the option group.
 - * **EngineName** *(string)* –
Indicates the name of the engine that this option group can be applied to.
 - * **MajorEngineVersion** *(string)* –
Indicates the major engine version associated with this option group.
 - * **Options** *(list)* –
Indicates what options are available in the option group.
 - *(dict)* –
Option details.
 - **OptionName** *(string)* –
The name of the option.
 - **OptionDescription** *(string)* –
The description of the option.
 - **Persistent** *(boolean)* –
Indicate if this option is persistent.
 - **Permanent** *(boolean)* –
Indicate if this option is permanent.
 - **Port** *(integer)* –
If required, the port configured for this option to use.
 - **OptionSettings** *(list)* –
The option settings for this option.
 - *(dict)* –
Option settings are the actual settings being applied or configured for that option. It is used when you modify an option group or describe option groups. For example, the NATIVE_NETWORK_ENCRYPTION option has a setting

called `SQLNET.ENCRYPTION_SERVER` that can have several different values.

- **Name** (*string*) –

The name of the option that has settings that you can set.

- **Value** (*string*) –

The current value of the option setting.

- **DefaultValue** (*string*) –

The default value of the option setting.

- **Description** (*string*) –

The description of the option setting.

- **ApplyType** (*string*) –

The DB engine specific parameter type.

- **DataType** (*string*) –

The data type of the option setting.

- **AllowedValues** (*string*) –

The allowed values of the option setting.

- **IsModifiable** (*boolean*) –

A Boolean value that, when true, indicates the option setting can be modified from the default.

- **IsCollection** (*boolean*) –

Indicates if the option setting is part of a collection.

- **DBSecurityGroupMemberships** (*list*) –

If the option requires access to a port, then this DB security group allows access to the port.

- (*dict*) –

This data type is used as a response element in the following actions:

- `ModifyDBInstance`
- `RebootDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceToPointInTime`
- **DBSecurityGroupName** (*string*) –

The name of the DB security group.

- **Status** (*string*) –

The status of the DB security group.

- **VpcSecurityGroupMemberships** (*list*) –

If the option requires access to a port, then this VPC security group allows access to the port.

- (*dict*) –

This data type is used as a response element for queries on VPC security group membership.

- **VpcSecurityGroupId** (*string*) –

The name of the VPC security group.

- **Status** (*string*) –

The status of the VPC security group.

* **AllowsVpcAndNonVpcInstanceMemberships** (*boolean*) –

Indicates whether this option group can be applied to both VPC and non-VPC instances. The value `true` indicates the option group can be applied to both VPC and non-VPC instances.

* **VpcId** (*string*) –

If **AllowsVpcAndNonVpcInstanceMemberships** is `false`, this field is blank. If **AllowsVpcAndNonVpcInstanceMemberships** is `true` and this field is blank, then this option group can be applied to both VPC and non-VPC instances. If this field contains a value, then this option group can only be applied to instances that are in the VPC indicated by this field.

delete_db_cluster (***kwargs*)

The DeleteDBCluster action deletes a previously provisioned DB cluster. A successful response from the web service indicates the request was received correctly. When you delete a DB cluster, all automated backups for that DB cluster are deleted and cannot be recovered. Manual DB cluster snapshots of the DB cluster to be deleted are not deleted.

For more information on Amazon Aurora, see [Aurora on Amazon RDS](#) in the *Amazon RDS User Guide*.

Request Syntax

```
response = client.delete_db_cluster(
    DBClusterIdentifier='string',
    SkipFinalSnapshot=True|False,
    FinalDBSnapshotIdentifier='string'
)
```

Parameters

- **DBClusterIdentifier** (*string*) – The DB cluster identifier for the DB cluster to be deleted. This parameter isn't case-sensitive.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **SkipFinalSnapshot** (*boolean*) – Determines whether a final DB cluster snapshot is created before the DB cluster is deleted. If `true` is specified, no DB cluster snapshot is created. If `false` is specified, a DB cluster snapshot is created before the DB cluster is deleted.

Note: You must specify a `FinalDBSnapshotIdentifier` parameter if `SkipFinalSnapshot` is `false`.

Default: `false`

- **FinalDBSnapshotIdentifier** (*string*) – The DB cluster snapshot identifier of the new DB cluster snapshot created when `SkipFinalSnapshot` is set to `false`.

Note: Specifying this parameter and also setting the `SkipFinalSnapshot` parameter to `true` results in an error.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter

- Cannot end with a hyphen or contain two consecutive hyphens

Return type dict

Returns

Response Syntax

```
{
  'DBCluster': {
    'AllocatedStorage': 123,
    'AvailabilityZones': [
      'string',
    ],
    'BackupRetentionPeriod': 123,
    'CharacterSetName': 'string',
    'DatabaseName': 'string',
    'DBClusterIdentifier': 'string',
    'DBClusterParameterGroup': 'string',
    'DBSubnetGroup': 'string',
    'Status': 'string',
    'PercentProgress': 'string',
    'EarliestRestorableTime': datetime(2015, 1, 1),
    'Endpoint': 'string',
    'Engine': 'string',
    'EngineVersion': 'string',
    'LatestRestorableTime': datetime(2015, 1, 1),
    'Port': 123,
    'MasterUsername': 'string',
    'DBClusterOptionGroupMemberships': [
      {
        'DBClusterOptionGroupName': 'string',
        'Status': 'string'
      },
    ],
    'PreferredBackupWindow': 'string',
    'PreferredMaintenanceWindow': 'string',
    'DBClusterMembers': [
      {
        'DBInstanceIdentifier': 'string',
        'IsClusterWriter': True|False,
        'DBClusterParameterGroupStatus': 'string'
      },
    ],
    'VpcSecurityGroups': [
      {
        'VpcSecurityGroupId': 'string',
        'Status': 'string'
      },
    ],
  }
}
```

Response Structure

- (*dict*)–

Contains the result of a successful invocation of the following actions:

- CreateDBCluster
- DeleteDBCluster
- FailoverDBCluster
- ModifyDBCluster

- RestoreDBClusterFromSnapshot

This data type is used as a response element in the DescribeDBClusters action.

- **DBCluster** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * CreateDBCluster
- * DeleteDBCluster
- * FailoverDBCluster
- * ModifyDBCluster
- * RestoreDBClusterFromSnapshot

This data type is used as a response element in the DescribeDBClusters action.

- * **AllocatedStorage** (*integer*) –

Specifies the allocated storage size in gigabytes (GB).

- * **AvailabilityZones** (*list*) –

Provides the list of EC2 Availability Zones that instances in the DB cluster can be created in.

· (*string*) –

- * **BackupRetentionPeriod** (*integer*) –

Specifies the number of days for which automatic DB snapshots are retained.

- * **CharacterSetName** (*string*) –

If present, specifies the name of the character set that this cluster is associated with.

- * **DatabaseName** (*string*) –

Contains the name of the initial database of this DB cluster that was provided at create time, if one was specified when the DB cluster was created. This same name is returned for the life of the DB cluster.

- * **DBClusterIdentifier** (*string*) –

Contains a user-supplied DB cluster identifier. This identifier is the unique key that identifies a DB cluster.

- * **DBClusterParameterGroup** (*string*) –

Specifies the name of the DB cluster parameter group for the DB cluster.

- * **DBSubnetGroup** (*string*) –

Specifies information on the subnet group associated with the DB cluster, including the name, description, and subnets in the subnet group.

- * **Status** (*string*) –

Specifies the current state of this DB cluster.

- * **PercentProgress** (*string*) –

Specifies the progress of the operation as a percentage.

- * **EarliestRestorableTime** (*datetime*) –

Specifies the earliest time to which a database can be restored with point-in-time restore.

- * **Endpoint** (*string*) –

Specifies the connection endpoint for the primary instance of the DB cluster.

* **Engine** (*string*) –

Provides the name of the database engine to be used for this DB cluster.

* **EngineVersion** (*string*) –

Indicates the database engine version.

* **LatestRestorableTime** (*datetime*) –

Specifies the latest time to which a database can be restored with point-in-time restore.

* **Port** (*integer*) –

Specifies the port that the database engine is listening on.

* **MasterUsername** (*string*) –

Contains the master username for the DB cluster.

* **DBClusterOptionGroupMemberships** (*list*) –

Provides the list of option group memberships for this DB cluster.

· (*dict*) –

Contains status information for a DB cluster option group.

· **DBClusterOptionGroupName** (*string*) –

Specifies the name of the DB cluster option group.

· **Status** (*string*) –

Specifies the status of the DB cluster option group.

* **PreferredBackupWindow** (*string*) –

Specifies the daily time range during which automated backups are created if automated backups are enabled, as determined by the `BackupRetentionPeriod`.

* **PreferredMaintenanceWindow** (*string*) –

Specifies the weekly time range during which system maintenance can occur, in Universal Coordinated Time (UTC).

* **DBClusterMembers** (*list*) –

Provides the list of instances that make up the DB cluster.

· (*dict*) –

Contains information about an instance that is part of a DB cluster.

· **DBInstanceIdentifier** (*string*) –

Specifies the instance identifier for this member of the DB cluster.

· **IsClusterWriter** (*boolean*) –

Value that is `true` if the cluster member is the primary instance for the DB cluster and `false` otherwise.

· **DBClusterParameterGroupStatus** (*string*) –

Specifies the status of the DB cluster parameter group for this member of the DB cluster.

* **VpcSecurityGroups** (*list*) –

Provides a list of VPC security groups that the DB cluster belongs to.

- *(dict)* –

This data type is used as a response element for queries on VPC security group membership.

- **VpcSecurityGroupId** (*string*) –

The name of the VPC security group.

- **Status** (*string*) –

The status of the VPC security group.

delete_db_cluster_parameter_group (***kwargs*)

Deletes a specified DB cluster parameter group. The DB cluster parameter group to be deleted cannot be associated with any DB clusters.

For more information on Amazon Aurora, see [Aurora on Amazon RDS](#) in the *Amazon RDS User Guide*.

Request Syntax

```
response = client.delete_db_cluster_parameter_group(
    DBClusterParameterGroupName='string'
)
```

Parameters **DBClusterParameterGroupName** (*string*) – [REQUIRED]

The name of the DB cluster parameter group.

Constraints:

- Must be the name of an existing DB cluster parameter group.
- You cannot delete a default DB cluster parameter group.
- Cannot be associated with any DB clusters.

Returns None

delete_db_cluster_snapshot (***kwargs*)

Deletes a DB cluster snapshot. If the snapshot is being copied, the copy operation is terminated.

Note: The DB cluster snapshot must be in the `available` state to be deleted.

For more information on Amazon Aurora, see [Aurora on Amazon RDS](#) in the *Amazon RDS User Guide*.

Request Syntax

```
response = client.delete_db_cluster_snapshot(
    DBClusterSnapshotIdentifier='string'
)
```

Parameters **DBClusterSnapshotIdentifier** (*string*) – [REQUIRED]

The identifier of the DB cluster snapshot to delete.

Constraints: Must be the name of an existing DB cluster snapshot in the `available` state.

Return type dict

Returns

Response Syntax

```
{
  'DBClusterSnapshot': {
    'AvailabilityZones': [
      'string',
    ],
    'DBClusterSnapshotIdentifier': 'string',
    'DBClusterIdentifier': 'string',
    'SnapshotCreateTime': datetime(2015, 1, 1),
    'Engine': 'string',
    'AllocatedStorage': 123,
    'Status': 'string',
    'Port': 123,
    'VpcId': 'string',
    'ClusterCreateTime': datetime(2015, 1, 1),
    'MasterUsername': 'string',
    'EngineVersion': 'string',
    'LicenseModel': 'string',
    'SnapshotType': 'string',
    'PercentProgress': 123
  }
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the following actions:

- CreateDBClusterSnapshot
- DeleteDBClusterSnapshot

This data type is used as a response element in the DescribeDBClusterSnapshots action.

- **DBClusterSnapshot** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * CreateDBClusterSnapshot
- * DeleteDBClusterSnapshot

This data type is used as a response element in the DescribeDBClusterSnapshots action.

- * **AvailabilityZones** (*list*) –

Provides the list of EC2 Availability Zones that instances in the DB cluster snapshot can be restored in.

· (*string*) –

- * **DBClusterSnapshotIdentifier** (*string*) –

Specifies the identifier for the DB cluster snapshot.

- * **DBClusterIdentifier** (*string*) –

Specifies the DB cluster identifier of the DB cluster that this DB cluster snapshot was created from.

- * **SnapshotCreateTime** (*datetime*) –

Provides the time when the snapshot was taken, in Universal Coordinated Time (UTC).

- * **Engine** (*string*) –

Specifies the name of the database engine.

- * **AllocatedStorage** (*integer*) –

Specifies the allocated storage size in gigabytes (GB).

- * **Status** (*string*) –
Specifies the status of this DB cluster snapshot.
- * **Port** (*integer*) –
Specifies the port that the DB cluster was listening on at the time of the snapshot.
- * **VpcId** (*string*) –
Provides the VPC ID associated with the DB cluster snapshot.
- * **ClusterCreateTime** (*datetime*) –
Specifies the time when the DB cluster was created, in Universal Coordinated Time (UTC).
- * **MasterUsername** (*string*) –
Provides the master username for the DB cluster snapshot.
- * **EngineVersion** (*string*) –
Provides the version of the database engine for this DB cluster snapshot.
- * **LicenseModel** (*string*) –
Provides the license model information for this DB cluster snapshot.
- * **SnapshotType** (*string*) –
Provides the type of the DB cluster snapshot.
- * **PercentProgress** (*integer*) –
Specifies the percentage of the estimated data that has been transferred.

delete_db_instance (***kwargs*)

The DeleteDBInstance action deletes a previously provisioned DB instance. A successful response from the web service indicates the request was received correctly. When you delete a DB instance, all automated backups for that instance are deleted and cannot be recovered. Manual DB snapshots of the DB instance to be deleted are not deleted.

If a final DB snapshot is requested the status of the RDS instance will be “deleting” until the DB snapshot is created. The API action `DescribeDBInstance` is used to monitor the status of this operation. The action cannot be canceled or reverted once submitted.

Note that when a DB instance is in a failure state and has a status of ‘failed’, ‘incompatible-restore’, or ‘incompatible-network’, it can only be deleted when the `SkipFinalSnapshot` parameter is set to “true”.

Request Syntax

```
response = client.delete_db_instance(  
    DBInstanceIdentifier='string',  
    SkipFinalSnapshot=True|False,  
    FinalDBSnapshotIdentifier='string'  
)
```

Parameters

- **DBInstanceIdentifier** (*string*) – [REQUIRED]

The DB instance identifier for the DB instance to be deleted. This parameter isn’t case-sensitive.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter

- Cannot end with a hyphen or contain two consecutive hyphens
- **SkipFinalSnapshot** (*boolean*) – Determines whether a final DB snapshot is created before the DB instance is deleted. If `true` is specified, no DBSnapshot is created. If `false` is specified, a DB snapshot is created before the DB instance is deleted.

Note that when a DB instance is in a failure state and has a status of ‘failed’, ‘incompatible-restore’, or ‘incompatible-network’, it can only be deleted when the `SkipFinalSnapshot` parameter is set to “true”.

Specify `true` when deleting a Read Replica.

Note: The `FinalDBSnapshotIdentifier` parameter must be specified if `SkipFinalSnapshot` is `false`.

Default: `false`

- **FinalDBSnapshotIdentifier** (*string*) – The `DBSnapshotIdentifier` of the new DBSnapshot created when `SkipFinalSnapshot` is set to `false`.

Note: Specifying this parameter and also setting the `SkipFinalSnapshot` parameter to `true` results in an error.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens
- Cannot be specified when deleting a Read Replica.

Return type dict

Returns

Response Syntax

```
{
  'DBInstance': {
    'DBInstanceIdentifier': 'string',
    'DBInstanceClass': 'string',
    'Engine': 'string',
    'DBInstanceStatus': 'string',
    'MasterUsername': 'string',
    'DBName': 'string',
    'Endpoint': {
      'Address': 'string',
      'Port': 123
    },
  },
  'AllocatedStorage': 123,
  'InstanceCreateTime': datetime(2015, 1, 1),
  'PreferredBackupWindow': 'string',
  'BackupRetentionPeriod': 123,
  'DBSecurityGroups': [
    {
      'DBSecurityGroupName': 'string',
      'Status': 'string'
    },
  ],
  'VpcSecurityGroups': [
    {
      'VpcSecurityGroupId': 'string',
```

```
        'Status': 'string'
    },
],
'DBParameterGroups': [
    {
        'DBParameterGroupName': 'string',
        'ParameterApplyStatus': 'string'
    },
],
'AvailabilityZone': 'string',
'DBSubnetGroup': {
    'DBSubnetGroupName': 'string',
    'DBSubnetGroupDescription': 'string',
    'VpcId': 'string',
    'SubnetGroupStatus': 'string',
    'Subnets': [
        {
            'SubnetIdentifier': 'string',
            'SubnetAvailabilityZone': {
                'Name': 'string'
            },
            'SubnetStatus': 'string'
        },
    ],
},
'PreferredMaintenanceWindow': 'string',
'PendingModifiedValues': {
    'DBInstanceClass': 'string',
    'AllocatedStorage': 123,
    'MasterUserPassword': 'string',
    'Port': 123,
    'BackupRetentionPeriod': 123,
    'MultiAZ': True|False,
    'EngineVersion': 'string',
    'Iops': 123,
    'DBInstanceIdentifier': 'string',
    'StorageType': 'string',
    'CACertificateIdentifier': 'string'
},
'LatestRestorableTime': datetime(2015, 1, 1),
'MultiAZ': True|False,
'EngineVersion': 'string',
'AutoMinorVersionUpgrade': True|False,
'ReadReplicaSourceDBInstanceIdentifier': 'string',
'ReadReplicaDBInstanceIdentifiers': [
    'string',
],
'LicenseModel': 'string',
'Iops': 123,
'OptionGroupMemberships': [
    {
        'OptionGroupName': 'string',
        'Status': 'string'
    },
],
'CharacterSetName': 'string',
'SecondaryAvailabilityZone': 'string',
'PubliclyAccessible': True|False,
```



```

        'StatusInfos': [
            {
                'StatusType': 'string',
                'Normal': True|False,
                'Status': 'string',
                'Message': 'string'
            },
        ],
        'StorageType': 'string',
        'TdeCredentialArn': 'string',
        'DbInstancePort': 123,
        'DBClusterIdentifier': 'string',
        'StorageEncrypted': True|False,
        'KmsKeyId': 'string',
        'DbiResourceId': 'string',
        'CACertificateIdentifier': 'string',
        'DomainMemberships': [
            {
                'Domain': 'string',
                'Status': 'string',
                'Connectivity': 'string'
            },
        ],
        'CopyTagsToSnapshot': True|False
    }
}

```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the following actions:

- CreateDBInstance
- DeleteDBInstance
- ModifyDBInstance

This data type is used as a response element in the DescribeDBInstances action.

- **DBInstance** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * CreateDBInstance
- * DeleteDBInstance
- * ModifyDBInstance

This data type is used as a response element in the DescribeDBInstances action.

- * **DBInstanceIdentifier** (*string*) –

Contains a user-supplied database identifier. This identifier is the unique key that identifies a DB instance.

- * **DBInstanceClass** (*string*) –

Contains the name of the compute and memory capacity class of the DB instance.

- * **Engine** (*string*) –

Provides the name of the database engine to be used for this DB instance.

- * **DBInstanceStatus** (*string*) –

Specifies the current state of this database.

- * **MasterUsername** (*string*) –

Contains the master username for the DB instance.

* **DBName** (*string*) –

The meaning of this parameter differs according to the database engine you use. For example, this value returns either MySQL or PostgreSQL information when returning values from `CreateDBInstanceReadReplica` since Read Replicas are only supported for MySQL and PostgreSQL.

MySQL, SQL Server, PostgreSQL

Contains the name of the initial database of this instance that was provided at create time, if one was specified when the DB instance was created. This same name is returned for the life of the DB instance.

Type: String

Oracle

Contains the Oracle System ID (SID) of the created DB instance. Not shown when the returned parameters do not apply to an Oracle DB instance.

* **Endpoint** (*dict*) –

Specifies the connection endpoint.

· **Address** (*string*) –

Specifies the DNS address of the DB instance.

· **Port** (*integer*) –

Specifies the port that the database engine is listening on.

* **AllocatedStorage** (*integer*) –

Specifies the allocated storage size specified in gigabytes.

* **InstanceCreateTime** (*datetime*) –

Provides the date and time the DB instance was created.

* **PreferredBackupWindow** (*string*) –

Specifies the daily time range during which automated backups are created if automated backups are enabled, as determined by the `BackupRetentionPeriod`.

* **BackupRetentionPeriod** (*integer*) –

Specifies the number of days for which automatic DB snapshots are retained.

* **DBSecurityGroups** (*list*) –

Provides List of DB security group elements containing only `DBSecurityGroup.Name` and `DBSecurityGroup.Status` subelements.

· (*dict*) –

This data type is used as a response element in the following actions:

- `ModifyDBInstance`
- `RebootDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceToPointInTime`
- **DBSecurityGroupName** (*string*) –

The name of the DB security group.

- **Status** (*string*) –
The status of the DB security group.
- * **VpcSecurityGroups** (*list*) –
Provides List of VPC security group elements that the DB instance belongs to.
 - (*dict*) –
This data type is used as a response element for queries on VPC security group membership.
 - **VpcSecurityGroupId** (*string*) –
The name of the VPC security group.
 - **Status** (*string*) –
The status of the VPC security group.
- * **DBParameterGroups** (*list*) –
Provides the list of DB parameter groups applied to this DB instance.
 - (*dict*) –
The status of the DB parameter group.
This data type is used as a response element in the following actions:
 - CreateDBInstance
 - CreateDBInstanceReadReplica
 - DeleteDBInstance
 - ModifyDBInstance
 - RebootDBInstance
 - RestoreDBInstanceFromDBSnapshot
 - **DBParameterGroupName** (*string*) –
The name of the DP parameter group.
 - **ParameterApplyStatus** (*string*) –
The status of parameter updates.
- * **AvailabilityZone** (*string*) –
Specifies the name of the Availability Zone the DB instance is located in.
- * **DBSubnetGroup** (*dict*) –
Specifies information on the subnet group associated with the DB instance, including the name, description, and subnets in the subnet group.
 - **DBSubnetGroupName** (*string*) –
Specifies the name of the DB subnet group.
 - **DBSubnetGroupDescription** (*string*) –
Provides the description of the DB subnet group.
 - **VpcId** (*string*) –
Provides the VpcId of the DB subnet group.
 - **SubnetGroupStatus** (*string*) –
Provides the status of the DB subnet group.
 - **Subnets** (*list*) –

Contains a list of Subnet elements.

- *(dict)* –

This data type is used as a response element in the DescribeDBSubnetGroups action.

- **SubnetIdentifier** (*string*) –

Specifies the identifier of the subnet.

- **SubnetAvailabilityZone** (*dict*) –

Contains Availability Zone information.

This data type is used as an element in the following data type:

- OrderableDBInstanceOption

- **Name** (*string*) –

The name of the availability zone.

- **SubnetStatus** (*string*) –

Specifies the status of the subnet.

- * **PreferredMaintenanceWindow** (*string*) –

Specifies the weekly time range during which system maintenance can occur, in Universal Coordinated Time (UTC).

- * **PendingModifiedValues** (*dict*) –

Specifies that changes to the DB instance are pending. This element is only included when changes are pending. Specific changes are identified by subelements.

- **DBInstanceClass** (*string*) –

Contains the new DBInstanceClass for the DB instance that will be applied or is in progress.

- **AllocatedStorage** (*integer*) –

Contains the new AllocatedStorage size for the DB instance that will be applied or is in progress.

- **MasterUserPassword** (*string*) –

Contains the pending or in-progress change of the master credentials for the DB instance.

- **Port** (*integer*) –

Specifies the pending port for the DB instance.

- **BackupRetentionPeriod** (*integer*) –

Specifies the pending number of days for which automated backups are retained.

- **MultiAZ** (*boolean*) –

Indicates that the Single-AZ DB instance is to change to a Multi-AZ deployment.

- **EngineVersion** (*string*) –

Indicates the database engine version.

- **Iops** (*integer*) –

Specifies the new Provisioned IOPS value for the DB instance that will be applied or is being applied.

- **DBInstanceIdentifier** (*string*) –

- Contains the new `DBInstanceIdentifier` for the DB instance that will be applied or is in progress.
- **StorageType** (*string*) –
 - Specifies the storage type to be associated with the DB instance.
- **CACertificateIdentifier** (*string*) –
 - Specifies the identifier of the CA certificate for the DB instance.
- * **LatestRestorableTime** (*datetime*) –
 - Specifies the latest time to which a database can be restored with point-in-time restore.
- * **MultiAZ** (*boolean*) –
 - Specifies if the DB instance is a Multi-AZ deployment.
- * **EngineVersion** (*string*) –
 - Indicates the database engine version.
- * **AutoMinorVersionUpgrade** (*boolean*) –
 - Indicates that minor version patches are applied automatically.
- * **ReadReplicaSourceDBInstanceIdentifier** (*string*) –
 - Contains the identifier of the source DB instance if this DB instance is a Read Replica.
- * **ReadReplicaDBInstanceIdentifiers** (*list*) –
 - Contains one or more identifiers of the Read Replicas associated with this DB instance.
 - (*string*) –
- * **LicenseModel** (*string*) –
 - License model information for this DB instance.
- * **Iops** (*integer*) –
 - Specifies the Provisioned IOPS (I/O operations per second) value.
- * **OptionGroupMemberships** (*list*) –
 - Provides the list of option group memberships for this DB instance.
 - (*dict*) –
 - Provides information on the option groups the DB instance is a member of.
 - **OptionGroupName** (*string*) –
 - The name of the option group that the instance belongs to.
 - **Status** (*string*) –
 - The status of the DB instance's option group membership (e.g. in-sync, pending, pending-maintenance, applying).
- * **CharacterSetName** (*string*) –
 - If present, specifies the name of the character set that this instance is associated with.
- * **SecondaryAvailabilityZone** (*string*) –
 - If present, specifies the name of the secondary Availability Zone for a DB instance with multi-AZ support.
- * **PubliclyAccessible** (*boolean*) –

Specifies the accessibility options for the DB instance. A value of true specifies an Internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. A value of false specifies an internal instance with a DNS name that resolves to a private IP address.

Default: The default behavior varies depending on whether a VPC has been requested or not. The following list shows the default behavior in each case.

- **Default VPC:** true
- **VPC:** false

If no DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be publicly accessible. If a specific DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be private.

* **StatusInfos** (*list*) –

The status of a Read Replica. If the instance is not a Read Replica, this will be blank.

- (*dict*) –

Provides a list of status information for a DB instance.

- **StatusType** (*string*) –

This value is currently “read replication.”

- **Normal** (*boolean*) –

Boolean value that is true if the instance is operating normally, or false if the instance is in an error state.

- **Status** (*string*) –

Status of the DB instance. For a StatusType of read replica, the values can be replicating, error, stopped, or terminated.

- **Message** (*string*) –

Details of the error if there is an error for the instance. If the instance is not in an error state, this value is blank.

* **StorageType** (*string*) –

Specifies the storage type associated with DB instance.

* **TdeCredentialArn** (*string*) –

The ARN from the Key Store with which the instance is associated for TDE encryption.

* **DbInstancePort** (*integer*) –

Specifies the port that the DB instance listens on. If the DB instance is part of a DB cluster, this can be a different port than the DB cluster port.

* **DBClusterIdentifier** (*string*) –

If the DB instance is a member of a DB cluster, contains the name of the DB cluster that the DB instance is a member of.

* **StorageEncrypted** (*boolean*) –

Specifies whether the DB instance is encrypted.

* **KmsKeyId** (*string*) –

If `StorageEncrypted` is true, the KMS key identifier for the encrypted DB instance.

* **DbiResourceId** (*string*) –

If `StorageEncrypted` is true, the region-unique, immutable identifier for the encrypted DB instance. This identifier is found in AWS CloudTrail log entries whenever the KMS key for the DB instance is accessed.

* **CACertificateIdentifier** (*string*) –

The identifier of the CA certificate for this DB instance.

* **DomainMemberships** (*list*) –

The Active Directory Domain membership records associated with the DB instance.

· (*dict*) –

An Active Directory Domain membership record associated with the DB instance.

· **Domain** (*string*) –

The identifier of the Active Directory Domain.

· **Status** (*string*) –

The status of the DB instance's Active Directory Domain membership (e.g. joined, pending-join, failed etc).

· **Connectivity** (*string*) –

The observed connectivity of the Active Directory Domain.

* **CopyTagsToSnapshot** (*boolean*) –

This property is not currently implemented.

delete_db_parameter_group (***kwargs*)

Deletes a specified DBParameterGroup. The DBParameterGroup to be deleted cannot be associated with any DB instances.

Request Syntax

```
response = client.delete_db_parameter_group(
    DBParameterGroupName='string'
)
```

Parameters **DBParameterGroupName** (*string*) – [REQUIRED]

The name of the DB parameter group.

Constraints:

- Must be the name of an existing DB parameter group
- You cannot delete a default DB parameter group
- Cannot be associated with any DB instances

Returns None

delete_db_security_group (***kwargs*)

Deletes a DB security group.

Note: The specified DB security group must not be associated with any DB instances.

Request Syntax

```
response = client.delete_db_security_group(
    DBSecurityGroupName='string'
)
```

Parameters `DBSecurityGroupName` (*string*) – [REQUIRED]

The name of the DB security group to delete.

Note: You cannot delete the default DB security group.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens
- Must not be “Default”
- Cannot contain spaces

Returns None**delete_db_snapshot** (***kwargs*)

Deletes a DBSnapshot. If the snapshot is being copied, the copy operation is terminated.

Note: The DBSnapshot must be in the `available` state to be deleted.

Request Syntax

```
response = client.delete_db_snapshot(
    DBSnapshotIdentifier='string'
)
```

Parameters `DBSnapshotIdentifier` (*string*) – [REQUIRED]

The DBSnapshot identifier.

Constraints: Must be the name of an existing DB snapshot in the `available` state.

Return type dict**Returns****Response Syntax**

```
{
  'DBSnapshot': {
    'DBSnapshotIdentifier': 'string',
    'DBInstanceIdentifier': 'string',
    'SnapshotCreateTime': datetime(2015, 1, 1),
    'Engine': 'string',
    'AllocatedStorage': 123,
    'Status': 'string',
    'Port': 123,
    'AvailabilityZone': 'string',
    'VpcId': 'string',
    'InstanceCreateTime': datetime(2015, 1, 1),
    'MasterUsername': 'string',
    'EngineVersion': 'string',
    'LicenseModel': 'string',
    'SnapshotType': 'string',
    'Iops': 123,
    'OptionGroupName': 'string',
```



```

        'PercentProgress': 123,
        'SourceRegion': 'string',
        'SourceDBSnapshotIdentifier': 'string',
        'StorageType': 'string',
        'TdeCredentialArn': 'string',
        'Encrypted': True|False,
        'KmsKeyId': 'string'
    }
}

```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the following actions:

- CreateDBSnapshot
- DeleteDBSnapshot

This data type is used as a response element in the DescribeDBSnapshots action.

- **DBSnapshot** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * CreateDBSnapshot
- * DeleteDBSnapshot

This data type is used as a response element in the DescribeDBSnapshots action.

- * **DBSnapshotIdentifier** (*string*) –

Specifies the identifier for the DB snapshot.

- * **DBInstanceIdentifier** (*string*) –

Specifies the DB instance identifier of the DB instance this DB snapshot was created from.

- * **SnapshotCreateTime** (*datetime*) –

Provides the time when the snapshot was taken, in Universal Coordinated Time (UTC).

- * **Engine** (*string*) –

Specifies the name of the database engine.

- * **AllocatedStorage** (*integer*) –

Specifies the allocated storage size in gigabytes (GB).

- * **Status** (*string*) –

Specifies the status of this DB snapshot.

- * **Port** (*integer*) –

Specifies the port that the database engine was listening on at the time of the snapshot.

- * **AvailabilityZone** (*string*) –

Specifies the name of the Availability Zone the DB instance was located in at the time of the DB snapshot.

- * **VpcId** (*string*) –

Provides the VPC ID associated with the DB snapshot.

- * **InstanceCreateTime** (*datetime*) –

Specifies the time when the snapshot was taken, in Universal Coordinated Time (UTC).

- * **MasterUsername** (*string*) –

Provides the master username for the DB snapshot.

* **EngineVersion** (*string*) –

Specifies the version of the database engine.

* **LicenseModel** (*string*) –

License model information for the restored DB instance.

* **SnapshotType** (*string*) –

Provides the type of the DB snapshot.

* **Iops** (*integer*) –

Specifies the Provisioned IOPS (I/O operations per second) value of the DB instance at the time of the snapshot.

* **OptionGroupName** (*string*) –

Provides the option group name for the DB snapshot.

* **PercentProgress** (*integer*) –

The percentage of the estimated data that has been transferred.

* **SourceRegion** (*string*) –

The region that the DB snapshot was created in or copied from.

* **SourceDBSnapshotIdentifier** (*string*) –

The DB snapshot Arn that the DB snapshot was copied from. It only has value in case of cross customer or cross region copy.

* **StorageType** (*string*) –

Specifies the storage type associated with DB Snapshot.

* **TdeCredentialArn** (*string*) –

The ARN from the Key Store with which to associate the instance for TDE encryption.

* **Encrypted** (*boolean*) –

Specifies whether the DB snapshot is encrypted.

* **KmsKeyId** (*string*) –

If `Encrypted` is true, the KMS key identifier for the encrypted DB snapshot.

delete_db_subnet_group (***kwargs*)

Deletes a DB subnet group.

Note: The specified database subnet group must not be associated with any DB instances.

Request Syntax

```
response = client.delete_db_subnet_group(
    DBSubnetGroupName='string'
)
```

Parameters **DBSubnetGroupName** (*string*) – [REQUIRED]

The name of the database subnet group to delete.

Note: You cannot delete the default subnet group.

Constraints:

- Must be 1 to 255 alphanumeric characters

- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Returns None

delete_event_subscription (**kwargs)

Deletes an RDS event notification subscription.

Request Syntax

```
response = client.delete_event_subscription(
    SubscriptionName='string'
)
```

Parameters **SubscriptionName** (*string*) – [REQUIRED]

The name of the RDS event notification subscription you want to delete.

Return type dict

Returns

Response Syntax

```
{
  'EventSubscription': {
    'CustomerAwsId': 'string',
    'CustSubscriptionId': 'string',
    'SnsTopicArn': 'string',
    'Status': 'string',
    'SubscriptionCreationTime': 'string',
    'SourceType': 'string',
    'SourceIdsList': [
      'string',
    ],
    'EventCategoriesList': [
      'string',
    ],
    'Enabled': True|False
  }
}
```

Response Structure

- (*dict*) –

Contains the results of a successful invocation of the DescribeEventSubscriptions action.

– **EventSubscription** (*dict*) –

Contains the results of a successful invocation of the DescribeEventSubscriptions action.

* **CustomerAwsId** (*string*) –

The AWS customer account associated with the RDS event notification subscription.

* **CustSubscriptionId** (*string*) –

The RDS event notification subscription Id.

* **SnsTopicArn** (*string*) –

The topic ARN of the RDS event notification subscription.

* **Status** (*string*) –

The status of the RDS event notification subscription.

Constraints:

Can be one of the following: creating | modifying | deleting | active | no-permission | topic-not-exist

The status “no-permission” indicates that RDS no longer has permission to post to the SNS topic. The status “topic-not-exist” indicates that the topic was deleted after the subscription was created.

* **SubscriptionCreationTime** (*string*) –

The time the RDS event notification subscription was created.

* **SourceType** (*string*) –

The source type for the RDS event notification subscription.

* **SourceIdsList** (*list*) –

A list of source IDs for the RDS event notification subscription.

· (*string*) –

* **EventCategoriesList** (*list*) –

A list of event categories for the RDS event notification subscription.

· (*string*) –

* **Enabled** (*boolean*) –

A Boolean value indicating if the subscription is enabled. True indicates the subscription is enabled.

delete_option_group (***kwargs*)

Deletes an existing option group.

Request Syntax

```
response = client.delete_option_group(
    OptionGroupName='string'
)
```

Parameters **OptionGroupName** (*string*) – [REQUIRED]

The name of the option group to be deleted.

Note: You cannot delete default option groups.

Returns None

describe_account_attributes ()

Lists all of the attributes for a customer account. The attributes include Amazon RDS quotas for the account, such as the number of DB instances allowed. The description for a quota includes the quota name, current usage toward that quota, and the quota’s maximum value.

This command does not take any parameters.

Request Syntax

```
response = client.describe_account_attributes()
```

Return type dict

Returns

Response Syntax

```
{
  'AccountQuotas': [
    {
      'AccountQuotaName': 'string',
      'Used': 123,
      'Max': 123
    },
  ]
}
```

Response Structure

- *(dict)* –

Data returned by the **DescribeAccountAttributes** action.

- **AccountQuotas** (*list*) –

A list of AccountQuota objects. Within this list, each quota has a name, a count of usage toward the quota maximum, and a maximum value for the quota.

- * *(dict)* –

Describes a quota for an AWS account, for example, the number of DB instances allowed.

- **AccountQuotaName** (*string*) –

The name of the Amazon RDS quota for this AWS account.

- **Used** (*integer*) –

The amount currently used toward the quota maximum.

- **Max** (*integer*) –

The maximum allowed value for the quota.

describe_certificates (**kwargs)

Lists the set of CA certificates provided by Amazon RDS for this AWS account.

Request Syntax

```
response = client.describe_certificates(
    CertificateIdentifier='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **CertificateIdentifier** (*string*) – The user-supplied certificate identifier. If this parameter is specified, information for only the identified certificate is returned. This parameter isn't case-sensitive.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter

- Cannot end with a hyphen or contain two consecutive hyphens
- **Filters** (*list*) – This parameter is not currently supported.
 - (*dict*) –
 - * **Name** (*string*) – [REQUIRED]

This parameter is not currently supported.
 - * **Values** (*list*) – [REQUIRED]

This parameter is not currently supported.
 - (*string*) –
- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: Minimum 20, maximum 100.
- **Marker** (*string*) – An optional pagination token provided by a previous `DescribeCertificates` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Return type dict

Returns

Response Syntax

```
{
  'Certificates': [
    {
      'CertificateIdentifier': 'string',
      'CertificateType': 'string',
      'Thumbprint': 'string',
      'ValidFrom': datetime(2015, 1, 1),
      'ValidTill': datetime(2015, 1, 1)
    },
  ],
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Data returned by the **DescribeCertificates** action.
 - **Certificates** (*list*) –

The list of Certificate objects for the AWS account.
 - * (*dict*) –

A CA certificate for an AWS account.
 - **CertificateIdentifier** (*string*) –

The unique key that identifies a certificate.
 - **CertificateType** (*string*) –

The type of the certificate.
 - **Thumbprint** (*string*) –

The thumbprint of the certificate.
 - **ValidFrom** (*datetime*) –

The starting date from which the certificate is valid.

· **ValidTill** (*datetime*) –

The final date that the certificate continues to be valid.

– **Marker** (*string*) –

An optional pagination token provided by a previous DescribeCertificates request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

describe_db_cluster_parameter_groups (***kwargs*)

Returns a list of `DBClusterParameterGroup` descriptions. If a `DBClusterParameterGroupName` parameter is specified, the list will contain only the description of the specified DB cluster parameter group.

For more information on Amazon Aurora, see [Aurora on Amazon RDS](#) in the *Amazon RDS User Guide*.

Request Syntax

```
response = client.describe_db_cluster_parameter_groups(
    DBClusterParameterGroupName='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **DBClusterParameterGroupName** (*string*) – The name of a specific DB cluster parameter group to return details for.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **Filters** (*list*) – This parameter is not currently supported.

– (*dict*) –

* **Name** (*string*) – [REQUIRED]

This parameter is not currently supported.

* **Values** (*list*) – [REQUIRED]

This parameter is not currently supported.

· (*string*) –

- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: Minimum 20, maximum 100.

- **Marker** (*string*) – An optional pagination token provided by a previous `DescribeDBClusterParameterGroups` request. If this parameter is

specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Return type dict

Returns

Response Syntax

```
{
    'Marker': 'string',
    'DBClusterParameterGroups': [
        {
            'DBClusterParameterGroupName': 'string',
            'DBParameterGroupFamily': 'string',
            'Description': 'string'
        },
    ],
}
```

Response Structure

- (*dict*) –

- **Marker** (*string*) –

An optional pagination token provided by a previous `DescribeDBClusterParameterGroups` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

- **DBClusterParameterGroups** (*list*) –

A list of DB cluster parameter groups.

- * (*dict*) –

Contains the result of a successful invocation of the `CreateDBClusterParameterGroup` action.

This data type is used as a request parameter in the `DeleteDBClusterParameterGroup` action, and as a response element in the `DescribeDBClusterParameterGroups` action.

- **DBClusterParameterGroupName** (*string*) –

Provides the name of the DB cluster parameter group.

- **DBParameterGroupFamily** (*string*) –

Provides the name of the DB parameter group family that this DB cluster parameter group is compatible with.

- **Description** (*string*) –

Provides the customer-specified description for this DB cluster parameter group.

describe_db_cluster_parameters (***kwargs*)

Returns the detailed parameter list for a particular DB cluster parameter group.

For more information on Amazon Aurora, see [Aurora on Amazon RDS](#) in the *Amazon RDS User Guide*.

Request Syntax

```
response = client.describe_db_cluster_parameters(
    DBClusterParameterGroupName='string',
    Source='string',
    Filters=[
```



```

        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxRecords=123,
    Marker='string'
)

```

Parameters

- **DBClusterParameterGroupName** (*string*) – [REQUIRED]

The name of a specific DB cluster parameter group to return parameter details for.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **Source** (*string*) – A value that indicates to return only parameters for a specific source. Parameter sources can be `engine`, `service`, or `customer`.
- **Filters** (*list*) – This parameter is not currently supported.

– (*dict*) –

- * **Name** (*string*) – [REQUIRED]

This parameter is not currently supported.

- * **Values** (*list*) – [REQUIRED]

This parameter is not currently supported.

· (*string*) –

- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: Minimum 20, maximum 100.

- **Marker** (*string*) – An optional pagination token provided by a previous `DescribeDBClusterParameters` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Return type dict

Returns

Response Syntax

```

{
    'Parameters': [
        {
            'ParameterName': 'string',
            'ParameterValue': 'string',
            'Description': 'string',
            'Source': 'string',
            'ApplyType': 'string',
            'DataType': 'string',

```

```
        'AllowedValues': 'string',
        'IsModifiable': True|False,
        'MinimumEngineVersion': 'string',
        'ApplyMethod': 'immediate'|'pending-reboot'
    },
],
'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Provides details about a DB cluster parameter group including the parameters in the DB cluster parameter group.

- **Parameters** (*list*) –

Provides a list of parameters for the DB cluster parameter group.

- * (*dict*) –

This data type is used as a request parameter in the ModifyDBParameterGroup and ResetDBParameterGroup actions.

This data type is used as a response element in the DescribeEngineDefaultParameters and DescribeDBParameters actions.

- **ParameterName** (*string*) –

Specifies the name of the parameter.

- **ParameterValue** (*string*) –

Specifies the value of the parameter.

- **Description** (*string*) –

Provides a description of the parameter.

- **Source** (*string*) –

Indicates the source of the parameter value.

- **ApplyType** (*string*) –

Specifies the engine specific parameters type.

- **DataType** (*string*) –

Specifies the valid data type for the parameter.

- **AllowedValues** (*string*) –

Specifies the valid range of values for the parameter.

- **IsModifiable** (*boolean*) –

Indicates whether (`true`) or not (`false`) the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.

- **MinimumEngineVersion** (*string*) –

The earliest engine version to which the parameter can apply.

- **ApplyMethod** (*string*) –

Indicates when to apply parameter updates.

- **Marker** (*string*) –

An optional pagination token provided by a previous DescribeDBClusterParameters request. If this parameter is specified, the response in-

cludes only records beyond the marker, up to the value specified by `MaxRecords`.

describe_db_cluster_snapshots (**kwargs)

Returns information about DB cluster snapshots. This API supports pagination.

For more information on Amazon Aurora, see [Aurora on Amazon RDS](#) in the *Amazon RDS User Guide*.

Request Syntax

```
response = client.describe_db_cluster_snapshots(
    DBClusterIdentifier='string',
    DBClusterSnapshotIdentifier='string',
    SnapshotType='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **DBClusterIdentifier** (*string*) – A DB cluster identifier to retrieve the list of DB cluster snapshots for. This parameter cannot be used in conjunction with the `DBClusterSnapshotIdentifier` parameter. This parameter is not case-sensitive.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **DBClusterSnapshotIdentifier** (*string*) – A specific DB cluster snapshot identifier to describe. This parameter cannot be used in conjunction with the `DBClusterIdentifier` parameter. This value is stored as a lowercase string.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens
- If this is the identifier of an automated snapshot, the `SnapshotType` parameter must also be specified.

- **SnapshotType** (*string*) – The type of DB cluster snapshots that will be returned. Values can be `automated` or `manual`. If this parameter is not specified, the returned results will include all snapshot types.

- **Filters** (*list*) – This parameter is not currently supported.

– (*dict*) –

* **Name** (*string*) – **[REQUIRED]**

This parameter is not currently supported.

* **Values** (*list*) – **[REQUIRED]**

This parameter is not currently supported.

· (*string*) –

- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: Minimum 20, maximum 100.

- **Marker** (*string*) – An optional pagination token provided by a previous `DescribeDBClusterSnapshots` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'DBClusterSnapshots': [
    {
      'AvailabilityZones': [
        'string',
      ],
      'DBClusterSnapshotIdentifier': 'string',
      'DBClusterIdentifier': 'string',
      'SnapshotCreateTime': datetime(2015, 1, 1),
      'Engine': 'string',
      'AllocatedStorage': 123,
      'Status': 'string',
      'Port': 123,
      'VpcId': 'string',
      'ClusterCreateTime': datetime(2015, 1, 1),
      'MasterUsername': 'string',
      'EngineVersion': 'string',
      'LicenseModel': 'string',
      'SnapshotType': 'string',
      'PercentProgress': 123
    },
  ]
}
```

Response Structure

- (*dict*) –

Provides a list of DB cluster snapshots for the user as the result of a call to the `DescribeDBClusterSnapshots` action.

- **Marker** (*string*) –

An optional pagination token provided by a previous `DescribeDBClusterSnapshots` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

- **DBClusterSnapshots** (*list*) –

Provides a list of DB cluster snapshots for the user.

- * (*dict*) –

Contains the result of a successful invocation of the following actions:

- `CreateDBClusterSnapshot`
- `DeleteDBClusterSnapshot`

This data type is used as a response element in the `DescribeDBClusterSnapshots` action.

- **AvailabilityZones** (*list*) –
Provides the list of EC2 Availability Zones that instances in the DB cluster snapshot can be restored in.
- (*string*) –
- **DBClusterSnapshotIdentifier** (*string*) –
Specifies the identifier for the DB cluster snapshot.
- **DBClusterIdentifier** (*string*) –
Specifies the DB cluster identifier of the DB cluster that this DB cluster snapshot was created from.
- **SnapshotCreateTime** (*datetime*) –
Provides the time when the snapshot was taken, in Universal Coordinated Time (UTC).
- **Engine** (*string*) –
Specifies the name of the database engine.
- **AllocatedStorage** (*integer*) –
Specifies the allocated storage size in gigabytes (GB).
- **Status** (*string*) –
Specifies the status of this DB cluster snapshot.
- **Port** (*integer*) –
Specifies the port that the DB cluster was listening on at the time of the snapshot.
- **VpcId** (*string*) –
Provides the VPC ID associated with the DB cluster snapshot.
- **ClusterCreateTime** (*datetime*) –
Specifies the time when the DB cluster was created, in Universal Coordinated Time (UTC).
- **MasterUsername** (*string*) –
Provides the master username for the DB cluster snapshot.
- **EngineVersion** (*string*) –
Provides the version of the database engine for this DB cluster snapshot.
- **LicenseModel** (*string*) –
Provides the license model information for this DB cluster snapshot.
- **SnapshotType** (*string*) –
Provides the type of the DB cluster snapshot.
- **PercentProgress** (*integer*) –
Specifies the percentage of the estimated data that has been transferred.

describe_db_clusters (**kwargs)

Returns information about provisioned Aurora DB clusters. This API supports pagination.

For more information on Amazon Aurora, see [Aurora on Amazon RDS](#) in the *Amazon RDS User Guide*.

Request Syntax

```
response = client.describe_db_clusters(
    DBClusterIdentifier='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **DBClusterIdentifier** (*string*) – The user-supplied DB cluster identifier. If this parameter is specified, information from only the specific DB cluster is returned. This parameter isn't case-sensitive.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **Filters** (*list*) – This parameter is not currently supported.
 - (*dict*) –

- * **Name** (*string*) – **[REQUIRED]**

- This parameter is not currently supported.

- * **Values** (*list*) – **[REQUIRED]**

- This parameter is not currently supported.

- (*string*) –

- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: Minimum 20, maximum 100.

- **Marker** (*string*) – An optional pagination token provided by a previous `DescribeDBClusters` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Return type dict

Returns**Response Syntax**

```
{
    'Marker': 'string',
    'DBClusters': [
        {
```

```

        'AllocatedStorage': 123,
        'AvailabilityZones': [
            'string',
        ],
        'BackupRetentionPeriod': 123,
        'CharacterSetName': 'string',
        'DatabaseName': 'string',
        'DBClusterIdentifier': 'string',
        'DBClusterParameterGroup': 'string',
        'DBSubnetGroup': 'string',
        'Status': 'string',
        'PercentProgress': 'string',
        'EarliestRestorableTime': datetime(2015, 1, 1),
        'Endpoint': 'string',
        'Engine': 'string',
        'EngineVersion': 'string',
        'LatestRestorableTime': datetime(2015, 1, 1),
        'Port': 123,
        'MasterUsername': 'string',
        'DBClusterOptionGroupMemberships': [
            {
                'DBClusterOptionGroupName': 'string',
                'Status': 'string'
            },
        ],
        'PreferredBackupWindow': 'string',
        'PreferredMaintenanceWindow': 'string',
        'DBClusterMembers': [
            {
                'DBInstanceIdentifier': 'string',
                'IsClusterWriter': True|False,
                'DBClusterParameterGroupStatus': 'string'
            },
        ],
        'VpcSecurityGroups': [
            {
                'VpcSecurityGroupId': 'string',
                'Status': 'string'
            },
        ],
    ],
}

```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the DescribeDBClusters action.

- **Marker** (*string*) –

A pagination token that can be used in a subsequent DescribeDBClusters request.

- **DBClusters** (*list*) –

Contains a list of DB clusters for the user.

- * (*dict*) –

Contains the result of a successful invocation of the following actions:

- CreateDBCluster
- DeleteDBCluster
- FailoverDBCluster
- ModifyDBCluster
- RestoreDBClusterFromSnapshot

This data type is used as a response element in the DescribeDBClusters action.

- **AllocatedStorage** (*integer*) –

Specifies the allocated storage size in gigabytes (GB).

- **AvailabilityZones** (*list*) –

Provides the list of EC2 Availability Zones that instances in the DB cluster can be created in.

- (*string*) –

- **BackupRetentionPeriod** (*integer*) –

Specifies the number of days for which automatic DB snapshots are retained.

- **CharacterSetName** (*string*) –

If present, specifies the name of the character set that this cluster is associated with.

- **DatabaseName** (*string*) –

Contains the name of the initial database of this DB cluster that was provided at create time, if one was specified when the DB cluster was created. This same name is returned for the life of the DB cluster.

- **DBClusterIdentifier** (*string*) –

Contains a user-supplied DB cluster identifier. This identifier is the unique key that identifies a DB cluster.

- **DBClusterParameterGroup** (*string*) –

Specifies the name of the DB cluster parameter group for the DB cluster.

- **DBSubnetGroup** (*string*) –

Specifies information on the subnet group associated with the DB cluster, including the name, description, and subnets in the subnet group.

- **Status** (*string*) –

Specifies the current state of this DB cluster.

- **PercentProgress** (*string*) –

Specifies the progress of the operation as a percentage.

- **EarliestRestorableTime** (*datetime*) –

Specifies the earliest time to which a database can be restored with point-in-time restore.

- **Endpoint** (*string*) –

Specifies the connection endpoint for the primary instance of the DB cluster.

- **Engine** (*string*) –

Provides the name of the database engine to be used for this DB cluster.

- **EngineVersion** (*string*) –
Indicates the database engine version.
- **LatestRestorableTime** (*datetime*) –
Specifies the latest time to which a database can be restored with point-in-time restore.
- **Port** (*integer*) –
Specifies the port that the database engine is listening on.
- **MasterUsername** (*string*) –
Contains the master username for the DB cluster.
- **DBClusterOptionGroupMemberships** (*list*) –
Provides the list of option group memberships for this DB cluster.
- (*dict*) –
Contains status information for a DB cluster option group.
- **DBClusterOptionGroupName** (*string*) –
Specifies the name of the DB cluster option group.
- **Status** (*string*) –
Specifies the status of the DB cluster option group.
- **PreferredBackupWindow** (*string*) –
Specifies the daily time range during which automated backups are created if automated backups are enabled, as determined by the `BackupRetentionPeriod`.
- **PreferredMaintenanceWindow** (*string*) –
Specifies the weekly time range during which system maintenance can occur, in Universal Coordinated Time (UTC).
- **DBClusterMembers** (*list*) –
Provides the list of instances that make up the DB cluster.
- (*dict*) –
Contains information about an instance that is part of a DB cluster.
- **DBInstanceIdentifier** (*string*) –
Specifies the instance identifier for this member of the DB cluster.
- **IsClusterWriter** (*boolean*) –
Value that is `true` if the cluster member is the primary instance for the DB cluster and `false` otherwise.
- **DBClusterParameterGroupStatus** (*string*) –
Specifies the status of the DB cluster parameter group for this member of the DB cluster.
- **VpcSecurityGroups** (*list*) –
Provides a list of VPC security groups that the DB cluster belongs to.
- (*dict*) –
This data type is used as a response element for queries on VPC security group membership.

- **VpcSecurityGroupId** (*string*) –
The name of the VPC security group.
- **Status** (*string*) –
The status of the VPC security group.

describe_db_engine_versions (***kwargs*)

Returns a list of the available DB engines.

Request Syntax

```
response = client.describe_db_engine_versions(  
    Engine='string',  
    EngineVersion='string',  
    DBParameterGroupFamily='string',  
    Filters=[  
        {  
            'Name': 'string',  
            'Values': [  
                'string',  
            ]  
        },  
    ],  
    MaxRecords=123,  
    Marker='string',  
    DefaultOnly=True|False,  
    ListSupportedCharacterSets=True|False  
)
```

Parameters

- **Engine** (*string*) – The database engine to return.
- **EngineVersion** (*string*) – The database engine version to return.

Example: 5.1.49

- **DBParameterGroupFamily** (*string*) – The name of a specific DB parameter group family to return details for.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **Filters** (*list*) – Not currently supported.

– (*dict*) –

* **Name** (*string*) – **[REQUIRED]**

This parameter is not currently supported.

* **Values** (*list*) – **[REQUIRED]**

This parameter is not currently supported.

· (*string*) –

- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more than the `MaxRecords` value is available, a pagination token called a marker is included in the response so that the following results can be retrieved.

Default: 100

Constraints: Minimum 20, maximum 100.

- **Marker** (*string*) – An optional pagination token provided by a previous request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.
- **DefaultOnly** (*boolean*) – Indicates that only the default version of the specified engine or engine and major version combination is returned.
- **ListSupportedCharacterSets** (*boolean*) – If this parameter is specified, and if the requested engine supports the `CharacterSetName` parameter for `CreateDBInstance`, the response includes a list of supported character sets for each engine version.

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'DBEngineVersions': [
    {
      'Engine': 'string',
      'EngineVersion': 'string',
      'DBParameterGroupFamily': 'string',
      'DBEngineDescription': 'string',
      'DBEngineVersionDescription': 'string',
      'DefaultCharacterSet': {
        'CharacterSetName': 'string',
        'CharacterSetDescription': 'string'
      },
      'SupportedCharacterSets': [
        {
          'CharacterSetName': 'string',
          'CharacterSetDescription': 'string'
        },
        ...
      ]
    },
    ...
  ]
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the `DescribeDBEngineVersions` action.

- **Marker** (*string*) –

An optional pagination token provided by a previous request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

- **DBEngineVersions** (*list*) –

A list of `DBEngineVersion` elements.

- * (*dict*) –

This data type is used as a response element in the action `DescribeDBEngineVersions`.

- **Engine** (*string*) –

The name of the database engine.

- **EngineVersion** (*string*) –

The version number of the database engine.

- **DBParameterGroupFamily** (*string*) –

The name of the DB parameter group family for the database engine.

- **DBEngineDescription** (*string*) –

The description of the database engine.

- **DBEngineVersionDescription** (*string*) –

The description of the database engine version.

- **DefaultCharacterSet** (*dict*) –

The default character set for new instances of this engine version, if the `CharacterSetName` parameter of the `CreateDBInstance` API is not specified.

- **CharacterSetName** (*string*) –

The name of the character set.

- **CharacterSetDescription** (*string*) –

The description of the character set.

- **SupportedCharacterSets** (*list*) –

A list of the character sets supported by this engine for the `CharacterSetName` parameter of the `CreateDBInstance` API.

- (*dict*) –

This data type is used as a response element in the action `DescribeDBEngineVersions`.

- **CharacterSetName** (*string*) –

The name of the character set.

- **CharacterSetDescription** (*string*) –

The description of the character set.

describe_db_instances (***kwargs*)

Returns information about provisioned RDS instances. This API supports pagination.

Request Syntax

```
response = client.describe_db_instances(
    DBInstanceIdentifier='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **DBInstanceIdentifier** (*string*) – The user-supplied instance identifier. If this parameter is specified, information from only the specific DB instance is returned. This parameter isn't case-sensitive.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens
- **Filters** (*list*) – This parameter is not currently supported.
 - (*dict*) –
 - * **Name** (*string*) – [REQUIRED]

This parameter is not currently supported.
 - * **Values** (*list*) – [REQUIRED]

This parameter is not currently supported.
 - (*string*) –
- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: Minimum 20, maximum 100.

- **Marker** (*string*) – An optional pagination token provided by a previous `DescribeDBInstances` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'DBInstances': [
    {
      'DBInstanceIdentifier': 'string',
      'DBInstanceClass': 'string',
      'Engine': 'string',
      'DBInstanceStatus': 'string',
      'MasterUsername': 'string',
      'DBName': 'string',
      'Endpoint': {
        'Address': 'string',
        'Port': 123
      },
      'AllocatedStorage': 123,
      'InstanceCreateTime': datetime(2015, 1, 1),
      'PreferredBackupWindow': 'string',
      'BackupRetentionPeriod': 123,
      'DBSecurityGroups': [
        {
          'DBSecurityGroupName': 'string',
          'Status': 'string'
        }
      ],
      'VpcSecurityGroups': [
        {
          'VpcSecurityGroupId': 'string',
          'Status': 'string'
        }
      ]
    }
  ]
}
```

```
    },
],
'DBParameterGroups': [
    {
        'DBParameterGroupName': 'string',
        'ParameterApplyStatus': 'string'
    },
],
'AvailabilityZone': 'string',
'DBSubnetGroup': {
    'DBSubnetGroupName': 'string',
    'DBSubnetGroupDescription': 'string',
    'VpcId': 'string',
    'SubnetGroupStatus': 'string',
    'Subnets': [
        {
            'SubnetIdentifier': 'string',
            'SubnetAvailabilityZone': {
                'Name': 'string'
            },
            'SubnetStatus': 'string'
        },
    ],
}
],
'PreferredMaintenanceWindow': 'string',
'PendingModifiedValues': {
    'DBInstanceClass': 'string',
    'AllocatedStorage': 123,
    'MasterUserPassword': 'string',
    'Port': 123,
    'BackupRetentionPeriod': 123,
    'MultiAZ': True|False,
    'EngineVersion': 'string',
    'Iops': 123,
    'DBInstanceIdentifier': 'string',
    'StorageType': 'string',
    'CACertificateIdentifier': 'string'
},
'LatestRestorableTime': datetime(2015, 1, 1),
'MultiAZ': True|False,
'EngineVersion': 'string',
'AutoMinorVersionUpgrade': True|False,
'ReadReplicaSourceDBInstanceIdentifier': 'string',
'ReadReplicaDBInstanceIdentifiers': [
    'string',
],
'LicenseModel': 'string',
'Iops': 123,
'OptionGroupMemberships': [
    {
        'OptionGroupName': 'string',
        'Status': 'string'
    },
],
'CharacterSetName': 'string',
'SecondaryAvailabilityZone': 'string',
'PubliclyAccessible': True|False,
'StatusInfos': [
```

```

        {
            'StatusType': 'string',
            'Normal': True|False,
            'Status': 'string',
            'Message': 'string'
        },
    ],
    'StorageType': 'string',
    'TdeCredentialArn': 'string',
    'DbInstancePort': 123,
    'DBClusterIdentifier': 'string',
    'StorageEncrypted': True|False,
    'KmsKeyId': 'string',
    'DbiResourceId': 'string',
    'CACertificateIdentifier': 'string',
    'DomainMemberships': [
        {
            'Domain': 'string',
            'Status': 'string',
            'Connectivity': 'string'
        },
    ],
    'CopyTagsToSnapshot': True|False
},
]
}

```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the DescribeDBInstances action.

- **Marker** (*string*) –

An optional pagination token provided by a previous request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

- **DBInstances** (*list*) –

A list of DBInstance instances.

- * (*dict*) –

Contains the result of a successful invocation of the following actions:

- CreateDBInstance
- DeleteDBInstance
- ModifyDBInstance

This data type is used as a response element in the DescribeDBInstances action.

- **DBInstanceIdentifier** (*string*) –

Contains a user-supplied database identifier. This identifier is the unique key that identifies a DB instance.

- **DBInstanceClass** (*string*) –

Contains the name of the compute and memory capacity class of the DB instance.

- **Engine** (*string*) –

Provides the name of the database engine to be used for this

DB instance.

- **DBInstanceStatus** (*string*) –

Specifies the current state of this database.

- **MasterUsername** (*string*) –

Contains the master username for the DB instance.

- **DBName** (*string*) –

The meaning of this parameter differs according to the database engine you use. For example, this value returns either MySQL or PostgreSQL information when returning values from `CreateDBInstanceReadReplica` since Read Replicas are only supported for MySQL and PostgreSQL.

MySQL, SQL Server, PostgreSQL

Contains the name of the initial database of this instance that was provided at create time, if one was specified when the DB instance was created. This same name is returned for the life of the DB instance.

Type: String

Oracle

Contains the Oracle System ID (SID) of the created DB instance. Not shown when the returned parameters do not apply to an Oracle DB instance.

- **Endpoint** (*dict*) –

Specifies the connection endpoint.

- **Address** (*string*) –

Specifies the DNS address of the DB instance.

- **Port** (*integer*) –

Specifies the port that the database engine is listening on.

- **AllocatedStorage** (*integer*) –

Specifies the allocated storage size specified in gigabytes.

- **InstanceCreateTime** (*datetime*) –

Provides the date and time the DB instance was created.

- **PreferredBackupWindow** (*string*) –

Specifies the daily time range during which automated backups are created if automated backups are enabled, as determined by the `BackupRetentionPeriod`.

- **BackupRetentionPeriod** (*integer*) –

Specifies the number of days for which automatic DB snapshots are retained.

- **DBSecurityGroups** (*list*) –

Provides List of DB security group elements containing only `DBSecurityGroup.Name` and `DBSecurityGroup.Status` subelements.

- (*dict*) –

This data type is used as a response element in the following actions:

- `ModifyDBInstance`
- `RebootDBInstance`

- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceToPointInTime`
- **`DBSecurityGroupName`** (*string*) –
The name of the DB security group.
- **`Status`** (*string*) –
The status of the DB security group.
- **`VpcSecurityGroups`** (*list*) –
Provides List of VPC security group elements that the DB instance belongs to.
- (*dict*) –
This data type is used as a response element for queries on VPC security group membership.
- **`VpcSecurityGroupId`** (*string*) –
The name of the VPC security group.
- **`Status`** (*string*) –
The status of the VPC security group.
- **`DBParameterGroups`** (*list*) –
Provides the list of DB parameter groups applied to this DB instance.
- (*dict*) –
The status of the DB parameter group.
This data type is used as a response element in the following actions:
- `CreateDBInstance`
- `CreateDBInstanceReadReplica`
- `DeleteDBInstance`
- `ModifyDBInstance`
- `RebootDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- **`DBParameterGroupName`** (*string*) –
The name of the DP parameter group.
- **`ParameterApplyStatus`** (*string*) –
The status of parameter updates.
- **`AvailabilityZone`** (*string*) –
Specifies the name of the Availability Zone the DB instance is located in.
- **`DBSubnetGroup`** (*dict*) –
Specifies information on the subnet group associated with the DB instance, including the name, description, and subnets in the subnet group.
- **`DBSubnetGroupName`** (*string*) –
Specifies the name of the DB subnet group.
- **`DBSubnetGroupDescription`** (*string*) –
Provides the description of the DB subnet group.
- **`VpcId`** (*string*) –
Provides the VpcId of the DB subnet group.

- **SubnetGroupStatus** (*string*) –
Provides the status of the DB subnet group.
- **Subnets** (*list*) –
Contains a list of Subnet elements.
- (*dict*) –
This data type is used as a response element in the DescribeDBSubnetGroups action.
- **SubnetIdentifier** (*string*) –
Specifies the identifier of the subnet.
- **SubnetAvailabilityZone** (*dict*) –
Contains Availability Zone information.

This data type is used as an element in the following data type:
 - OrderableDBInstanceOption
- **Name** (*string*) –
The name of the availability zone.
- **SubnetStatus** (*string*) –
Specifies the status of the subnet.
- **PreferredMaintenanceWindow** (*string*) –
Specifies the weekly time range during which system maintenance can occur, in Universal Coordinated Time (UTC).
- **PendingModifiedValues** (*dict*) –
Specifies that changes to the DB instance are pending. This element is only included when changes are pending. Specific changes are identified by subelements.
- **DBInstanceClass** (*string*) –
Contains the new DBInstanceClass for the DB instance that will be applied or is in progress.
- **AllocatedStorage** (*integer*) –
Contains the new AllocatedStorage size for the DB instance that will be applied or is in progress.
- **MasterUserPassword** (*string*) –
Contains the pending or in-progress change of the master credentials for the DB instance.
- **Port** (*integer*) –
Specifies the pending port for the DB instance.
- **BackupRetentionPeriod** (*integer*) –
Specifies the pending number of days for which automated backups are retained.
- **MultiAZ** (*boolean*) –
Indicates that the Single-AZ DB instance is to change to a Multi-AZ deployment.
- **EngineVersion** (*string*) –
Indicates the database engine version.
- **Iops** (*integer*) –

Specifies the new Provisioned IOPS value for the DB instance that will be applied or is being applied.

- **DBInstanceIdentifier** (*string*) –

Contains the new `DBInstanceIdentifier` for the DB instance that will be applied or is in progress.

- **StorageType** (*string*) –

Specifies the storage type to be associated with the DB instance.

- **CACertificateIdentifier** (*string*) –

Specifies the identifier of the CA certificate for the DB instance.

- **LatestRestorableTime** (*datetime*) –

Specifies the latest time to which a database can be restored with point-in-time restore.

- **MultiAZ** (*boolean*) –

Specifies if the DB instance is a Multi-AZ deployment.

- **EngineVersion** (*string*) –

Indicates the database engine version.

- **AutoMinorVersionUpgrade** (*boolean*) –

Indicates that minor version patches are applied automatically.

- **ReadReplicaSourceDBInstanceIdentifier** (*string*) –

Contains the identifier of the source DB instance if this DB instance is a Read Replica.

- **ReadReplicaDBInstanceIdentifiers** (*list*) –

Contains one or more identifiers of the Read Replicas associated with this DB instance.

- (*string*) –

- **LicenseModel** (*string*) –

License model information for this DB instance.

- **Iops** (*integer*) –

Specifies the Provisioned IOPS (I/O operations per second) value.

- **OptionGroupMemberships** (*list*) –

Provides the list of option group memberships for this DB instance.

- (*dict*) –

Provides information on the option groups the DB instance is a member of.

- **OptionGroupName** (*string*) –

The name of the option group that the instance belongs to.

- **Status** (*string*) –

The status of the DB instance's option group membership (e.g. in-sync, pending, pending-maintenance, applying).

- **CharacterSetName** (*string*) –

If present, specifies the name of the character set that this instance is associated with.

- **SecondaryAvailabilityZone** (*string*) –

If present, specifies the name of the secondary Availability Zone for a DB instance with multi-AZ support.

- **PubliclyAccessible** (*boolean*) –

Specifies the accessibility options for the DB instance. A value of true specifies an Internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. A value of false specifies an internal instance with a DNS name that resolves to a private IP address.

Default: The default behavior varies depending on whether a VPC has been requested or not. The following list shows the default behavior in each case.

- **Default VPC:** true
- **VPC:** false

If no DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be publicly accessible. If a specific DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be private.

- **StatusInfos** (*list*) –

The status of a Read Replica. If the instance is not a Read Replica, this will be blank.

- (*dict*) –

Provides a list of status information for a DB instance.

- **StatusType** (*string*) –

This value is currently “read replication.”

- **Normal** (*boolean*) –

Boolean value that is true if the instance is operating normally, or false if the instance is in an error state.

- **Status** (*string*) –

Status of the DB instance. For a StatusType of read replica, the values can be replicating, error, stopped, or terminated.

- **Message** (*string*) –

Details of the error if there is an error for the instance. If the instance is not in an error state, this value is blank.

- **StorageType** (*string*) –

Specifies the storage type associated with DB instance.

- **TdeCredentialArn** (*string*) –

The ARN from the Key Store with which the instance is associated for TDE encryption.

- **DbInstancePort** (*integer*) –

Specifies the port that the DB instance listens on. If the DB instance is part of a DB cluster, this can be a different port than the DB cluster port.

- **DBClusterIdentifier** (*string*) –
If the DB instance is a member of a DB cluster, contains the name of the DB cluster that the DB instance is a member of.
- **StorageEncrypted** (*boolean*) –
Specifies whether the DB instance is encrypted.
- **KmsKeyId** (*string*) –
If `StorageEncrypted` is true, the KMS key identifier for the encrypted DB instance.
- **DbiResourceId** (*string*) –
If `StorageEncrypted` is true, the region-unique, immutable identifier for the encrypted DB instance. This identifier is found in AWS CloudTrail log entries whenever the KMS key for the DB instance is accessed.
- **CACertificateIdentifier** (*string*) –
The identifier of the CA certificate for this DB instance.
- **DomainMemberships** (*list*) –
The Active Directory Domain membership records associated with the DB instance.
- (*dict*) –
An Active Directory Domain membership record associated with the DB instance.
- **Domain** (*string*) –
The identifier of the Active Directory Domain.
- **Status** (*string*) –
The status of the DB instance's Active Directory Domain membership (e.g. joined, pending-join, failed etc).
- **Connectivity** (*string*) –
The observed connectivity of the Active Directory Domain.
- **CopyTagsToSnapshot** (*boolean*) –
This property is not currently implemented.

describe_db_log_files (***kwargs*)

Returns a list of DB log files for the DB instance.

Request Syntax

```
response = client.describe_db_log_files(
    DBInstanceIdentifier='string',
    FilenameContains='string',
    FileLastWritten=123,
    FileSize=123,
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxRecords=123,
```

```
Marker='string'  
)
```

Parameters

- **DBInstanceIdentifier** (*string*) – **[REQUIRED]**

The customer-assigned name of the DB instance that contains the log files you want to list.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens
- **FilenameContains** (*string*) – Filters the available log files for log file names that contain the specified string.
- **FileLastWritten** (*integer*) – Filters the available log files for files written since the specified date, in POSIX timestamp format with milliseconds.
- **FileSize** (*integer*) – Filters the available log files for files larger than the specified size.
- **Filters** (*list*) – This parameter is not currently supported.
 - (*dict*) –
 - * **Name** (*string*) – **[REQUIRED]**

This parameter is not currently supported.
 - * **Values** (*list*) – **[REQUIRED]**

This parameter is not currently supported.
 - (*string*) –
- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified MaxRecords value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.
- **Marker** (*string*) – The pagination token provided in the previous request. If this parameter is specified the response includes only records beyond the marker, up to MaxRecords.

Return type dict

Returns

Response Syntax

```
{  
    'DescribeDBLogFiles': [  
        {  
            'LogFileName': 'string',  
            'LastWritten': 123,  
            'Size': 123  
        },  
    ],  
    'Marker': 'string'  
}
```

Response Structure

- (*dict*) –

The response from a call to DescribeDBLogFiles .

- **DescribeDBLogFiles** (*list*) –

The DB log files returned.

- * (*dict*) –
This data type is used as a response element to DescribeDBLogFiles
- **LogFileName** (*string*) –
The name of the log file for the specified DB instance.
- **LastWritten** (*integer*) –
A POSIX timestamp when the last log entry was written.
- **Size** (*integer*) –
The size, in bytes, of the log file for the specified DB instance.
- **Marker** (*string*) –
A pagination token that can be used in a subsequent DescribeDBLogFiles request.

describe_db_parameter_groups (***kwargs*)

Returns a list of DBParameterGroup descriptions. If a DBParameterGroupName is specified, the list will contain only the description of the specified DB parameter group.

Request Syntax

```
response = client.describe_db_parameter_groups(
    DBParameterGroupName='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **DBParameterGroupName** (*string*) – The name of a specific DB parameter group to return details for.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens
- **Filters** (*list*) – This parameter is not currently supported.
 - (*dict*) –
 - * **Name** (*string*) – **[REQUIRED]**
This parameter is not currently supported.
 - * **Values** (*list*) – **[REQUIRED]**
This parameter is not currently supported.
 - (*string*) –
- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified MaxRecords value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: Minimum 20, maximum 100.

- **Marker** (*string*) – An optional pagination token provided by a previous `DescribeDBParameterGroups` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'DBParameterGroups': [
    {
      'DBParameterGroupName': 'string',
      'DBParameterGroupFamily': 'string',
      'Description': 'string'
    },
  ],
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the `DescribeDBParameterGroups` action.

- **Marker** (*string*) –

An optional pagination token provided by a previous request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

- **DBParameterGroups** (*list*) –

A list of `DBParameterGroup` instances.

- * (*dict*) –

Contains the result of a successful invocation of the `CreateDBParameterGroup` action.

This data type is used as a request parameter in the `DeleteDBParameterGroup` action, and as a response element in the `DescribeDBParameterGroups` action.

- **DBParameterGroupName** (*string*) –

Provides the name of the DB parameter group.

- **DBParameterGroupFamily** (*string*) –

Provides the name of the DB parameter group family that this DB parameter group is compatible with.

- **Description** (*string*) –

Provides the customer-specified description for this DB parameter group.

describe_db_parameters (***kwargs*)

Returns the detailed parameter list for a particular DB parameter group.

Request Syntax


```

response = client.describe_db_parameters(
    DBParameterGroupName='string',
    Source='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxRecords=123,
    Marker='string'
)

```

Parameters

- **DBParameterGroupName** (*string*) – [REQUIRED]

The name of a specific DB parameter group to return details for.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **Source** (*string*) – The parameter types to return.

Default: All parameter types returned

Valid Values: user | system | engine-default

- **Filters** (*list*) – This parameter is not currently supported.

– (*dict*) –

- * **Name** (*string*) – [REQUIRED]

This parameter is not currently supported.

- * **Values** (*list*) – [REQUIRED]

This parameter is not currently supported.

· (*string*) –

- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: Minimum 20, maximum 100.

- **Marker** (*string*) – An optional pagination token provided by a previous `DescribeDBParameters` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Return type dict

Returns

Response Syntax

```

{
    'Parameters': [
        {
            'ParameterName': 'string',

```

```
        'ParameterValue': 'string',
        'Description': 'string',
        'Source': 'string',
        'ApplyType': 'string',
        'DataType': 'string',
        'AllowedValues': 'string',
        'IsModifiable': True|False,
        'MinimumEngineVersion': 'string',
        'ApplyMethod': 'immediate'|'pending-reboot'
    },
],
'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the DescribeDBParameters action.

- **Parameters** (*list*) –

A list of Parameter values.

- * (*dict*) –

This data type is used as a request parameter in the ModifyDBParameterGroup and ResetDBParameterGroup actions.

This data type is used as a response element in the DescribeEngineDefaultParameters and DescribeDBParameters actions.

- **ParameterName** (*string*) –

Specifies the name of the parameter.

- **ParameterValue** (*string*) –

Specifies the value of the parameter.

- **Description** (*string*) –

Provides a description of the parameter.

- **Source** (*string*) –

Indicates the source of the parameter value.

- **ApplyType** (*string*) –

Specifies the engine specific parameters type.

- **DataType** (*string*) –

Specifies the valid data type for the parameter.

- **AllowedValues** (*string*) –

Specifies the valid range of values for the parameter.

- **IsModifiable** (*boolean*) –

Indicates whether (`true`) or not (`false`) the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.

- **MinimumEngineVersion** (*string*) –

The earliest engine version to which the parameter can apply.

- **ApplyMethod** (*string*) –

Indicates when to apply parameter updates.

– **Marker** (*string*) –

An optional pagination token provided by a previous request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

describe_db_security_groups (***kwargs*)

Returns a list of `DBSecurityGroup` descriptions. If a `DBSecurityGroupName` is specified, the list will contain only the descriptions of the specified DB security group.

Request Syntax

```
response = client.describe_db_security_groups(
    DBSecurityGroupName='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **DBSecurityGroupName** (*string*) – The name of the DB security group to return details for.
- **Filters** (*list*) – This parameter is not currently supported.
 - (*dict*) –
 - * **Name** (*string*) – [REQUIRED]
This parameter is not currently supported.
 - * **Values** (*list*) – [REQUIRED]
This parameter is not currently supported.
 - (*string*) –
- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.
Default: 100
Constraints: Minimum 20, maximum 100.
- **Marker** (*string*) – An optional pagination token provided by a previous `DescribeDBSecurityGroups` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Return type dict

Returns

Response Syntax

```
{
    'Marker': 'string',
    'DBSecurityGroups': [
        {
```

```
    'OwnerId': 'string',
    'DBSecurityGroupName': 'string',
    'DBSecurityGroupDescription': 'string',
    'VpcId': 'string',
    'EC2SecurityGroups': [
        {
            'Status': 'string',
            'EC2SecurityGroupName': 'string',
            'EC2SecurityGroupId': 'string',
            'EC2SecurityGroupOwnerId': 'string'
        },
    ],
    'IPRanges': [
        {
            'Status': 'string',
            'CIDRIP': 'string'
        },
    ],
]
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the DescribeDBSecurityGroups action.

- **Marker** (*string*) –

An optional pagination token provided by a previous request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

- **DBSecurityGroups** (*list*) –

A list of DBSecurityGroup instances.

- * (*dict*) –

Contains the result of a successful invocation of the following actions:

- DescribeDBSecurityGroups
- AuthorizeDBSecurityGroupIngress
- CreateDBSecurityGroup
- RevokeDBSecurityGroupIngress

This data type is used as a response element in the DescribeDBSecurityGroups action.

- **OwnerId** (*string*) –

Provides the AWS ID of the owner of a specific DB security group.

- **DBSecurityGroupName** (*string*) –

Specifies the name of the DB security group.

- **DBSecurityGroupDescription** (*string*) –

Provides the description of the DB security group.

- **VpcId** (*string*) –

Provides the VpcId of the DB security group.

- **EC2SecurityGroups** (*list*) –

Contains a list of EC2SecurityGroup elements.

- *(dict)* –

This data type is used as a response element in the following actions:

- AuthorizeDBSecurityGroupIngress
- DescribeDBSecurityGroups
- RevokeDBSecurityGroupIngress
- **Status** (*string*) –

Provides the status of the EC2 security group. Status can be “authorizing”, “authorized”, “revoking”, and “revoked”.

- **EC2SecurityGroupName** (*string*) –

Specifies the name of the EC2 security group.

- **EC2SecurityGroupId** (*string*) –

Specifies the id of the EC2 security group.

- **EC2SecurityGroupOwnerId** (*string*) –

Specifies the AWS ID of the owner of the EC2 security group specified in the EC2SecurityGroupName field.

- **IPRanges** (*list*) –

Contains a list of IPRange elements.

- *(dict)* –

This data type is used as a response element in the DescribeDBSecurityGroups action.

- **Status** (*string*) –

Specifies the status of the IP range. Status can be “authorizing”, “authorized”, “revoking”, and “revoked”.

- **CIDRIP** (*string*) –

Specifies the IP range.

describe_db_snapshots (***kwargs*)

Returns information about DB snapshots. This API supports pagination.

Request Syntax

```
response = client.describe_db_snapshots(
    DBInstanceIdentifier='string',
    DBSnapshotIdentifier='string',
    SnapshotType='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **DBInstanceIdentifier** (*string*) – A DB instance identifier to retrieve the list of DB snapshots for. This parameter cannot be used in conjunction with `DBSnapshotIdentifier`. This parameter is not case-sensitive.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
 - First character must be a letter
 - Cannot end with a hyphen or contain two consecutive hyphens
- **DBSnapshotIdentifier** (*string*) – A specific DB snapshot identifier to describe. This parameter cannot be used in conjunction with `DBInstanceIdentifier`. This value is stored as a lowercase string.

Constraints:

- Must be 1 to 255 alphanumeric characters.
 - First character must be a letter.
 - Cannot end with a hyphen or contain two consecutive hyphens.
 - If this is the identifier of an automated snapshot, the `SnapshotType` parameter must also be specified.
- **SnapshotType** (*string*) – The type of snapshots that will be returned. Values can be “automated” or “manual.” If not specified, the returned results will include all snapshots types.
 - **Filters** (*list*) – This parameter is not currently supported.
 - (*dict*) –
 - * **Name** (*string*) – **[REQUIRED]**
This parameter is not currently supported.
 - * **Values** (*list*) – **[REQUIRED]**
This parameter is not currently supported.
 - (*string*) –
 - **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: Minimum 20, maximum 100.

- **Marker** (*string*) – An optional pagination token provided by a previous `DescribeDBSnapshots` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'DBSnapshots': [
    {
      'DBSnapshotIdentifier': 'string',
      'DBInstanceIdentifier': 'string',
      'SnapshotCreateTime': datetime(2015, 1, 1),
      'Engine': 'string',
      'AllocatedStorage': 123,
      'Status': 'string',
      'Port': 123,
```

```

        'AvailabilityZone': 'string',
        'VpcId': 'string',
        'InstanceCreateTime': datetime(2015, 1, 1),
        'MasterUsername': 'string',
        'EngineVersion': 'string',
        'LicenseModel': 'string',
        'SnapshotType': 'string',
        'Iops': 123,
        'OptionGroupName': 'string',
        'PercentProgress': 123,
        'SourceRegion': 'string',
        'SourceDBSnapshotIdentifier': 'string',
        'StorageType': 'string',
        'TdeCredentialArn': 'string',
        'Encrypted': True|False,
        'KmsKeyId': 'string'
    },
]
}

```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the DescribeDBSnapshots action.

- **Marker** (*string*) –

An optional pagination token provided by a previous request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

- **DBSnapshots** (*list*) –

A list of DBSnapshot instances.

- * (*dict*) –

Contains the result of a successful invocation of the following actions:

- CreateDBSnapshot
- DeleteDBSnapshot

This data type is used as a response element in the DescribeDBSnapshots action.

- **DBSnapshotIdentifier** (*string*) –

Specifies the identifier for the DB snapshot.

- **DBInstanceIdentifier** (*string*) –

Specifies the DB instance identifier of the DB instance this DB snapshot was created from.

- **SnapshotCreateTime** (*datetime*) –

Provides the time when the snapshot was taken, in Universal Coordinated Time (UTC).

- **Engine** (*string*) –

Specifies the name of the database engine.

- **AllocatedStorage** (*integer*) –

Specifies the allocated storage size in gigabytes (GB).

- **Status** (*string*) –

Specifies the status of this DB snapshot.

- **Port** (*integer*) –

Specifies the port that the database engine was listening on at the time of the snapshot.

- **AvailabilityZone** (*string*) –

Specifies the name of the Availability Zone the DB instance was located in at the time of the DB snapshot.

- **VpcId** (*string*) –

Provides the VPC ID associated with the DB snapshot.

- **InstanceCreateTime** (*datetime*) –

Specifies the time when the snapshot was taken, in Universal Coordinated Time (UTC).

- **MasterUsername** (*string*) –

Provides the master username for the DB snapshot.

- **EngineVersion** (*string*) –

Specifies the version of the database engine.

- **LicenseModel** (*string*) –

License model information for the restored DB instance.

- **SnapshotType** (*string*) –

Provides the type of the DB snapshot.

- **Iops** (*integer*) –

Specifies the Provisioned IOPS (I/O operations per second) value of the DB instance at the time of the snapshot.

- **OptionGroupName** (*string*) –

Provides the option group name for the DB snapshot.

- **PercentProgress** (*integer*) –

The percentage of the estimated data that has been transferred.

- **SourceRegion** (*string*) –

The region that the DB snapshot was created in or copied from.

- **SourceDBSnapshotIdentifier** (*string*) –

The DB snapshot Arn that the DB snapshot was copied from. It only has value in case of cross customer or cross region copy.

- **StorageType** (*string*) –

Specifies the storage type associated with DB Snapshot.

- **TdeCredentialArn** (*string*) –

The ARN from the Key Store with which to associate the instance for TDE encryption.

- **Encrypted** (*boolean*) –

Specifies whether the DB snapshot is encrypted.

- **KmsKeyId** (*string*) –

If `Encrypted` is true, the KMS key identifier for the encrypted DB snapshot.

describe_db_subnet_groups (***kwargs*)

Returns a list of DBSubnetGroup descriptions. If a DBSubnetGroupName is specified, the list will contain only the descriptions of the specified DBSubnetGroup.

For an overview of CIDR ranges, go to the [Wikipedia Tutorial](#).

Request Syntax

```
response = client.describe_db_subnet_groups(
    DBSubnetGroupName='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **DBSubnetGroupName** (*string*) – The name of the DB subnet group to return details for.
- **Filters** (*list*) – This parameter is not currently supported.
 - (*dict*) –
 - * **Name** (*string*) – **[REQUIRED]**
This parameter is not currently supported.
 - * **Values** (*list*) – **[REQUIRED]**
This parameter is not currently supported.
 - (*string*) –
- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified MaxRecords value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.
Default: 100
Constraints: Minimum 20, maximum 100.
- **Marker** (*string*) – An optional pagination token provided by a previous DescribeDBSubnetGroups request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by MaxRecords.

Return type dict

Returns**Response Syntax**

```
{
    'Marker': 'string',
    'DBSubnetGroups': [
        {
            'DBSubnetGroupName': 'string',
            'DBSubnetGroupDescription': 'string',
            'VpcId': 'string',
            'SubnetGroupStatus': 'string',
```

```
        'Subnets': [
            {
                'SubnetIdentifier': 'string',
                'SubnetAvailabilityZone': {
                    'Name': 'string'
                },
                'SubnetStatus': 'string'
            },
        ]
    },
]
```

Response Structure

- *(dict)* –

Contains the result of a successful invocation of the DescribeDBSubnetGroups action.

- **Marker** (*string*) –

An optional pagination token provided by a previous request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

- **DBSubnetGroups** (*list*) –

A list of DBSubnetGroup instances.

- * *(dict)* –

Contains the result of a successful invocation of the following actions:

- CreateDBSubnetGroup
- ModifyDBSubnetGroup
- DescribeDBSubnetGroups
- DeleteDBSubnetGroup

This data type is used as a response element in the DescribeDBSubnetGroups action.

- **DBSubnetGroupName** (*string*) –

Specifies the name of the DB subnet group.

- **DBSubnetGroupDescription** (*string*) –

Provides the description of the DB subnet group.

- **VpcId** (*string*) –

Provides the VpcId of the DB subnet group.

- **SubnetGroupStatus** (*string*) –

Provides the status of the DB subnet group.

- **Subnets** (*list*) –

Contains a list of Subnet elements.

- *(dict)* –

This data type is used as a response element in the DescribeDBSubnetGroups action.

- **SubnetIdentifier** (*string*) –

Specifies the identifier of the subnet.

- **SubnetAvailabilityZone** (*dict*) –

Contains Availability Zone information.

This data type is used as an element in the following data type:

- **OrderableDBInstanceOption**
- **Name** (*string*) –
The name of the availability zone.
- **SubnetStatus** (*string*) –
Specifies the status of the subnet.

describe_engine_default_cluster_parameters (**kwargs)

Returns the default engine and system parameter information for the cluster database engine.

For more information on Amazon Aurora, see [Aurora on Amazon RDS](#) in the *Amazon RDS User Guide*.

Request Syntax

```
response = client.describe_engine_default_cluster_parameters(
    DBParameterGroupFamily='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **DBParameterGroupFamily** (*string*) – **[REQUIRED]**
The name of the DB cluster parameter group family to return engine parameter information for.
- **Filters** (*list*) – This parameter is not currently supported.
– (*dict*) –
 - * **Name** (*string*) – **[REQUIRED]**
This parameter is not currently supported.
 - * **Values** (*list*) – **[REQUIRED]**
This parameter is not currently supported.
- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.
Default: 100
Constraints: Minimum 20, maximum 100.
- **Marker** (*string*) – An optional pagination token provided by a previous `DescribeEngineDefaultClusterParameters` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Return type dict

Returns

Response Syntax

```
{
  'EngineDefaults': {
    'DBParameterGroupFamily': 'string',
    'Marker': 'string',
    'Parameters': [
      {
        'ParameterName': 'string',
        'ParameterValue': 'string',
        'Description': 'string',
        'Source': 'string',
        'ApplyType': 'string',
        'DataType': 'string',
        'AllowedValues': 'string',
        'IsModifiable': True|False,
        'MinimumEngineVersion': 'string',
        'ApplyMethod': 'immediate'|'pending-reboot'
      },
    ]
  }
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the DescribeEngineDefaultParameters action.

- **EngineDefaults** (*dict*) –

Contains the result of a successful invocation of the DescribeEngineDefaultParameters action.

- * **DBParameterGroupFamily** (*string*) –

Specifies the name of the DB parameter group family that the engine default parameters apply to.

- * **Marker** (*string*) –

An optional pagination token provided by a previous EngineDefaults request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

- * **Parameters** (*list*) –

Contains a list of engine default parameters.

- (*dict*) –

This data type is used as a request parameter in the ModifyDBParameterGroup and ResetDBParameterGroup actions.

This data type is used as a response element in the DescribeEngineDefaultParameters and DescribeDBParameters actions.

- **ParameterName** (*string*) –

Specifies the name of the parameter.

- **ParameterValue** (*string*) –

Specifies the value of the parameter.

- **Description** (*string*) –

Provides a description of the parameter.

- **Source** (*string*) –
Indicates the source of the parameter value.
- **ApplyType** (*string*) –
Specifies the engine specific parameters type.
- **DataType** (*string*) –
Specifies the valid data type for the parameter.
- **AllowedValues** (*string*) –
Specifies the valid range of values for the parameter.
- **IsModifiable** (*boolean*) –
Indicates whether (`true`) or not (`false`) the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.
- **MinimumEngineVersion** (*string*) –
The earliest engine version to which the parameter can apply.
- **ApplyMethod** (*string*) –
Indicates when to apply parameter updates.

describe_engine_default_parameters (***kwargs*)

Returns the default engine and system parameter information for the specified database engine.

Request Syntax

```
response = client.describe_engine_default_parameters(
    DBParameterGroupFamily='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **DBParameterGroupFamily** (*string*) – [REQUIRED]
The name of the DB parameter group family.
- **Filters** (*list*) – Not currently supported.
 - (*dict*) –
 - * **Name** (*string*) – [REQUIRED]
This parameter is not currently supported.
 - * **Values** (*list*) – [REQUIRED]
This parameter is not currently supported.
 - (*string*) –
- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: Minimum 20, maximum 100.

- **Marker** (*string*) – An optional pagination token provided by a previous `DescribeEngineDefaultParameters` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Return type dict

Returns

Response Syntax

```
{
  'EngineDefaults': {
    'DBParameterGroupFamily': 'string',
    'Marker': 'string',
    'Parameters': [
      {
        'ParameterName': 'string',
        'ParameterValue': 'string',
        'Description': 'string',
        'Source': 'string',
        'ApplyType': 'string',
        'DataType': 'string',
        'AllowedValues': 'string',
        'IsModifiable': True|False,
        'MinimumEngineVersion': 'string',
        'ApplyMethod': 'immediate'|'pending-reboot'
      },
    ]
  }
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the `DescribeEngineDefaultParameters` action.

- **EngineDefaults** (*dict*) –

Contains the result of a successful invocation of the `DescribeEngineDefaultParameters` action.

- * **DBParameterGroupFamily** (*string*) –

Specifies the name of the DB parameter group family that the engine default parameters apply to.

- * **Marker** (*string*) –

An optional pagination token provided by a previous `EngineDefaults` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

- * **Parameters** (*list*) –

Contains a list of engine default parameters.

- (*dict*) –

This data type is used as a request parameter in the `ModifyDBParameterGroup` and `ResetDBParameterGroup` actions.

This data type is used as a response element in the `DescribeEngineDefaultParameters` and `DescribeDBParameters`

- actions.
- **ParameterName** (*string*) –
Specifies the name of the parameter.
- **ParameterValue** (*string*) –
Specifies the value of the parameter.
- **Description** (*string*) –
Provides a description of the parameter.
- **Source** (*string*) –
Indicates the source of the parameter value.
- **ApplyType** (*string*) –
Specifies the engine specific parameters type.
- **DataType** (*string*) –
Specifies the valid data type for the parameter.
- **AllowedValues** (*string*) –
Specifies the valid range of values for the parameter.
- **IsModifiable** (*boolean*) –
Indicates whether (`true`) or not (`false`) the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.
- **MinimumEngineVersion** (*string*) –
The earliest engine version to which the parameter can apply.
- **ApplyMethod** (*string*) –
Indicates when to apply parameter updates.

describe_event_categories (**kwargs)

Displays a list of categories for all event source types, or, if specified, for a specified source type. You can see a list of the event categories and source types in the [Events](#) topic in the *Amazon RDS User Guide*.

Request Syntax

```
response = client.describe_event_categories(
    SourceType='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)
```

Parameters

- **SourceType** (*string*) – The type of source that will be generating the events.
Valid values: `db-instance` | `db-parameter-group` | `db-security-group` | `db-snapshot`
- **Filters** (*list*) – This parameter is not currently supported.
 - (*dict*) –
 - * **Name** (*string*) – **[REQUIRED]**
This parameter is not currently supported.

* **Values** (*list*) – [REQUIRED]

This parameter is not currently supported.

· (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'EventCategoriesMapList': [
    {
      'SourceType': 'string',
      'EventCategories': [
        'string',
      ]
    },
  ]
}
```

Response Structure

• (*dict*) –

Data returned from the **DescribeEventCategories** action.

– **EventCategoriesMapList** (*list*) –

A list of EventCategoriesMap data types.

* (*dict*) –

Contains the results of a successful invocation of the DescribeEventCategories action.

· **SourceType** (*string*) –

The source type that the returned categories belong to

· **EventCategories** (*list*) –

The event categories for the specified source type

· (*string*) –

describe_event_subscriptions (***kwargs*)

Lists all the subscription descriptions for a customer account. The description for a subscription includes SubscriptionName, SNSTopicARN, CustomerID, SourceType, SourceID, CreationTime, and Status.

If you specify a SubscriptionName, lists the description for that subscription.

Request Syntax

```
response = client.describe_event_subscriptions(
    SubscriptionName='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxRecords=123,
    Marker='string'
)
```


Parameters

- **SubscriptionName** (*string*) – The name of the RDS event notification subscription you want to describe.
- **Filters** (*list*) – This parameter is not currently supported.
 - (*dict*) –
 - * **Name** (*string*) – **[REQUIRED]**
This parameter is not currently supported.
 - * **Values** (*list*) – **[REQUIRED]**
This parameter is not currently supported.
 - (*string*) –
- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.
Default: 100
Constraints: Minimum 20, maximum 100.
- **Marker** (*string*) – An optional pagination token provided by a previous `DescribeOrderableDBInstanceOptions` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Return type dict**Returns****Response Syntax**

```
{
  'Marker': 'string',
  'EventSubscriptionsList': [
    {
      'CustomerAwsId': 'string',
      'CustSubscriptionId': 'string',
      'SnsTopicArn': 'string',
      'Status': 'string',
      'SubscriptionCreationTime': 'string',
      'SourceType': 'string',
      'SourceIdsList': [
        'string',
      ],
      'EventCategoriesList': [
        'string',
      ],
      'Enabled': True|False
    },
  ],
}
```

Response Structure

- (*dict*) –
Data returned by the **DescribeEventSubscriptions** action.
 - **Marker** (*string*) –
An optional pagination token provided by a previous `DescribeOrderableDBInstanceOptions` request. If this parameter is specified, the response

includes only records beyond the marker, up to the value specified by `MaxRecords`.

– **EventSubscriptionsList** (*list*) –

A list of EventSubscriptions data types.

* (*dict*) –

Contains the results of a successful invocation of the `DescribeEventSubscriptions` action.

· **CustomerAwsId** (*string*) –

The AWS customer account associated with the RDS event notification subscription.

· **CustSubscriptionId** (*string*) –

The RDS event notification subscription Id.

· **SnsTopicArn** (*string*) –

The topic ARN of the RDS event notification subscription.

· **Status** (*string*) –

The status of the RDS event notification subscription.

Constraints:

Can be one of the following: `creating` | `modifying` | `deleting` | `active` | `no-permission` | `topic-not-exist`

The status “no-permission” indicates that RDS no longer has permission to post to the SNS topic. The status “topic-not-exist” indicates that the topic was deleted after the subscription was created.

· **SubscriptionCreationTime** (*string*) –

The time the RDS event notification subscription was created.

· **SourceType** (*string*) –

The source type for the RDS event notification subscription.

· **SourceIdsList** (*list*) –

A list of source IDs for the RDS event notification subscription.

· (*string*) –

· **EventCategoriesList** (*list*) –

A list of event categories for the RDS event notification subscription.

· (*string*) –

· **Enabled** (*boolean*) –

A Boolean value indicating if the subscription is enabled. True indicates the subscription is enabled.

describe_events (***kwargs*)

Returns events related to DB instances, DB security groups, DB snapshots, and DB parameter groups for the past 14 days. Events specific to a particular DB instance, DB security group, database snapshot, or DB parameter group can be obtained by providing the name as a parameter. By default, the past hour of events are returned.

Request Syntax

```

response = client.describe_events(
    SourceIdentifier='string',
    SourceType='db-instance'|'db-parameter-group'|'db-security-group'|'db-snapshot',
    StartTime=datetime(2015, 1, 1),
    EndTime=datetime(2015, 1, 1),
    Duration=123,
    EventCategories=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxRecords=123,
    Marker='string'
)

```

Parameters

- **SourceIdentifier** (*string*) – The identifier of the event source for which events will be returned. If not specified, then all sources are included in the response.

Constraints:

- If SourceIdentifier is supplied, SourceType must also be provided.
- If the source type is DBInstance, then a DBInstanceIdentifier must be supplied.
- If the source type is DBSecurityGroup, a DBSecurityGroupName must be supplied.
- If the source type is DBParameterGroup, a DBParameterGroupName must be supplied.
- If the source type is DBSnapshot, a DBSnapshotIdentifier must be supplied.
- Cannot end with a hyphen or contain two consecutive hyphens.
- **SourceType** (*string*) – The event source to retrieve events for. If no value is specified, all events are returned.
- **StartTime** (*datetime*) – The beginning of the time interval to retrieve events for, specified in ISO 8601 format. For more information about ISO 8601, go to the [ISO8601 Wikipedia page](#).

Example: 2009-07-08T18:00Z

- **EndTime** (*datetime*) – The end of the time interval for which to retrieve events, specified in ISO 8601 format. For more information about ISO 8601, go to the [ISO8601 Wikipedia page](#).

Example: 2009-07-08T18:00Z

- **Duration** (*integer*) – The number of minutes to retrieve events for.

Default: 60

- **EventCategories** (*list*) – A list of event categories that trigger notifications for a event notification subscription.
 - (*string*) –
- **Filters** (*list*) – This parameter is not currently supported.
 - (*dict*) –
 - * **Name** (*string*) – [REQUIRED]

This parameter is not currently supported.

* **Values** (*list*) – [REQUIRED]

This parameter is not currently supported.

· (*string*) –

- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: Minimum 20, maximum 100.

- **Marker** (*string*) – An optional pagination token provided by a previous `DescribeEvents` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'Events': [
    {
      'SourceIdentifier': 'string',
      'SourceType': 'db-instance'|'db-parameter-group'|'db-security-group'|
      'Message': 'string',
      'EventCategories': [
        'string',
      ],
      'Date': datetime(2015, 1, 1)
    },
  ],
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the `DescribeEvents` action.

- **Marker** (*string*) –

An optional pagination token provided by a previous `Events` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

- **Events** (*list*) –

A list of Event instances.

- * (*dict*) –

This data type is used as a response element in the `DescribeEvents` action.

- **SourceIdentifier** (*string*) –

Provides the identifier for the source of the event.

- **SourceType** (*string*) –

Specifies the source type for this event.

- **Message** (*string*) –

Provides the text of this event.

- **EventCategories** (*list*) –
Specifies the category for the event.
- (*string*) –
- **Date** (*datetime*) –
Specifies the date and time of the event.

describe_option_group_options (**kwargs)

Describes all available options.

Request Syntax

```
response = client.describe_option_group_options(
    EngineName='string',
    MajorEngineVersion='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **EngineName** (*string*) – [REQUIRED]
A required parameter. Options available for the given engine name will be described.
- **MajorEngineVersion** (*string*) – If specified, filters the results to include only options for the specified major engine version.
- **Filters** (*list*) – This parameter is not currently supported.
 - (*dict*) –
 - * **Name** (*string*) – [REQUIRED]
This parameter is not currently supported.
 - * **Values** (*list*) – [REQUIRED]
This parameter is not currently supported.
 - (*string*) –
- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.
Default: 100
Constraints: Minimum 20, maximum 100.
- **Marker** (*string*) – An optional pagination token provided by a previous request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Return type dict

Returns

Response Syntax

```
{
  'OptionGroupOptions': [
    {
      'Name': 'string',
      'Description': 'string',
      'EngineName': 'string',
      'MajorEngineVersion': 'string',
      'MinimumRequiredMinorEngineVersion': 'string',
      'PortRequired': True|False,
      'DefaultPort': 123,
      'OptionsDependedOn': [
        'string',
      ],
      'Persistent': True|False,
      'Permanent': True|False,
      'OptionGroupOptionSettings': [
        {
          'SettingName': 'string',
          'SettingDescription': 'string',
          'DefaultValue': 'string',
          'ApplyType': 'string',
          'AllowedValues': 'string',
          'IsModifiable': True|False
        },
      ],
    },
  ],
  'Marker': 'string'
}
```

Response Structure

- (dict) –

- **OptionGroupOptions** (list) –

List of available option group options.

- * (dict) –

Available option.

- **Name** (string) –

The name of the option.

- **Description** (string) –

The description of the option.

- **EngineName** (string) –

The name of the engine that this option can be applied to.

- **MajorEngineVersion** (string) –

Indicates the major engine version that the option is available for.

- **MinimumRequiredMinorEngineVersion** (string) –

The minimum required engine version for the option to be applied.

- **PortRequired** (boolean) –

Specifies whether the option requires a port.

- **DefaultPort** (integer) –

If the option requires a port, specifies the default port for the option.

- **OptionsDependedOn** (*list*) –

List of all options that are prerequisites for this option.

- (*string*) –
- **Persistent** (*boolean*) –

A persistent option cannot be removed from the option group once the option group is used, but this option can be removed from the db instance while modifying the related data and assigning another option group without this option.

- **Permanent** (*boolean*) –

A permanent option cannot be removed from the option group once the option group is used, and it cannot be removed from the db instance after assigning an option group with this permanent option.

- **OptionGroupOptionSettings** (*list*) –

Specifies the option settings that are available (and the default value) for each option in an option group.

- (*dict*) –

Option group option settings are used to display settings available for each option with their default values and other information. These values are used with the DescribeOptionGroupOptions action.

- **SettingName** (*string*) –

The name of the option group option.

- **SettingDescription** (*string*) –

The description of the option group option.

- **DefaultValue** (*string*) –

The default value for the option group option.

- **ApplyType** (*string*) –

The DB engine specific parameter type for the option group option.

- **AllowedValues** (*string*) –

Indicates the acceptable values for the option group option.

- **IsModifiable** (*boolean*) –

Boolean value where true indicates that this option group option can be changed from the default value.

- **Marker** (*string*) –

An optional pagination token provided by a previous request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

describe_option_groups (***kwargs*)

Describes the available option groups.

Request Syntax

```
response = client.describe_option_groups(
    OptionGroupName='string',
```

```
Filters=[
    {
        'Name': 'string',
        'Values': [
            'string',
        ]
    },
],
Marker='string',
MaxRecords=123,
EngineName='string',
MajorEngineVersion='string'
)
```

Parameters

- **OptionGroupName** (*string*) – The name of the option group to describe. Cannot be supplied together with EngineName or MajorEngineVersion.
- **Filters** (*list*) – This parameter is not currently supported.
 - (*dict*) –
 - * **Name** (*string*) – [REQUIRED]
This parameter is not currently supported.
 - * **Values** (*list*) – [REQUIRED]
This parameter is not currently supported.
 - (*string*) –
- **Marker** (*string*) – An optional pagination token provided by a previous DescribeOptionGroups request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by MaxRecords .
- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified MaxRecords value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: Minimum 20, maximum 100.

- **EngineName** (*string*) – Filters the list of option groups to only include groups associated with a specific database engine.
- **MajorEngineVersion** (*string*) – Filters the list of option groups to only include groups associated with a specific database engine version. If specified, then EngineName must also be specified.

Return type dict

Returns

Response Syntax

```
{
    'OptionGroupsList': [
        {
            'OptionGroupName': 'string',
            'OptionGroupDescription': 'string',
            'EngineName': 'string',
            'MajorEngineVersion': 'string',
            'Options': [
                {
                    'OptionName': 'string',
                    'OptionDescription': 'string',
```



```

'Persistent': True|False,
'Permanent': True|False,
'Port': 123,
'OptionSettings': [
    {
        'Name': 'string',
        'Value': 'string',
        'DefaultValue': 'string',
        'Description': 'string',
        'ApplyType': 'string',
        'DataType': 'string',
        'AllowedValues': 'string',
        'IsModifiable': True|False,
        'IsCollection': True|False
    },
],
'DBSecurityGroupMemberships': [
    {
        'DBSecurityGroupName': 'string',
        'Status': 'string'
    },
],
'VpcSecurityGroupMemberships': [
    {
        'VpcSecurityGroupId': 'string',
        'Status': 'string'
    },
],
],
'AllowsVpcAndNonVpcInstanceMemberships': True|False,
'VpcId': 'string'
},
],
'Marker': 'string'
}

```

Response Structure

- (dict) –

List of option groups.

- **OptionGroupsList** (list) –

List of option groups.

- * (dict) –

- **OptionGroupName** (string) –

Specifies the name of the option group.

- **OptionGroupDescription** (string) –

Provides a description of the option group.

- **EngineName** (string) –

Indicates the name of the engine that this option group can be applied to.

- **MajorEngineVersion** (string) –

Indicates the major engine version associated with this option group.

- **Options** (list) –

Indicates what options are available in the option group.

- *(dict)* –

Option details.

- **OptionName** (*string*) –

The name of the option.

- **OptionDescription** (*string*) –

The description of the option.

- **Persistent** (*boolean*) –

Indicate if this option is persistent.

- **Permanent** (*boolean*) –

Indicate if this option is permanent.

- **Port** (*integer*) –

If required, the port configured for this option to use.

- **OptionSettings** (*list*) –

The option settings for this option.

- *(dict)* –

Option settings are the actual settings being applied or configured for that option. It is used when you modify an option group or describe option groups. For example, the `NATIVE_NETWORK_ENCRYPTION` option has a setting called `SQLNET.ENCRYPTION_SERVER` that can have several different values.

- **Name** (*string*) –

The name of the option that has settings that you can set.

- **Value** (*string*) –

The current value of the option setting.

- **DefaultValue** (*string*) –

The default value of the option setting.

- **Description** (*string*) –

The description of the option setting.

- **ApplyType** (*string*) –

The DB engine specific parameter type.

- **DataType** (*string*) –

The data type of the option setting.

- **AllowedValues** (*string*) –

The allowed values of the option setting.

- **IsModifiable** (*boolean*) –

A Boolean value that, when true, indicates the option setting can be modified from the default.

- **IsCollection** (*boolean*) –

Indicates if the option setting is part of a collection.

- **DBSecurityGroupMemberships** (*list*) –

If the option requires access to a port, then this DB security group allows access to the port.

- *(dict)* –

This data type is used as a response element in the following actions:

- `ModifyDBInstance`
- `RebootDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceToPointInTime`
- **`DBSecurityGroupName`** (*string*) –

The name of the DB security group.

- **`Status`** (*string*) –

The status of the DB security group.

- **`VpcSecurityGroupMemberships`** (*list*) –

If the option requires access to a port, then this VPC security group allows access to the port.

- *(dict)* –

This data type is used as a response element for queries on VPC security group membership.

- **`VpcSecurityGroupId`** (*string*) –

The name of the VPC security group.

- **`Status`** (*string*) –

The status of the VPC security group.

- **`AllowsVpcAndNonVpcInstanceMemberships`** (*boolean*) –

Indicates whether this option group can be applied to both VPC and non-VPC instances. The value `true` indicates the option group can be applied to both VPC and non-VPC instances.

- **`VpcId`** (*string*) –

If **`AllowsVpcAndNonVpcInstanceMemberships`** is `false`, this field is blank. If **`AllowsVpcAndNonVpcInstanceMemberships`** is `true` and this field is blank, then this option group can be applied to both VPC and non-VPC instances. If this field contains a value, then this option group can only be applied to instances that are in the VPC indicated by this field.

- **`Marker`** (*string*) –

An optional pagination token provided by a previous request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

`describe_orderable_db_instance_options` (***kwargs*)

Returns a list of orderable DB instance options for the specified engine.

Request Syntax

```
response = client.describe_orderable_db_instance_options(
    Engine='string',
    EngineVersion='string',
    DBInstanceClass='string',
    LicenseModel='string',
    Vpc=True|False,
    Filters=[
```

```
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **Engine** (*string*) – [REQUIRED]

The name of the engine to retrieve DB instance options for.

- **EngineVersion** (*string*) – The engine version filter value. Specify this parameter to show only the available offerings matching the specified engine version.
- **DBInstanceClass** (*string*) – The DB instance class filter value. Specify this parameter to show only the available offerings matching the specified DB instance class.
- **LicenseModel** (*string*) – The license model filter value. Specify this parameter to show only the available offerings matching the specified license model.
- **Vpc** (*boolean*) – The VPC filter value. Specify this parameter to show only the available VPC or non-VPC offerings.
- **Filters** (*list*) – This parameter is not currently supported.

- (*dict*) –

- * **Name** (*string*) – [REQUIRED]

- This parameter is not currently supported.

- * **Values** (*list*) – [REQUIRED]

- This parameter is not currently supported.

- (*string*) –

- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: Minimum 20, maximum 100.

- **Marker** (*string*) – An optional pagination token provided by a previous `DescribeOrderableDBInstanceOptions` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Return type dict

Returns

Response Syntax

```
{
    'OrderableDBInstanceOptions': [
        {
            'Engine': 'string',
            'EngineVersion': 'string',
            'DBInstanceClass': 'string',
            'LicenseModel': 'string',
            'AvailabilityZones': [
```

```

        {
            'Name': 'string'
        },
    ],
    'MultiAZCapable': True|False,
    'ReadReplicaCapable': True|False,
    'Vpc': True|False,
    'SupportsStorageEncryption': True|False,
    'StorageType': 'string',
    'SupportsIops': True|False
},
],
'Marker': 'string'
}

```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the DescribeOrderableDBInstanceOptions action.

- **OrderableDBInstanceOptions** (*list*) –

An OrderableDBInstanceOption structure containing information about orderable options for the DB instance.

- * (*dict*) –

Contains a list of available options for a DB instance

This data type is used as a response element in the DescribeOrderableDBInstanceOptions action.

- **Engine** (*string*) –

The engine type of the orderable DB instance.

- **EngineVersion** (*string*) –

The engine version of the orderable DB instance.

- **DBInstanceClass** (*string*) –

The DB instance class for the orderable DB instance.

- **LicenseModel** (*string*) –

The license model for the orderable DB instance.

- **AvailabilityZones** (*list*) –

A list of Availability Zones for the orderable DB instance.

- (*dict*) –

Contains Availability Zone information.

This data type is used as an element in the following data type:

- OrderableDBInstanceOption

- **Name** (*string*) –

The name of the availability zone.

- **MultiAZCapable** (*boolean*) –

Indicates whether this orderable DB instance is multi-AZ capable.

- **ReadReplicaCapable** (*boolean*) –

Indicates whether this orderable DB instance can have a Read Replica.

- **Vpc** (*boolean*) –
Indicates whether this is a VPC orderable DB instance.
- **SupportsStorageEncryption** (*boolean*) –
Indicates whether this orderable DB instance supports encrypted storage.
- **StorageType** (*string*) –
Indicates the storage type for this orderable DB instance.
- **SupportsIops** (*boolean*) –
Indicates whether this orderable DB instance supports provisioned IOPS.
- **Marker** (*string*) –
An optional pagination token provided by a previous `OrderableDBInstanceOptions` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

describe_pending_maintenance_actions (***kwargs*)

Returns a list of resources (for example, DB instances) that have at least one pending maintenance action.

Request Syntax

```
response = client.describe_pending_maintenance_actions(
    ResourceIdentifier='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    Marker='string',
    MaxRecords=123
)
```

Parameters

- **ResourceIdentifier** (*string*) – The ARN of a resource to return pending maintenance actions for.
- **Filters** (*list*) – A filter that specifies one or more resources to return pending maintenance actions for.

Supported filters:

- `db-instance-id` - Accepts DB instance identifiers and DB instance Amazon Resource Names (ARNs). The results list will only include pending maintenance actions for the DB instances identified by these ARNs.
- (*dict*) –
 - * **Name** (*string*) – **[REQUIRED]**
This parameter is not currently supported.
 - * **Values** (*list*) – **[REQUIRED]**
This parameter is not currently supported.
- (*string*) –

- **Marker** (*string*) – An optional pagination token provided by a previous `DescribePendingMaintenanceActions` request. If this parameter is specified, the response includes only records beyond the marker, up to a number of records specified by `MaxRecords`.
- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: Minimum 20, maximum 100.

Return type dict

Returns

Response Syntax

```
{
  'PendingMaintenanceActions': [
    {
      'ResourceIdentifier': 'string',
      'PendingMaintenanceActionDetails': [
        {
          'Action': 'string',
          'AutoAppliedAfterDate': datetime(2015, 1, 1),
          'ForcedApplyDate': datetime(2015, 1, 1),
          'OptInStatus': 'string',
          'CurrentApplyDate': datetime(2015, 1, 1),
          'Description': 'string'
        },
      ],
    },
  ],
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Data returned from the **DescribePendingMaintenanceActions** action.

- **PendingMaintenanceActions** (*list*) –

A list of the pending maintenance actions for the resource.

- * (*dict*) –

Describes the pending maintenance actions for a resource.

- **ResourceIdentifier** (*string*) –

The ARN of the resource that has pending maintenance actions.

- **PendingMaintenanceActionDetails** (*list*) –

A list that provides details about the pending maintenance actions for the resource.

- (*dict*) –

Provides information about a pending maintenance action for a resource.

- **Action** (*string*) –

The type of pending maintenance action that is available for the resource.

- **AutoAppliedAfterDate** (*datetime*) –

The date of the maintenance window when the action will be applied. The maintenance action will be applied to the resource during its first maintenance window after this date. If this date is specified, any `next-maintenance` opt-in requests are ignored.

- **ForcedApplyDate** (*datetime*) –

The date when the maintenance action will be automatically applied. The maintenance action will be applied to the resource on this date regardless of the maintenance window for the resource. If this date is specified, any `immediate` opt-in requests are ignored.

- **OptInStatus** (*string*) –

Indicates the type of opt-in request that has been received for the resource.

- **CurrentApplyDate** (*datetime*) –

The effective date when the pending maintenance action will be applied to the resource. This date takes into account opt-in requests received from the `ApplyPendingMaintenanceAction` API, the `AutoAppliedAfterDate`, and the `ForcedApplyDate`. This value is blank if an opt-in request has not been received and nothing has been specified as `AutoAppliedAfterDate` or `ForcedApplyDate`.

- **Description** (*string*) –

A description providing more detail about the maintenance action.

- **Marker** (*string*) –

An optional pagination token provided by a previous `DescribePendingMaintenanceActions` request. If this parameter is specified, the response includes only records beyond the marker, up to a number of records specified by `MaxRecords`.

describe_reserved_db_instances (***kwargs*)

Returns information about reserved DB instances for this account, or about a specified reserved DB instance.

Request Syntax

```
response = client.describe_reserved_db_instances(
    ReservedDBInstanceId='string',
    ReservedDBInstancesOfferingId='string',
    DBInstanceClass='string',
    Duration='string',
    ProductDescription='string',
    OfferingType='string',
    MultiAZ=True|False,
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
```



```

    ],
    },
],
MaxRecords=123,
Marker='string'
)

```

Parameters

- **ReservedDBInstanceId** (*string*) – The reserved DB instance identifier filter value. Specify this parameter to show only the reservation that matches the specified reservation ID.
- **ReservedDBInstancesOfferingId** (*string*) – The offering identifier filter value. Specify this parameter to show only purchased reservations matching the specified offering identifier.
- **DBInstanceClass** (*string*) – The DB instance class filter value. Specify this parameter to show only those reservations matching the specified DB instances class.
- **Duration** (*string*) – The duration filter value, specified in years or seconds. Specify this parameter to show only reservations for this duration.

Valid Values: 1 | 3 | 31536000 | 94608000

- **ProductDescription** (*string*) – The product description filter value. Specify this parameter to show only those reservations matching the specified product description.
- **OfferingType** (*string*) – The offering type filter value. Specify this parameter to show only the available offerings matching the specified offering type.

Valid Values: "Partial Upfront" | "All Upfront" | "No Upfront"

- **MultiAZ** (*boolean*) – The Multi-AZ filter value. Specify this parameter to show only those reservations matching the specified Multi-AZ parameter.
- **Filters** (*list*) – This parameter is not currently supported.

– (*dict*) –

* **Name** (*string*) – [REQUIRED]

This parameter is not currently supported.

* **Values** (*list*) – [REQUIRED]

This parameter is not currently supported.

· (*string*) –

- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more than the `MaxRecords` value is available, a pagination token called a marker is included in the response so that the following results can be retrieved.

Default: 100

Constraints: Minimum 20, maximum 100.

- **Marker** (*string*) – An optional pagination token provided by a previous request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'ReservedDBInstances': [
    {
      'ReservedDBInstanceId': 'string',
      'ReservedDBInstancesOfferingId': 'string',
      'DBInstanceClass': 'string',
      'StartTime': datetime(2015, 1, 1),
      'Duration': 123,
      'FixedPrice': 123.0,
      'UsagePrice': 123.0,
      'CurrencyCode': 'string',
      'DBInstanceCount': 123,
      'ProductDescription': 'string',
      'OfferingType': 'string',
      'MultiAZ': True|False,
      'State': 'string',
      'RecurringCharges': [
        {
          'RecurringChargeAmount': 123.0,
          'RecurringChargeFrequency': 'string'
        },
      ],
    },
  ],
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the DescribeReservedDBInstances action.

- **Marker** (*string*) –

An optional pagination token provided by a previous request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

- **ReservedDBInstances** (*list*) –

A list of reserved DB instances.

- * (*dict*) –

This data type is used as a response element in the DescribeReservedDBInstances and PurchaseReservedDBInstancesOffering actions.

- **ReservedDBInstanceId** (*string*) –
The unique identifier for the reservation.
- **ReservedDBInstancesOfferingId** (*string*) –
The offering identifier.
- **DBInstanceClass** (*string*) –
The DB instance class for the reserved DB instance.
- **StartTime** (*datetime*) –
The time the reservation started.
- **Duration** (*integer*) –
The duration of the reservation in seconds.

- **FixedPrice** (*float*) –
The fixed price charged for this reserved DB instance.
- **UsagePrice** (*float*) –
The hourly price charged for this reserved DB instance.
- **CurrencyCode** (*string*) –
The currency code for the reserved DB instance.
- **DBInstanceCount** (*integer*) –
The number of reserved DB instances.
- **ProductDescription** (*string*) –
The description of the reserved DB instance.
- **OfferingType** (*string*) –
The offering type of this reserved DB instance.
- **MultiAZ** (*boolean*) –
Indicates if the reservation applies to Multi-AZ deployments.
- **State** (*string*) –
The state of the reserved DB instance.
- **RecurringCharges** (*list*) –
The recurring price charged to run this reserved DB instance.
- (*dict*) –
This data type is used as a response element in the DescribeReservedDBInstances and DescribeReservedDBInstancesOfferings actions.
- **RecurringChargeAmount** (*float*) –
The amount of the recurring charge.
- **RecurringChargeFrequency** (*string*) –
The frequency of the recurring charge.

describe_reserved_db_instances_offerings (***kwargs*)

Lists available reserved DB instance offerings.

Request Syntax

```
response = client.describe_reserved_db_instances_offerings(
    ReservedDBInstancesOfferingId='string',
    DBInstanceClass='string',
    Duration='string',
    ProductDescription='string',
    OfferingType='string',
    MultiAZ=True|False,
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **ReservedDBInstancesOfferingId** (*string*) – The offering identifier filter value. Specify this parameter to show only the available offering that matches the specified reservation identifier.

Example: 438012d3-4052-4cc7-b2e3-8d3372e0e706

- **DBInstanceClass** (*string*) – The DB instance class filter value. Specify this parameter to show only the available offerings matching the specified DB instance class.
- **Duration** (*string*) – Duration filter value, specified in years or seconds. Specify this parameter to show only reservations for this duration.

Valid Values: 1 | 3 | 31536000 | 94608000

- **ProductDescription** (*string*) – Product description filter value. Specify this parameter to show only the available offerings matching the specified product description.
- **OfferingType** (*string*) – The offering type filter value. Specify this parameter to show only the available offerings matching the specified offering type.

Valid Values: "Partial Upfront" | "All Upfront" | "No Upfront"

- **MultiAZ** (*boolean*) – The Multi-AZ filter value. Specify this parameter to show only the available offerings matching the specified Multi-AZ parameter.
- **Filters** (*list*) – This parameter is not currently supported.

– (*dict*) –

* **Name** (*string*) – [REQUIRED]

This parameter is not currently supported.

* **Values** (*list*) – [REQUIRED]

This parameter is not currently supported.

· (*string*) –

- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more than the `MaxRecords` value is available, a pagination token called a marker is included in the response so that the following results can be retrieved.

Default: 100

Constraints: Minimum 20, maximum 100.

- **Marker** (*string*) – An optional pagination token provided by a previous request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'ReservedDBInstancesOfferings': [
    {
      'ReservedDBInstancesOfferingId': 'string',
      'DBInstanceClass': 'string',
      'Duration': 123,
      'FixedPrice': 123.0,
      'UsagePrice': 123.0,
      'CurrencyCode': 'string',
```

```

        'ProductDescription': 'string',
        'OfferingType': 'string',
        'MultiAZ': True|False,
        'RecurringCharges': [
            {
                'RecurringChargeAmount': 123.0,
                'RecurringChargeFrequency': 'string'
            },
        ]
    },
]
}

```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the `DescribeReservedDBInstancesOfferings` action.

- **Marker** (*string*) –

An optional pagination token provided by a previous request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

- **ReservedDBInstancesOfferings** (*list*) –

A list of reserved DB instance offerings.

- * (*dict*) –

This data type is used as a response element in the `DescribeReservedDBInstancesOfferings` action.

- **ReservedDBInstancesOfferingId** (*string*) –

The offering identifier.

- **DBInstanceClass** (*string*) –

The DB instance class for the reserved DB instance.

- **Duration** (*integer*) –

The duration of the offering in seconds.

- **FixedPrice** (*float*) –

The fixed price charged for this offering.

- **UsagePrice** (*float*) –

The hourly price charged for this offering.

- **CurrencyCode** (*string*) –

The currency code for the reserved DB instance offering.

- **ProductDescription** (*string*) –

The database engine used by the offering.

- **OfferingType** (*string*) –

The offering type.

- **MultiAZ** (*boolean*) –

Indicates if the offering applies to Multi-AZ deployments.

- **RecurringCharges** (*list*) –

The recurring price charged to run this reserved DB instance.

- (*dict*) –

This data type is used as a response element in the DescribeReservedDBInstances and DescribeReservedDBInstancesOfferings actions.

- **RecurringChargeAmount** (*float*) –

The amount of the recurring charge.

- **RecurringChargeFrequency** (*string*) –

The frequency of the recurring charge.

download_db_log_file_portion (***kwargs*)

Downloads all or a portion of the specified log file, up to 1 MB in size.

Request Syntax

```
response = client.download_db_log_file_portion(  
    DBInstanceIdentifier='string',  
    LogFileName='string',  
    Marker='string',  
    NumberOfLines=123  
)
```

Parameters

- **DBInstanceIdentifier** (*string*) – [REQUIRED]

The customer-assigned name of the DB instance that contains the log files you want to list.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **LogFileName** (*string*) – [REQUIRED]

The name of the log file to be downloaded.

- **Marker** (*string*) – The pagination token provided in the previous request or “0”. If the Marker parameter is specified the response includes only records beyond the marker until the end of the file or up to NumberOfLines.
- **NumberOfLines** (*integer*) – The number of lines to download. If the number of lines specified results in a file over 1 MB in size, the file will be truncated at 1 MB in size.

If the NumberOfLines parameter is specified, then the block of lines returned can be from the beginning or the end of the log file, depending on the value of the Marker parameter.

- If neither Marker or NumberOfLines are specified, the entire log file is returned.
- If NumberOfLines is specified and Marker is not specified, then the most recent lines from the end of the log file are returned.
- If Marker is specified as “0”, then the specified number of lines from the beginning of the log file are returned.
- You can download the log file in blocks of lines by specifying the size of the block using the NumberOfLines parameter, and by specifying a value of “0” for the Marker parameter in your first request. Include the Marker value returned in the response as the Marker value for the next request, continuing until the AdditionalDataPending response element returns false.

Return type dict

Returns

Response Syntax

```
{
    'LogFileData': 'string',
    'Marker': 'string',
    'AdditionalDataPending': True|False
}
```

Response Structure

- (*dict*) –

This data type is used as a response element to DownloadDBLogFilePortion .

- **LogFileData** (*string*) –

Entries from the specified log file.

- **Marker** (*string*) –

A pagination token that can be used in a subsequent DownloadDBLogFilePortion request.

- **AdditionalDataPending** (*boolean*) –

Boolean value that if true, indicates there is more data to be downloaded.

failover_db_cluster (***kwargs*)

Forces a failover for a DB cluster.

A failover for a DB cluster promotes one of the read-only instances in the DB cluster to the master DB instance (the cluster writer) and deletes the current primary instance.

Amazon Aurora will automatically fail over to a read-only instance, if one exists, when the primary instance fails. You can force a failover when you want to simulate a failure of a DB instance for testing. Because each instance in a DB cluster has its own endpoint address, you will need to clean up and re-establish any existing connections that use those endpoint addresses when the failover is complete.

For more information on Amazon Aurora, see [Aurora on Amazon RDS](#) in the *Amazon RDS User Guide*.

Request Syntax

```
response = client.failover_db_cluster(
    DBClusterIdentifier='string'
)
```

Parameters **DBClusterIdentifier** (*string*) – A DB cluster identifier to force a failover for.

This parameter is not case-sensitive.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Return type dict

Returns

Response Syntax

```
{
    'DBCluster': {
        'AllocatedStorage': 123,
        'AvailabilityZones': [
            'string',

```

```
],
'BackupRetentionPeriod': 123,
'CharacterSetName': 'string',
'DatabaseName': 'string',
'DBClusterIdentifier': 'string',
'DBClusterParameterGroup': 'string',
'DBSubnetGroup': 'string',
'Status': 'string',
'PercentProgress': 'string',
'EarliestRestorableTime': datetime(2015, 1, 1),
'Endpoint': 'string',
'Engine': 'string',
'EngineVersion': 'string',
'LatestRestorableTime': datetime(2015, 1, 1),
'Port': 123,
'MasterUsername': 'string',
'DBClusterOptionGroupMemberships': [
    {
        'DBClusterOptionGroupName': 'string',
        'Status': 'string'
    },
],
'PreferredBackupWindow': 'string',
'PreferredMaintenanceWindow': 'string',
'DBClusterMembers': [
    {
        'DBInstanceIdentifier': 'string',
        'IsClusterWriter': True|False,
        'DBClusterParameterGroupStatus': 'string'
    },
],
'VpcSecurityGroups': [
    {
        'VpcSecurityGroupId': 'string',
        'Status': 'string'
    },
],
]
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the following actions:

- CreateDBCluster
- DeleteDBCluster
- FailoverDBCluster
- ModifyDBCluster
- RestoreDBClusterFromSnapshot

This data type is used as a response element in the DescribeDBClusters action.

- **DBCluster** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * CreateDBCluster
- * DeleteDBCluster
- * FailoverDBCluster
- * ModifyDBCluster
- * RestoreDBClusterFromSnapshot

This data type is used as a response element in the DescribeDBClusters action.

- * **AllocatedStorage** (*integer*) –
Specifies the allocated storage size in gigabytes (GB).
- * **AvailabilityZones** (*list*) –
Provides the list of EC2 Availability Zones that instances in the DB cluster can be created in.
· (*string*) –
- * **BackupRetentionPeriod** (*integer*) –
Specifies the number of days for which automatic DB snapshots are retained.
- * **CharacterSetName** (*string*) –
If present, specifies the name of the character set that this cluster is associated with.
- * **DatabaseName** (*string*) –
Contains the name of the initial database of this DB cluster that was provided at create time, if one was specified when the DB cluster was created. This same name is returned for the life of the DB cluster.
- * **DBClusterIdentifier** (*string*) –
Contains a user-supplied DB cluster identifier. This identifier is the unique key that identifies a DB cluster.
- * **DBClusterParameterGroup** (*string*) –
Specifies the name of the DB cluster parameter group for the DB cluster.
- * **DBSubnetGroup** (*string*) –
Specifies information on the subnet group associated with the DB cluster, including the name, description, and subnets in the subnet group.
- * **Status** (*string*) –
Specifies the current state of this DB cluster.
- * **PercentProgress** (*string*) –
Specifies the progress of the operation as a percentage.
- * **EarliestRestorableTime** (*datetime*) –
Specifies the earliest time to which a database can be restored with point-in-time restore.
- * **Endpoint** (*string*) –
Specifies the connection endpoint for the primary instance of the DB cluster.
- * **Engine** (*string*) –
Provides the name of the database engine to be used for this DB cluster.
- * **EngineVersion** (*string*) –
Indicates the database engine version.
- * **LatestRestorableTime** (*datetime*) –

Specifies the latest time to which a database can be restored with point-in-time restore.

* **Port** (*integer*) –

Specifies the port that the database engine is listening on.

* **MasterUsername** (*string*) –

Contains the master username for the DB cluster.

* **DBClusterOptionGroupMemberships** (*list*) –

Provides the list of option group memberships for this DB cluster.

· (*dict*) –

Contains status information for a DB cluster option group.

· **DBClusterOptionGroupName** (*string*) –

Specifies the name of the DB cluster option group.

· **Status** (*string*) –

Specifies the status of the DB cluster option group.

* **PreferredBackupWindow** (*string*) –

Specifies the daily time range during which automated backups are created if automated backups are enabled, as determined by the `BackupRetentionPeriod`.

* **PreferredMaintenanceWindow** (*string*) –

Specifies the weekly time range during which system maintenance can occur, in Universal Coordinated Time (UTC).

* **DBClusterMembers** (*list*) –

Provides the list of instances that make up the DB cluster.

· (*dict*) –

Contains information about an instance that is part of a DB cluster.

· **DBInstanceIdentifier** (*string*) –

Specifies the instance identifier for this member of the DB cluster.

· **IsClusterWriter** (*boolean*) –

Value that is `true` if the cluster member is the primary instance for the DB cluster and `false` otherwise.

· **DBClusterParameterGroupStatus** (*string*) –

Specifies the status of the DB cluster parameter group for this member of the DB cluster.

* **VpcSecurityGroups** (*list*) –

Provides a list of VPC security groups that the DB cluster belongs to.

· (*dict*) –

This data type is used as a response element for queries on VPC security group membership.

· **VpcSecurityGroupId** (*string*) –

The name of the VPC security group.

· **Status** (*string*) –

The status of the VPC security group.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_waiter (*waiter_name*)

list_tags_for_resource (***kwargs*)

Lists all tags on an Amazon RDS resource.

For an overview on tagging an Amazon RDS resource, see [Tagging Amazon RDS Resources](#).

Request Syntax

```
response = client.list_tags_for_resource(
    ResourceName='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ]
)
```

Parameters

- **ResourceName** (*string*) – **[REQUIRED]**

The Amazon RDS resource with tags to be listed. This value is an Amazon Resource Name (ARN). For information about creating an ARN, see [Constructing an RDS Amazon Resource Name \(ARN\)](#).

- **Filters** (*list*) – This parameter is not currently supported.

– (*dict*) –

- * **Name** (*string*) – **[REQUIRED]**

This parameter is not currently supported.

- * **Values** (*list*) – **[REQUIRED]**

This parameter is not currently supported.

· (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'TagList': [
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
}
```

Response Structure

- (*dict*) –
 - **TagList** (*list*) –

List of tags returned by the ListTagsForResource operation.

- * (*dict*) –

Metadata assigned to an Amazon RDS resource consisting of a key-value pair.

- **Key** (*string*) –

A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

- **Value** (*string*) –

A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

modify_db_cluster (***kwargs*)

Modify a setting for an Amazon Aurora DB cluster. You can change one or more database configuration parameters by specifying these parameters and the new values in the request. For more information on Amazon Aurora, see [Aurora on Amazon RDS](#) in the *Amazon RDS User Guide*.

Request Syntax

```
response = client.modify_db_cluster(
    DBClusterIdentifier='string',
    NewDBClusterIdentifier='string',
    ApplyImmediately=True|False,
    BackupRetentionPeriod=123,
    DBClusterParameterGroupName='string',
    VpcSecurityGroupIds=[
        'string',
    ],
    Port=123,
    MasterUserPassword='string',
    OptionGroupName='string',
    PreferredBackupWindow='string',
```

```
PreferredMaintenanceWindow='string'
)
```

Parameters

- **DBClusterIdentifier** (*string*) – The DB cluster identifier for the cluster being modified. This parameter is not case-sensitive.

Constraints:

- Must be the identifier for an existing DB cluster.
- Must contain from 1 to 63 alphanumeric characters or hyphens.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.

- **NewDBClusterIdentifier** (*string*) – The new DB cluster identifier for the DB cluster when renaming a DB cluster. This value is stored as a lowercase string.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Example: my-cluster2

- **ApplyImmediately** (*boolean*) – A value that specifies whether the modifications in this request and any pending modifications are asynchronously applied as soon as possible, regardless of the `PreferredMaintenanceWindow` setting for the DB cluster.

If this parameter is set to `false`, changes to the DB cluster are applied during the next maintenance window.

Default: `false`

- **BackupRetentionPeriod** (*integer*) – The number of days for which automated backups are retained. Setting this parameter to a positive number enables backups. Setting this parameter to 0 disables automated backups.

Default: 1

Constraints:

- Must be a value from 0 to 35

- **DBClusterParameterGroupName** (*string*) – The name of the DB cluster parameter group to use for the DB cluster.
- **VpcSecurityGroupIds** (*list*) – A list of VPC security groups that the DB cluster will belong to.
 - (*string*) –
- **Port** (*integer*) – The port number on which the DB cluster accepts connections.

Constraints: Value must be 1150–65535

Default: The same port as the original DB cluster.

- **MasterUserPassword** (*string*) – The new password for the master database user. This password can contain any printable ASCII character except “/”, “””, or “@”.

Constraints: Must contain from 8 to 41 characters.

- **OptionGroupName** (*string*) – A value that indicates that the DB cluster should be associated with the specified option group. Changing this parameter does not result in an outage except in the following case, and the change is applied during the next maintenance window unless the `ApplyImmediately` parameter is set to `true` for this request. If the parameter change results in an option group that enables OEM, this change can cause a brief (sub-second) period dur-

ing which new connections are rejected but existing connections are not interrupted.

Permanent options cannot be removed from an option group. The option group cannot be removed from a DB cluster once it is associated with a DB cluster.

- **PreferredBackupWindow** (*string*) – The daily time range during which automated backups are created if automated backups are enabled, using the `BackupRetentionPeriod` parameter.

Default: A 30-minute window selected at random from an 8-hour block of time per region. To see the time blocks available, see [Adjusting the Preferred Maintenance Window](#) in the *Amazon RDS User Guide*.

Constraints:

- Must be in the format `hh24:mi-hh24:mi`.
 - Times should be in Universal Coordinated Time (UTC).
 - Must not conflict with the preferred maintenance window.
 - Must be at least 30 minutes.
- **PreferredMaintenanceWindow** (*string*) – The weekly time range during which system maintenance can occur, in Universal Coordinated Time (UTC).

Format: `ddd:hh24:mi-ddd:hh24:mi`

Default: A 30-minute window selected at random from an 8-hour block of time per region, occurring on a random day of the week. To see the time blocks available, see [Adjusting the Preferred Maintenance Window](#) in the *Amazon RDS User Guide*.

Valid Days: Mon, Tue, Wed, Thu, Fri, Sat, Sun

Constraints: Minimum 30-minute window.

Return type dict

Returns

Response Syntax

```
{
  'DBCluster': {
    'AllocatedStorage': 123,
    'AvailabilityZones': [
      'string',
    ],
    'BackupRetentionPeriod': 123,
    'CharacterSetName': 'string',
    'DatabaseName': 'string',
    'DBClusterIdentifier': 'string',
    'DBClusterParameterGroup': 'string',
    'DBSubnetGroup': 'string',
    'Status': 'string',
    'PercentProgress': 'string',
    'EarliestRestorableTime': datetime(2015, 1, 1),
    'Endpoint': 'string',
    'Engine': 'string',
    'EngineVersion': 'string',
    'LatestRestorableTime': datetime(2015, 1, 1),
    'Port': 123,
    'MasterUsername': 'string',
    'DBClusterOptionGroupMemberships': [
      {
        'DBClusterOptionGroupName': 'string',
```

```

        'Status': 'string'
    },
],
'PreferredBackupWindow': 'string',
'PreferredMaintenanceWindow': 'string',
'DBClusterMembers': [
    {
        'DBInstanceIdentifier': 'string',
        'IsClusterWriter': True|False,
        'DBClusterParameterGroupStatus': 'string'
    },
],
'VpcSecurityGroups': [
    {
        'VpcSecurityGroupId': 'string',
        'Status': 'string'
    },
],
]
}

```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the following actions:

- CreateDBCluster
- DeleteDBCluster
- FailoverDBCluster
- ModifyDBCluster
- RestoreDBClusterFromSnapshot

This data type is used as a response element in the DescribeDBClusters action.

- **DBCluster** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * CreateDBCluster
- * DeleteDBCluster
- * FailoverDBCluster
- * ModifyDBCluster
- * RestoreDBClusterFromSnapshot

This data type is used as a response element in the DescribeDBClusters action.

- * **AllocatedStorage** (*integer*) –

Specifies the allocated storage size in gigabytes (GB).

- * **AvailabilityZones** (*list*) –

Provides the list of EC2 Availability Zones that instances in the DB cluster can be created in.

· (*string*) –

- * **BackupRetentionPeriod** (*integer*) –

Specifies the number of days for which automatic DB snapshots are retained.

- * **CharacterSetName** (*string*) –

If present, specifies the name of the character set that this cluster is associated with.

- * **DatabaseName** (*string*) –

Contains the name of the initial database of this DB cluster that was provided at create time, if one was specified when the DB cluster was created. This same name is returned for the life of the DB cluster.

* **DBClusterIdentifier** (*string*) –

Contains a user-supplied DB cluster identifier. This identifier is the unique key that identifies a DB cluster.

* **DBClusterParameterGroup** (*string*) –

Specifies the name of the DB cluster parameter group for the DB cluster.

* **DBSubnetGroup** (*string*) –

Specifies information on the subnet group associated with the DB cluster, including the name, description, and subnets in the subnet group.

* **Status** (*string*) –

Specifies the current state of this DB cluster.

* **PercentProgress** (*string*) –

Specifies the progress of the operation as a percentage.

* **EarliestRestorableTime** (*datetime*) –

Specifies the earliest time to which a database can be restored with point-in-time restore.

* **Endpoint** (*string*) –

Specifies the connection endpoint for the primary instance of the DB cluster.

* **Engine** (*string*) –

Provides the name of the database engine to be used for this DB cluster.

* **EngineVersion** (*string*) –

Indicates the database engine version.

* **LatestRestorableTime** (*datetime*) –

Specifies the latest time to which a database can be restored with point-in-time restore.

* **Port** (*integer*) –

Specifies the port that the database engine is listening on.

* **MasterUsername** (*string*) –

Contains the master username for the DB cluster.

* **DBClusterOptionGroupMemberships** (*list*) –

Provides the list of option group memberships for this DB cluster.

· (*dict*) –

Contains status information for a DB cluster option group.

· **DBClusterOptionGroupName** (*string*) –

Specifies the name of the DB cluster option group.

· **Status** (*string*) –

Specifies the status of the DB cluster option group.

- * **PreferredBackupWindow** (*string*) –

Specifies the daily time range during which automated backups are created if automated backups are enabled, as determined by the `BackupRetentionPeriod`.

- * **PreferredMaintenanceWindow** (*string*) –

Specifies the weekly time range during which system maintenance can occur, in Universal Coordinated Time (UTC).

- * **DBClusterMembers** (*list*) –

Provides the list of instances that make up the DB cluster.

- (*dict*) –

Contains information about an instance that is part of a DB cluster.

- **DBInstanceIdentifier** (*string*) –

Specifies the instance identifier for this member of the DB cluster.

- **IsClusterWriter** (*boolean*) –

Value that is `true` if the cluster member is the primary instance for the DB cluster and `false` otherwise.

- **DBClusterParameterGroupStatus** (*string*) –

Specifies the status of the DB cluster parameter group for this member of the DB cluster.

- * **VpcSecurityGroups** (*list*) –

Provides a list of VPC security groups that the DB cluster belongs to.

- (*dict*) –

This data type is used as a response element for queries on VPC security group membership.

- **VpcSecurityGroupId** (*string*) –

The name of the VPC security group.

- **Status** (*string*) –

The status of the VPC security group.

modify_db_cluster_parameter_group (***kwargs*)

Modifies the parameters of a DB cluster parameter group. To modify more than one parameter, submit a list of the following: `ParameterName`, `ParameterValue`, and `ApplyMethod`. A maximum of 20 parameters can be modified in a single request.

For more information on Amazon Aurora, see [Aurora on Amazon RDS](#) in the *Amazon RDS User Guide*.

Note: Changes to dynamic parameters are applied immediately. Changes to static parameters require a reboot without failover to the DB cluster associated with the parameter group before the change can take effect.

Warning: After you create a DB cluster parameter group, you should wait at least 5 minutes before creating your first DB cluster that uses that DB cluster parameter group as the default parameter group. This allows Amazon RDS to fully complete the create action before the parameter group is used as the default for a new DB cluster. This is especially important for parameters that are critical when creating the default database for a DB cluster, such as the character set for the default database defined by the `character_set_database` parameter. You can use the *Parameter Groups* option of the [Amazon RDS console](#) or the `DescribeDBClusterParameters` command to verify that your DB cluster parameter group has been created or modified.

Request Syntax

```
response = client.modify_db_cluster_parameter_group(
    DBClusterParameterGroupName='string',
    Parameters=[
        {
            'ParameterName': 'string',
            'ParameterValue': 'string',
            'Description': 'string',
            'Source': 'string',
            'ApplyType': 'string',
            'DataType': 'string',
            'AllowedValues': 'string',
            'IsModifiable': True|False,
            'MinimumEngineVersion': 'string',
            'ApplyMethod': 'immediate'|'pending-reboot'
        },
    ]
)
```

Parameters

- **DBClusterParameterGroupName** (*string*) – **[REQUIRED]**

The name of the DB cluster parameter group to modify.

- **Parameters** (*list*) – **[REQUIRED]**

A list of parameters in the DB cluster parameter group to modify.

– (*dict*) –

This data type is used as a request parameter in the `ModifyDBParameterGroup` and `ResetDBParameterGroup` actions.

This data type is used as a response element in the `DescribeEngineDefaultParameters` and `DescribeDBParameters` actions.

- * **ParameterName** (*string*) –

Specifies the name of the parameter.

- * **ParameterValue** (*string*) –

Specifies the value of the parameter.

- * **Description** (*string*) –

Provides a description of the parameter.

- * **Source** (*string*) –

Indicates the source of the parameter value.

- * **ApplyType** (*string*) –

Specifies the engine specific parameters type.

- * **DataType** (*string*) –
Specifies the valid data type for the parameter.
- * **AllowedValues** (*string*) –
Specifies the valid range of values for the parameter.
- * **IsModifiable** (*boolean*) –
Indicates whether (`true`) or not (`false`) the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.
- * **MinimumEngineVersion** (*string*) –
The earliest engine version to which the parameter can apply.
- * **ApplyMethod** (*string*) –
Indicates when to apply parameter updates.

Return type dict

Returns

Response Syntax

```
{
    'DBClusterParameterGroupName': 'string'
}
```

Response Structure

- (*dict*) –
 - **DBClusterParameterGroupName** (*string*) –

The name of the DB cluster parameter group.

Constraints:

- * Must be 1 to 255 alphanumeric characters
- * First character must be a letter
- * Cannot end with a hyphen or contain two consecutive hyphens

Note: This value is stored as a lowercase string.

modify_db_instance (***kwargs*)

Modify settings for a DB instance. You can change one or more database configuration parameters by specifying these parameters and the new values in the request.

Request Syntax

```
response = client.modify_db_instance(
    DBInstanceIdentifier='string',
    AllocatedStorage=123,
    DBInstanceClass='string',
    DBSecurityGroups=[
        'string',
    ],
    VpcSecurityGroupIds=[
        'string',
    ],
    ApplyImmediately=True|False,
    MasterUserPassword='string',
    DBParameterGroupName='string',
    BackupRetentionPeriod=123,
    PreferredBackupWindow='string',
```

```
PreferredMaintenanceWindow='string',
MultiAZ=True|False,
EngineVersion='string',
AllowMajorVersionUpgrade=True|False,
AutoMinorVersionUpgrade=True|False,
Iops=123,
OptionGroupName='string',
NewDBInstanceIdentifier='string',
StorageType='string',
TdeCredentialArn='string',
TdeCredentialPassword='string',
CACertificateIdentifier='string',
Domain='string',
CopyTagsToSnapshot=True|False
)
```

Parameters

- **DBInstanceIdentifier** (*string*) – [REQUIRED]

The DB instance identifier. This value is stored as a lowercase string.

Constraints:

- Must be the identifier for an existing DB instance
 - Must contain from 1 to 63 alphanumeric characters or hyphens
 - First character must be a letter
 - Cannot end with a hyphen or contain two consecutive hyphens
- **AllocatedStorage** (*integer*) – The new storage capacity of the RDS instance. Changing this setting does not result in an outage and the change is applied during the next maintenance window unless `ApplyImmediately` is set to `true` for this request.

MySQL

Default: Uses existing setting

Valid Values: 5-6144

Constraints: Value supplied must be at least 10% greater than the current value. Values that are not at least 10% greater than the existing value are rounded up so that they are 10% greater than the current value.

Type: Integer

PostgreSQL

Default: Uses existing setting

Valid Values: 5-6144

Constraints: Value supplied must be at least 10% greater than the current value. Values that are not at least 10% greater than the existing value are rounded up so that they are 10% greater than the current value.

Type: Integer

Oracle

Default: Uses existing setting

Valid Values: 10-6144

Constraints: Value supplied must be at least 10% greater than the current value. Values that are not at least 10% greater than the existing value are rounded up so that they are 10% greater than the current value.

SQL Server

Cannot be modified.

If you choose to migrate your DB instance from using standard storage to using Provisioned IOPS, or from using Provisioned IOPS to using standard storage, the process can take time. The duration of the migration depends on several factors such as database load, storage size, storage type (standard or Provisioned IOPS), amount of IOPS provisioned (if any), and the number of prior scale storage operations. Typical migration times are under 24 hours, but the process can take up to several days in some cases. During the migration, the DB instance will be available for use, but might experience performance degradation. While the migration takes place, nightly backups for the instance will be suspended. No other Amazon RDS operations can take place for the instance, including modifying the instance, rebooting the instance, deleting the instance, creating a Read Replica for the instance, and creating a DB snapshot of the instance.

- **DBInstanceClass** (*string*) – The new compute and memory capacity of the DB instance. To determine the instance classes that are available for a particular DB engine, use the `DescribeOrderableDBInstanceOptions` action.

Passing a value for this setting causes an outage during the change and is applied during the next maintenance window, unless `ApplyImmediately` is specified as `true` for this request.

Default: Uses existing setting

Valid Values: `db.t1.micro` | `db.m1.small` | `db.m1.medium` | `db.m1.large` | `db.m1.xlarge` | `db.m2.xlarge` | `db.m2.2xlarge` | `db.m2.4xlarge` | `db.m3.medium` | `db.m3.large` | `db.m3.xlarge` | `db.m3.2xlarge` | `db.r3.large` | `db.r3.xlarge` | `db.r3.2xlarge` | `db.r3.4xlarge` | `db.r3.8xlarge` | `db.t2.micro` | `db.t2.small` | `db.t2.medium`

- **DBSecurityGroups** (*list*) – A list of DB security groups to authorize on this DB instance. Changing this setting does not result in an outage and the change is asynchronously applied as soon as possible.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens
- (*string*) –

- **VpcSecurityGroupIds** (*list*) – A list of EC2 VPC security groups to authorize on this DB instance. This change is asynchronously applied as soon as possible.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens
- (*string*) –

- **ApplyImmediately** (*boolean*) – Specifies whether the modifications in this request and any pending modifications are asynchronously applied as soon as possible, regardless of the `PreferredMaintenanceWindow` setting for the DB instance.

If this parameter is set to `false`, changes to the DB instance are applied during the next maintenance window. Some parameter changes can cause an outage and will be applied on the next call to `RebootDBInstance`, or the next failure reboot.

Review the table of parameters in [Modifying a DB Instance and Using the Apply Immediately Parameter](#) to see the impact that setting `ApplyImmediately` to `true` or `false` has for each modified parameter and to determine when the changes will be applied.

Default: `false`

- **MasterUserPassword** (*string*) – The new password for the DB instance master user. Can be any printable ASCII character except `/`, `""`, or `@`.

Changing this parameter does not result in an outage and the change is asynchronously applied as soon as possible. Between the time of the request and the completion of the request, the `MasterUserPassword` element exists in the `PendingModifiedValues` element of the operation response.

Default: Uses existing setting

Constraints: Must be 8 to 41 alphanumeric characters (MySQL), 8 to 30 alphanumeric characters (Oracle), or 8 to 128 alphanumeric characters (SQL Server).

Note: Amazon RDS API actions never return the password, so this action provides a way to regain access to a primary instance user if the password is lost. This includes restoring privileges that might have been accidentally revoked.

- **DBParameterGroupName** (*string*) – The name of the DB parameter group to apply to the DB instance. Changing this setting does not result in an outage. The parameter group name itself is changed immediately, but the actual parameter changes are not applied until you reboot the instance without failover. The db instance will NOT be rebooted automatically and the parameter changes will NOT be applied during the next maintenance window.

Default: Uses existing setting

Constraints: The DB parameter group must be in the same DB parameter group family as this DB instance.

- **BackupRetentionPeriod** (*integer*) – The number of days to retain automated backups. Setting this parameter to a positive number enables backups. Setting this parameter to 0 disables automated backups.

Changing this parameter can result in an outage if you change from 0 to a non-zero value or from a non-zero value to 0. These changes are applied during the next maintenance window unless the `ApplyImmediately` parameter is set to `true` for this request. If you change the parameter from one non-zero value to another non-zero value, the change is asynchronously applied as soon as possible.

Default: Uses existing setting

Constraints:

- Must be a value from 0 to 35
 - Can be specified for a MySQL Read Replica only if the source is running MySQL 5.6
 - Can be specified for a PostgreSQL Read Replica only if the source is running PostgreSQL 9.3.5
 - Cannot be set to 0 if the DB instance is a source to Read Replicas
- **PreferredBackupWindow** (*string*) – The daily time range during which automated backups are created if automated backups are enabled, as determined by the `BackupRetentionPeriod` parameter. Changing this parameter does not result in an outage and the change is asynchronously applied as soon as possible.

Constraints:

- Must be in the format hh24:mi-hh24:mi
- Times should be in Universal Time Coordinated (UTC)
- Must not conflict with the preferred maintenance window
- Must be at least 30 minutes
- **PreferredMaintenanceWindow** (*string*) – The weekly time range (in UTC) during which system maintenance can occur, which might result in an outage. Changing this parameter does not result in an outage, except in the following situation, and the change is asynchronously applied as soon as possible. If there are pending actions that cause a reboot, and the maintenance window is changed to include the current time, then changing this parameter will cause a reboot of the DB instance. If moving this window to the current time, there must be at least 30 minutes between the current time and end of the window to ensure pending changes are applied.

Default: Uses existing setting

Format: ddd:hh24:mi-ddd:hh24:mi

Valid Days: Mon | Tue | Wed | Thu | Fri | Sat | Sun

Constraints: Must be at least 30 minutes

- **MultiAZ** (*boolean*) – Specifies if the DB instance is a Multi-AZ deployment. Changing this parameter does not result in an outage and the change is applied during the next maintenance window unless the `ApplyImmediately` parameter is set to `true` for this request.

Constraints: Cannot be specified if the DB instance is a Read Replica. This parameter cannot be used with SQL Server DB instances. Multi-AZ for SQL Server DB instances is set using the Mirroring option in an option group associated with the DB instance.

- **EngineVersion** (*string*) – The version number of the database engine to upgrade to. Changing this parameter results in an outage and the change is applied during the next maintenance window unless the `ApplyImmediately` parameter is set to `true` for this request.

For major version upgrades, if a non-default DB parameter group is currently in use, a new DB parameter group in the DB parameter group family for the new engine version must be specified. The new DB parameter group can be the default for that DB parameter group family.

For a list of valid engine versions, see `CreateDBInstance`.

- **AllowMajorVersionUpgrade** (*boolean*) – Indicates that major version upgrades are allowed. Changing this parameter does not result in an outage and the change is asynchronously applied as soon as possible.

Constraints: This parameter must be set to `true` when specifying a value for the `EngineVersion` parameter that is a different major version than the DB instance's current version.

- **AutoMinorVersionUpgrade** (*boolean*) – Indicates that minor version upgrades will be applied automatically to the DB instance during the maintenance window. Changing this parameter does not result in an outage except in the following case and the change is asynchronously applied as soon as possible. An outage will result if this parameter is set to `true` during the maintenance window, and a newer minor version is available, and RDS has enabled auto patching for that engine version.
- **Iops** (*integer*) – The new Provisioned IOPS (I/O operations per second) value for the RDS instance. Changing this setting does not result in an outage

and the change is applied during the next maintenance window unless the `ApplyImmediately` parameter is set to `true` for this request.

Default: Uses existing setting

Constraints: Value supplied must be at least 10% greater than the current value. Values that are not at least 10% greater than the existing value are rounded up so that they are 10% greater than the current value. If you are migrating from Provisioned IOPS to standard storage, set this value to 0. The DB instance will require a reboot for the change in storage type to take effect.

SQL Server

Setting the IOPS value for the SQL Server database engine is not supported.

Type: Integer

If you choose to migrate your DB instance from using standard storage to using Provisioned IOPS, or from using Provisioned IOPS to using standard storage, the process can take time. The duration of the migration depends on several factors such as database load, storage size, storage type (standard or Provisioned IOPS), amount of IOPS provisioned (if any), and the number of prior scale storage operations. Typical migration times are under 24 hours, but the process can take up to several days in some cases. During the migration, the DB instance will be available for use, but might experience performance degradation. While the migration takes place, nightly backups for the instance will be suspended. No other Amazon RDS operations can take place for the instance, including modifying the instance, rebooting the instance, deleting the instance, creating a Read Replica for the instance, and creating a DB snapshot of the instance.

- **OptionGroupName** (*string*) – Indicates that the DB instance should be associated with the specified option group. Changing this parameter does not result in an outage except in the following case and the change is applied during the next maintenance window unless the `ApplyImmediately` parameter is set to `true` for this request. If the parameter change results in an option group that enables OEM, this change can cause a brief (sub-second) period during which new connections are rejected but existing connections are not interrupted.

Permanent options, such as the TDE option for Oracle Advanced Security TDE, cannot be removed from an option group, and that option group cannot be removed from a DB instance once it is associated with a DB instance

- **NewDBInstanceIdentifier** (*string*) – The new DB instance identifier for the DB instance when renaming a DB instance. When you change the DB instance identifier, an instance reboot will occur immediately if you set `ApplyImmediately` to `true`, or will occur during the next maintenance window if `ApplyImmediately` to `false`. This value is stored as a lowercase string.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens
- **StorageType** (*string*) – Specifies the storage type to be associated with the DB instance.

Valid values: `standard` | `gp2` | `io1`

If you specify `io1`, you must also include a value for the `Iops` parameter.

Default: `io1` if the `Iops` parameter is specified; otherwise `standard`

- **TdeCredentialArn** (*string*) – The ARN from the Key Store with which to associate the instance for TDE encryption.

- **TdeCredentialPassword** (*string*) – The password for the given ARN from the Key Store in order to access the device.
- **CACertificateIdentifier** (*string*) – Indicates the certificate that needs to be associated with the instance.
- **Domain** (*string*) – Specify the Active Directory Domain to move the instance to.
The specified Active Directory Domain must be created prior to this operation. Currently only SQL Server instance can be created in a Domain
- **CopyTagsToSnapshot** (*boolean*) – This property is not currently implemented.

Return type dict

Returns

Response Syntax

```
{
  'DBInstance': {
    'DBInstanceIdentifier': 'string',
    'DBInstanceClass': 'string',
    'Engine': 'string',
    'DBInstanceStatus': 'string',
    'MasterUsername': 'string',
    'DBName': 'string',
    'Endpoint': {
      'Address': 'string',
      'Port': 123
    },
  },
  'AllocatedStorage': 123,
  'InstanceCreateTime': datetime(2015, 1, 1),
  'PreferredBackupWindow': 'string',
  'BackupRetentionPeriod': 123,
  'DBSecurityGroups': [
    {
      'DBSecurityGroupName': 'string',
      'Status': 'string'
    },
  ],
  'VpcSecurityGroups': [
    {
      'VpcSecurityGroupId': 'string',
      'Status': 'string'
    },
  ],
  'DBParameterGroups': [
    {
      'DBParameterGroupName': 'string',
      'ParameterApplyStatus': 'string'
    },
  ],
  'AvailabilityZone': 'string',
  'DBSubnetGroup': {
    'DBSubnetGroupName': 'string',
    'DBSubnetGroupDescription': 'string',
    'VpcId': 'string',
    'SubnetGroupStatus': 'string',
    'Subnets': [
      {
        'SubnetIdentifier': 'string',
        'SubnetAvailabilityZone': {
```

```
        'Name': 'string'
    },
    'SubnetStatus': 'string'
},
]
},
'PreferredMaintenanceWindow': 'string',
'PendingModifiedValues': {
    'DBInstanceClass': 'string',
    'AllocatedStorage': 123,
    'MasterUserPassword': 'string',
    'Port': 123,
    'BackupRetentionPeriod': 123,
    'MultiAZ': True|False,
    'EngineVersion': 'string',
    'Iops': 123,
    'DBInstanceIdentifier': 'string',
    'StorageType': 'string',
    'CACertificateIdentifier': 'string'
},
'LatestRestorableTime': datetime(2015, 1, 1),
'MultiAZ': True|False,
'EngineVersion': 'string',
'AutoMinorVersionUpgrade': True|False,
'ReadReplicaSourceDBInstanceIdentifier': 'string',
'ReadReplicaDBInstanceIdentifiers': [
    'string',
],
'LicenseModel': 'string',
'Iops': 123,
'OptionGroupMemberships': [
    {
        'OptionGroupName': 'string',
        'Status': 'string'
    },
],
'CharacterSetName': 'string',
'SecondaryAvailabilityZone': 'string',
'PubliclyAccessible': True|False,
'StatusInfos': [
    {
        'StatusType': 'string',
        'Normal': True|False,
        'Status': 'string',
        'Message': 'string'
    },
],
'StorageType': 'string',
'TdeCredentialArn': 'string',
'DbInstancePort': 123,
'DBClusterIdentifier': 'string',
'StorageEncrypted': True|False,
'KmsKeyId': 'string',
'DbiResourceId': 'string',
'CACertificateIdentifier': 'string',
'DomainMemberships': [
    {
        'Domain': 'string',
```

```

        'Status': 'string',
        'Connectivity': 'string'
    },
],
'CopyTagsToSnapshot': True|False
}

```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the following actions:

- CreateDBInstance
- DeleteDBInstance
- ModifyDBInstance

This data type is used as a response element in the DescribeDBInstances action.

- **DBInstance** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * CreateDBInstance
- * DeleteDBInstance
- * ModifyDBInstance

This data type is used as a response element in the DescribeDBInstances action.

- * **DBInstanceIdentifier** (*string*) –

Contains a user-supplied database identifier. This identifier is the unique key that identifies a DB instance.

- * **DBInstanceClass** (*string*) –

Contains the name of the compute and memory capacity class of the DB instance.

- * **Engine** (*string*) –

Provides the name of the database engine to be used for this DB instance.

- * **DBInstanceStatus** (*string*) –

Specifies the current state of this database.

- * **MasterUsername** (*string*) –

Contains the master username for the DB instance.

- * **DBName** (*string*) –

The meaning of this parameter differs according to the database engine you use. For example, this value returns either MySQL or PostgreSQL information when returning values from CreateDBInstanceReadReplica since Read Replicas are only supported for MySQL and PostgreSQL.

MySQL, SQL Server, PostgreSQL

Contains the name of the initial database of this instance that was provided at create time, if one was specified when the DB instance was created. This same name is returned for the life of the DB instance.

Type: String

Oracle

Contains the Oracle System ID (SID) of the created DB instance. Not shown when the returned parameters do not apply to an Oracle

DB instance.

* **Endpoint** (*dict*) –

Specifies the connection endpoint.

· **Address** (*string*) –

Specifies the DNS address of the DB instance.

· **Port** (*integer*) –

Specifies the port that the database engine is listening on.

* **AllocatedStorage** (*integer*) –

Specifies the allocated storage size specified in gigabytes.

* **InstanceCreateTime** (*datetime*) –

Provides the date and time the DB instance was created.

* **PreferredBackupWindow** (*string*) –

Specifies the daily time range during which automated backups are created if automated backups are enabled, as determined by the `BackupRetentionPeriod`.

* **BackupRetentionPeriod** (*integer*) –

Specifies the number of days for which automatic DB snapshots are retained.

* **DBSecurityGroups** (*list*) –

Provides List of DB security group elements containing only `DBSecurityGroup.Name` and `DBSecurityGroup.Status` subelements.

· (*dict*) –

This data type is used as a response element in the following actions:

- `ModifyDBInstance`
- `RebootDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceToPointInTime`
- **DBSecurityGroupName** (*string*) –

The name of the DB security group.

· **Status** (*string*) –

The status of the DB security group.

* **VpcSecurityGroups** (*list*) –

Provides List of VPC security group elements that the DB instance belongs to.

· (*dict*) –

This data type is used as a response element for queries on VPC security group membership.

· **VpcSecurityGroupId** (*string*) –

The name of the VPC security group.

· **Status** (*string*) –

The status of the VPC security group.

* **DBParameterGroups** (*list*) –

Provides the list of DB parameter groups applied to this DB instance.

- (*dict*) –

The status of the DB parameter group.

This data type is used as a response element in the following actions:

- CreateDBInstance
- CreateDBInstanceReadReplica
- DeleteDBInstance
- ModifyDBInstance
- RebootDBInstance
- RestoreDBInstanceFromDBSnapshot
- **DBParameterGroupName** (*string*) –

The name of the DP parameter group.

- **ParameterApplyStatus** (*string*) –

The status of parameter updates.

- * **AvailabilityZone** (*string*) –

Specifies the name of the Availability Zone the DB instance is located in.

- * **DBSubnetGroup** (*dict*) –

Specifies information on the subnet group associated with the DB instance, including the name, description, and subnets in the subnet group.

- **DBSubnetGroupName** (*string*) –

Specifies the name of the DB subnet group.

- **DBSubnetGroupDescription** (*string*) –

Provides the description of the DB subnet group.

- **VpcId** (*string*) –

Provides the VpcId of the DB subnet group.

- **SubnetGroupStatus** (*string*) –

Provides the status of the DB subnet group.

- **Subnets** (*list*) –

Contains a list of Subnet elements.

- (*dict*) –

This data type is used as a response element in the DescribeDBSubnetGroups action.

- **SubnetIdentifier** (*string*) –

Specifies the identifier of the subnet.

- **SubnetAvailabilityZone** (*dict*) –

Contains Availability Zone information.

This data type is used as an element in the following data type:

- OrderableDBInstanceOption
- **Name** (*string*) –

The name of the availability zone.

- **SubnetStatus** (*string*) –

Specifies the status of the subnet.

* **PreferredMaintenanceWindow** (*string*) –

Specifies the weekly time range during which system maintenance can occur, in Universal Coordinated Time (UTC).

* **PendingModifiedValues** (*dict*) –

Specifies that changes to the DB instance are pending. This element is only included when changes are pending. Specific changes are identified by subelements.

· **DBInstanceClass** (*string*) –

Contains the new `DBInstanceClass` for the DB instance that will be applied or is in progress.

· **AllocatedStorage** (*integer*) –

Contains the new `AllocatedStorage` size for the DB instance that will be applied or is in progress.

· **MasterUserPassword** (*string*) –

Contains the pending or in-progress change of the master credentials for the DB instance.

· **Port** (*integer*) –

Specifies the pending port for the DB instance.

· **BackupRetentionPeriod** (*integer*) –

Specifies the pending number of days for which automated backups are retained.

· **MultiAZ** (*boolean*) –

Indicates that the Single-AZ DB instance is to change to a Multi-AZ deployment.

· **EngineVersion** (*string*) –

Indicates the database engine version.

· **Iops** (*integer*) –

Specifies the new Provisioned IOPS value for the DB instance that will be applied or is being applied.

· **DBInstanceIdentifier** (*string*) –

Contains the new `DBInstanceIdentifier` for the DB instance that will be applied or is in progress.

· **StorageType** (*string*) –

Specifies the storage type to be associated with the DB instance.

· **CACertificateIdentifier** (*string*) –

Specifies the identifier of the CA certificate for the DB instance.

* **LatestRestorableTime** (*datetime*) –

Specifies the latest time to which a database can be restored with point-in-time restore.

* **MultiAZ** (*boolean*) –

Specifies if the DB instance is a Multi-AZ deployment.

* **EngineVersion** (*string*) –

Indicates the database engine version.

- * **AutoMinorVersionUpgrade** (*boolean*) –
Indicates that minor version patches are applied automatically.
- * **ReadReplicaSourceDBInstanceIdentifier** (*string*) –
Contains the identifier of the source DB instance if this DB instance is a Read Replica.
- * **ReadReplicaDBInstanceIdentifiers** (*list*) –
Contains one or more identifiers of the Read Replicas associated with this DB instance.
 - (*string*) –
- * **LicenseModel** (*string*) –
License model information for this DB instance.
- * **Iops** (*integer*) –
Specifies the Provisioned IOPS (I/O operations per second) value.
- * **OptionGroupMemberships** (*list*) –
Provides the list of option group memberships for this DB instance.
 - (*dict*) –
 - Provides information on the option groups the DB instance is a member of.
 - **OptionGroupName** (*string*) –
The name of the option group that the instance belongs to.
 - **Status** (*string*) –
The status of the DB instance's option group membership (e.g. in-sync, pending, pending-maintenance, applying).
- * **CharacterSetName** (*string*) –
If present, specifies the name of the character set that this instance is associated with.
- * **SecondaryAvailabilityZone** (*string*) –
If present, specifies the name of the secondary Availability Zone for a DB instance with multi-AZ support.
- * **PubliclyAccessible** (*boolean*) –
Specifies the accessibility options for the DB instance. A value of true specifies an Internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. A value of false specifies an internal instance with a DNS name that resolves to a private IP address.

Default: The default behavior varies depending on whether a VPC has been requested or not. The following list shows the default behavior in each case.
 - **Default VPC:** true
 - **VPC:** false
 If no DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be publicly accessible. If a specific DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be private.
- * **StatusInfos** (*list*) –

The status of a Read Replica. If the instance is not a Read Replica, this will be blank.

- *(dict)* –

Provides a list of status information for a DB instance.

- **StatusType** (*string*) –

This value is currently “read replication.”

- **Normal** (*boolean*) –

Boolean value that is true if the instance is operating normally, or false if the instance is in an error state.

- **Status** (*string*) –

Status of the DB instance. For a StatusType of read replica, the values can be replicating, error, stopped, or terminated.

- **Message** (*string*) –

Details of the error if there is an error for the instance. If the instance is not in an error state, this value is blank.

- * **StorageType** (*string*) –

Specifies the storage type associated with DB instance.

- * **TdeCredentialArn** (*string*) –

The ARN from the Key Store with which the instance is associated for TDE encryption.

- * **DbInstancePort** (*integer*) –

Specifies the port that the DB instance listens on. If the DB instance is part of a DB cluster, this can be a different port than the DB cluster port.

- * **DBClusterIdentifier** (*string*) –

If the DB instance is a member of a DB cluster, contains the name of the DB cluster that the DB instance is a member of.

- * **StorageEncrypted** (*boolean*) –

Specifies whether the DB instance is encrypted.

- * **KmsKeyId** (*string*) –

If StorageEncrypted is true, the KMS key identifier for the encrypted DB instance.

- * **DbiResourceId** (*string*) –

If StorageEncrypted is true, the region-unique, immutable identifier for the encrypted DB instance. This identifier is found in AWS CloudTrail log entries whenever the KMS key for the DB instance is accessed.

- * **CACertificateIdentifier** (*string*) –

The identifier of the CA certificate for this DB instance.

- * **DomainMemberships** (*list*) –

The Active Directory Domain membership records associated with the DB instance.

- *(dict)* –

An Active Directory Domain membership record associated with the DB instance.

- **Domain** (*string*) –
The identifier of the Active Directory Domain.
- **Status** (*string*) –
The status of the DB instance's Active Directory Domain membership (e.g. joined, pending-join, failed etc).
- **Connectivity** (*string*) –
The observed connectivity of the Active Directory Domain.

* **CopyTagsToSnapshot** (*boolean*) –

This property is not currently implemented.

modify_db_parameter_group (***kwargs*)

Modifies the parameters of a DB parameter group. To modify more than one parameter, submit a list of the following: `ParameterName`, `ParameterValue`, and `ApplyMethod`. A maximum of 20 parameters can be modified in a single request.

Note: Changes to dynamic parameters are applied immediately. Changes to static parameters require a reboot without failover to the DB instance associated with the parameter group before the change can take effect.

Warning: After you modify a DB parameter group, you should wait at least 5 minutes before creating your first DB instance that uses that DB parameter group as the default parameter group. This allows Amazon RDS to fully complete the modify action before the parameter group is used as the default for a new DB instance. This is especially important for parameters that are critical when creating the default database for a DB instance, such as the character set for the default database defined by the `character_set_database` parameter. You can use the *Parameter Groups* option of the [Amazon RDS console](#) or the `DescribeDBParameters` command to verify that your DB parameter group has been created or modified.

Request Syntax

```
response = client.modify_db_parameter_group(
    DBParameterGroupName='string',
    Parameters=[
        {
            'ParameterName': 'string',
            'ParameterValue': 'string',
            'Description': 'string',
            'Source': 'string',
            'ApplyType': 'string',
            'DataType': 'string',
            'AllowedValues': 'string',
            'IsModifiable': True|False,
            'MinimumEngineVersion': 'string',
            'ApplyMethod': 'immediate'|'pending-reboot'
        },
    ]
)
```

Parameters

- **DBParameterGroupName** (*string*) – [REQUIRED]

The name of the DB parameter group.

Constraints:

- Must be the name of an existing DB parameter group
- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **Parameters** (*list*) – [REQUIRED]

An array of parameter names, values, and the apply method for the parameter update. At least one parameter name, value, and apply method must be supplied; subsequent arguments are optional. A maximum of 20 parameters can be modified in a single request.

Valid Values (for the application method): `immediate` | `pending-reboot`

Note: You can use the `immediate` value with dynamic parameters only. You can use the `pending-reboot` value for both dynamic and static parameters, and changes are applied when you reboot the DB instance without failover.

- (*dict*) –

This data type is used as a request parameter in the `ModifyDBParameterGroup` and `ResetDBParameterGroup` actions.

This data type is used as a response element in the `DescribeEngineDefaultParameters` and `DescribeDBParameters` actions.

- * **ParameterName** (*string*) –

Specifies the name of the parameter.

- * **ParameterValue** (*string*) –

Specifies the value of the parameter.

- * **Description** (*string*) –

Provides a description of the parameter.

- * **Source** (*string*) –

Indicates the source of the parameter value.

- * **ApplyType** (*string*) –

Specifies the engine specific parameters type.

- * **DataType** (*string*) –

Specifies the valid data type for the parameter.

- * **AllowedValues** (*string*) –

Specifies the valid range of values for the parameter.

- * **IsModifiable** (*boolean*) –

Indicates whether (`true`) or not (`false`) the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.

- * **MinimumEngineVersion** (*string*) –

The earliest engine version to which the parameter can apply.

- * **ApplyMethod** (*string*) –

Indicates when to apply parameter updates.

Return type `dict`

Returns

Response Syntax

```
{
    'DBParameterGroupName': 'string'
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the ModifyDBParameterGroup or ResetDBParameterGroup action.

- **DBParameterGroupName** (*string*) –

Provides the name of the DB parameter group.

modify_db_subnet_group (***kwargs*)

Modifies an existing DB subnet group. DB subnet groups must contain at least one subnet in at least two AZs in the region.

Request Syntax

```
response = client.modify_db_subnet_group(
    DBSubnetGroupName='string',
    DBSubnetGroupDescription='string',
    SubnetIds=[
        'string',
    ]
)
```

Parameters

- **DBSubnetGroupName** (*string*) – **[REQUIRED]**

The name for the DB subnet group. This value is stored as a lowercase string.

Constraints: Must contain no more than 255 alphanumeric characters or hyphens. Must not be “Default”.

Example: mySubnetgroup

- **DBSubnetGroupDescription** (*string*) – The description for the DB subnet group.
- **SubnetIds** (*list*) – **[REQUIRED]**

The EC2 subnet IDs for the DB subnet group.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'DBSubnetGroup': {
        'DBSubnetGroupName': 'string',
        'DBSubnetGroupDescription': 'string',
        'VpcId': 'string',
        'SubnetGroupStatus': 'string',
        'Subnets': [
            {
                'SubnetIdentifier': 'string',
                'SubnetAvailabilityZone': {
                    'Name': 'string'
                }
            },
        ],
    },
}
```

```
        'SubnetStatus': 'string'
    },
]
}
}
```

Response Structure

- *(dict)* –

Contains the result of a successful invocation of the following actions:

- CreateDBSubnetGroup
- ModifyDBSubnetGroup
- DescribeDBSubnetGroups
- DeleteDBSubnetGroup

This data type is used as a response element in the DescribeDBSubnetGroups action.

- **DBSubnetGroup** *(dict)* –

Contains the result of a successful invocation of the following actions:

- * CreateDBSubnetGroup
- * ModifyDBSubnetGroup
- * DescribeDBSubnetGroups
- * DeleteDBSubnetGroup

This data type is used as a response element in the DescribeDBSubnetGroups action.

- * **DBSubnetGroupName** *(string)* –

Specifies the name of the DB subnet group.

- * **DBSubnetGroupDescription** *(string)* –

Provides the description of the DB subnet group.

- * **VpcId** *(string)* –

Provides the VpcId of the DB subnet group.

- * **SubnetGroupStatus** *(string)* –

Provides the status of the DB subnet group.

- * **Subnets** *(list)* –

Contains a list of Subnet elements.

- *(dict)* –

This data type is used as a response element in the DescribeDBSubnetGroups action.

- **SubnetIdentifier** *(string)* –

Specifies the identifier of the subnet.

- **SubnetAvailabilityZone** *(dict)* –

Contains Availability Zone information.

This data type is used as an element in the following data type:

- OrderableDBInstanceOption

- **Name** *(string)* –

The name of the availability zone.

- **SubnetStatus** *(string)* –

Specifies the status of the subnet.

modify_event_subscription (**kwargs)

Modifies an existing RDS event notification subscription. Note that you cannot modify the source identifiers using this call; to change source identifiers for a subscription, use the `AddSourceIdentifierToSubscription` and `RemoveSourceIdentifierFromSubscription` calls.

You can see a list of the event categories for a given `SourceType` in the [Events](#) topic in the Amazon RDS User Guide or by using the `DescribeEventCategories` action.

Request Syntax

```
response = client.modify_event_subscription(
    SubscriptionName='string',
    SnsTopicArn='string',
    SourceType='string',
    EventCategories=[
        'string',
    ],
    Enabled=True|False
)
```

Parameters

- **SubscriptionName** (*string*) – **[REQUIRED]**
The name of the RDS event notification subscription.
- **SnsTopicArn** (*string*) – The Amazon Resource Name (ARN) of the SNS topic created for event notification. The ARN is created by Amazon SNS when you create a topic and subscribe to it.
- **SourceType** (*string*) – The type of source that will be generating the events. For example, if you want to be notified of events generated by a DB instance, you would set this parameter to `db-instance`. If this value is not specified, all events are returned.

Valid values: `db-instance` | `db-parameter-group` | `db-security-group` | `db-snapshot`
- **EventCategories** (*list*) – A list of event categories for a `SourceType` that you want to subscribe to. You can see a list of the categories for a given `SourceType` in the [Events](#) topic in the Amazon RDS User Guide or by using the `DescribeEventCategories` action.
– (*string*) –
- **Enabled** (*boolean*) – A Boolean value; set to **true** to activate the subscription.

Return type dict

Returns**Response Syntax**

```
{
  'EventSubscription': {
    'CustomerAwsId': 'string',
    'CustSubscriptionId': 'string',
    'SnsTopicArn': 'string',
    'Status': 'string',
    'SubscriptionCreationTime': 'string',
    'SourceType': 'string',
    'SourceIdsList': [
        'string',
    ],
    'EventCategoriesList': [
        'string',
    ],
  },
}
```

```
        'Enabled': True|False
    }
}
```

Response Structure

- (*dict*) –

Contains the results of a successful invocation of the DescribeEventSubscriptions action.

- **EventSubscription** (*dict*) –

Contains the results of a successful invocation of the DescribeEventSubscriptions action.

- * **CustomerAwsId** (*string*) –

The AWS customer account associated with the RDS event notification subscription.

- * **CustSubscriptionId** (*string*) –

The RDS event notification subscription Id.

- * **SnsTopicArn** (*string*) –

The topic ARN of the RDS event notification subscription.

- * **Status** (*string*) –

The status of the RDS event notification subscription.

Constraints:

Can be one of the following: creating | modifying | deleting | active | no-permission | topic-not-exist

The status “no-permission” indicates that RDS no longer has permission to post to the SNS topic. The status “topic-not-exist” indicates that the topic was deleted after the subscription was created.

- * **SubscriptionCreationTime** (*string*) –

The time the RDS event notification subscription was created.

- * **SourceType** (*string*) –

The source type for the RDS event notification subscription.

- * **SourceIdsList** (*list*) –

A list of source IDs for the RDS event notification subscription.

· (*string*) –

- * **EventCategoriesList** (*list*) –

A list of event categories for the RDS event notification subscription.

· (*string*) –

- * **Enabled** (*boolean*) –

A Boolean value indicating if the subscription is enabled. True indicates the subscription is enabled.

modify_option_group (***kwargs*)

Modifies an existing option group.

Request Syntax

```

response = client.modify_option_group(
    OptionGroupName='string',
    OptionsToInclude=[
        {
            'OptionName': 'string',
            'Port': 123,
            'DBSecurityGroupMemberships': [
                'string',
            ],
            'VpcSecurityGroupMemberships': [
                'string',
            ],
            'OptionSettings': [
                {
                    'Name': 'string',
                    'Value': 'string',
                    'DefaultValue': 'string',
                    'Description': 'string',
                    'ApplyType': 'string',
                    'DataType': 'string',
                    'AllowedValues': 'string',
                    'IsModifiable': True|False,
                    'IsCollection': True|False
                },
            ]
        },
    ],
    OptionsToRemove=[
        'string',
    ],
    ApplyImmediately=True|False
)

```

Parameters

- **OptionGroupName** (*string*) – [REQUIRED]

The name of the option group to be modified.

Permanent options, such as the TDE option for Oracle Advanced Security TDE, cannot be removed from an option group, and that option group cannot be removed from a DB instance once it is associated with a DB instance

- **OptionsToInclude** (*list*) – Options in this list are added to the option group or, if already present, the specified configuration is used to update the existing configuration.
 - (*dict*) –

A list of all available options

- * **OptionName** (*string*) – [REQUIRED]

The configuration of options to include in a group.

- * **Port** (*integer*) –

The optional port for the option.

- * **DBSecurityGroupMemberships** (*list*) –

A list of DBSecurityGroupMemembrship name strings used for this option.

· (*string*) –

- * **VpcSecurityGroupMemberships** (*list*) –

A list of VpcSecurityGroupMembership name strings used for this option.

- *(string)* –

* **OptionSettings** (*list*) –

The option settings to include in an option group.

- *(dict)* –

Option settings are the actual settings being applied or configured for that option. It is used when you modify an option group or describe option groups. For example, the NATIVE_NETWORK_ENCRYPTION option has a setting called SQLNET.ENCRYPTION_SERVER that can have several different values.

- **Name** (*string*) –

The name of the option that has settings that you can set.

- **Value** (*string*) –

The current value of the option setting.

- **DefaultValue** (*string*) –

The default value of the option setting.

- **Description** (*string*) –

The description of the option setting.

- **ApplyType** (*string*) –

The DB engine specific parameter type.

- **DataType** (*string*) –

The data type of the option setting.

- **AllowedValues** (*string*) –

The allowed values of the option setting.

- **IsModifiable** (*boolean*) –

A Boolean value that, when true, indicates the option setting can be modified from the default.

- **IsCollection** (*boolean*) –

Indicates if the option setting is part of a collection.

- **OptionsToRemove** (*list*) – Options in this list are removed from the option group.

- (*string*) –

- **ApplyImmediately** (*boolean*) – Indicates whether the changes should be applied immediately, or during the next maintenance window for each instance associated with the option group.

Return type dict

Returns

Response Syntax

```
{
  'OptionGroup': {
    'OptionGroupName': 'string',
    'OptionGroupDescription': 'string',
    'EngineName': 'string',
    'MajorEngineVersion': 'string',
    'Options': [
```



```

{
    'OptionName': 'string',
    'OptionDescription': 'string',
    'Persistent': True|False,
    'Permanent': True|False,
    'Port': 123,
    'OptionSettings': [
        {
            'Name': 'string',
            'Value': 'string',
            'DefaultValue': 'string',
            'Description': 'string',
            'ApplyType': 'string',
            'DataType': 'string',
            'AllowedValues': 'string',
            'IsModifiable': True|False,
            'IsCollection': True|False
        },
    ],
    'DBSecurityGroupMemberships': [
        {
            'DBSecurityGroupName': 'string',
            'Status': 'string'
        },
    ],
    'VpcSecurityGroupMemberships': [
        {
            'VpcSecurityGroupId': 'string',
            'Status': 'string'
        },
    ],
    ],
    'AllowsVpcAndNonVpcInstanceMemberships': True|False,
    'VpcId': 'string'
}

```

Response Structure

- (*dict*) –
 - **OptionGroup** (*dict*) –
 - * **OptionGroupName** (*string*) –

Specifies the name of the option group.
 - * **OptionGroupDescription** (*string*) –

Provides a description of the option group.
 - * **EngineName** (*string*) –

Indicates the name of the engine that this option group can be applied to.
 - * **MajorEngineVersion** (*string*) –

Indicates the major engine version associated with this option group.
 - * **Options** (*list*) –

Indicates what options are available in the option group.

 - (*dict*) –

Option details.

- **OptionName** (*string*) –
The name of the option.
- **OptionDescription** (*string*) –
The description of the option.
- **Persistent** (*boolean*) –
Indicate if this option is persistent.
- **Permanent** (*boolean*) –
Indicate if this option is permanent.
- **Port** (*integer*) –
If required, the port configured for this option to use.
- **OptionSettings** (*list*) –
The option settings for this option.
- (*dict*) –
Option settings are the actual settings being applied or configured for that option. It is used when you modify an option group or describe option groups. For example, the `NATIVE_NETWORK_ENCRYPTION` option has a setting called `SQLNET.ENCRYPTION_SERVER` that can have several different values.
- **Name** (*string*) –
The name of the option that has settings that you can set.
- **Value** (*string*) –
The current value of the option setting.
- **DefaultValue** (*string*) –
The default value of the option setting.
- **Description** (*string*) –
The description of the option setting.
- **ApplyType** (*string*) –
The DB engine specific parameter type.
- **DataType** (*string*) –
The data type of the option setting.
- **AllowedValues** (*string*) –
The allowed values of the option setting.
- **IsModifiable** (*boolean*) –
A Boolean value that, when true, indicates the option setting can be modified from the default.
- **IsCollection** (*boolean*) –
Indicates if the option setting is part of a collection.
- **DBSecurityGroupMemberships** (*list*) –
If the option requires access to a port, then this DB security group allows access to the port.
- (*dict*) –
This data type is used as a response element in the following actions:
 - `ModifyDBInstance`

- `RebootDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceToPointInTime`
- **`DBSecurityGroupName`** (*string*) –
The name of the DB security group.
- **`Status`** (*string*) –
The status of the DB security group.
- **`VpcSecurityGroupMemberships`** (*list*) –
If the option requires access to a port, then this VPC security group allows access to the port.
- (*dict*) –
This data type is used as a response element for queries on VPC security group membership.
- **`VpcSecurityGroupId`** (*string*) –
The name of the VPC security group.
- **`Status`** (*string*) –
The status of the VPC security group.
- * **`AllowsVpcAndNonVpcInstanceMemberships`** (*boolean*) –
Indicates whether this option group can be applied to both VPC and non-VPC instances. The value `true` indicates the option group can be applied to both VPC and non-VPC instances.
- * **`VpcId`** (*string*) –
If **`AllowsVpcAndNonVpcInstanceMemberships`** is `false`, this field is blank. If **`AllowsVpcAndNonVpcInstanceMemberships`** is `true` and this field is blank, then this option group can be applied to both VPC and non-VPC instances. If this field contains a value, then this option group can only be applied to instances that are in the VPC indicated by this field.

`promote_read_replica` (***kwargs*)

Promotes a Read Replica DB instance to a standalone DB instance.

Note: We recommend that you enable automated backups on your Read Replica before promoting the Read Replica. This ensures that no backup is taken during the promotion process. Once the instance is promoted to a primary instance, backups are taken based on your backup settings.

Request Syntax

```
response = client.promote_read_replica(
    DBInstanceIdentifier='string',
    BackupRetentionPeriod=123,
    PreferredBackupWindow='string'
)
```

Parameters

- **`DBInstanceIdentifier`** (*string*) – [REQUIRED]

The DB instance identifier. This value is stored as a lowercase string.

Constraints:

- Must be the identifier for an existing Read Replica DB instance
- Must contain from 1 to 63 alphanumeric characters or hyphens

- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Example: mydbinstance

- **BackupRetentionPeriod** (*integer*) – The number of days to retain automated backups. Setting this parameter to a positive number enables backups. Setting this parameter to 0 disables automated backups.

Default: 1

Constraints:

- Must be a value from 0 to 8

- **PreferredBackupWindow** (*string*) – The daily time range during which automated backups are created if automated backups are enabled, using the BackupRetentionPeriod parameter.

Default: A 30-minute window selected at random from an 8-hour block of time per region. To see the time blocks available, see [Adjusting the Preferred Maintenance Window](#) in the *Amazon RDS User Guide*.

Constraints:

- Must be in the format hh24:mi-hh24:mi .
- Times should be in Universal Coordinated Time (UTC).
- Must not conflict with the preferred maintenance window.
- Must be at least 30 minutes.

Return type dict

Returns

Response Syntax

```
{
  'DBInstance': {
    'DBInstanceIdentifier': 'string',
    'DBInstanceClass': 'string',
    'Engine': 'string',
    'DBInstanceStatus': 'string',
    'MasterUsername': 'string',
    'DBName': 'string',
    'Endpoint': {
      'Address': 'string',
      'Port': 123
    },
    'AllocatedStorage': 123,
    'InstanceCreateTime': datetime(2015, 1, 1),
    'PreferredBackupWindow': 'string',
    'BackupRetentionPeriod': 123,
    'DBSecurityGroups': [
      {
        'DBSecurityGroupName': 'string',
        'Status': 'string'
      },
    ],
    'VpcSecurityGroups': [
      {
        'VpcSecurityGroupId': 'string',
        'Status': 'string'
      },
    ],
    'DBParameterGroups': [
      {
```

```

        'DBParameterGroupName': 'string',
        'ParameterApplyStatus': 'string'
    },
],
'AvailabilityZone': 'string',
'DBSubnetGroup': {
    'DBSubnetGroupName': 'string',
    'DBSubnetGroupDescription': 'string',
    'VpcId': 'string',
    'SubnetGroupStatus': 'string',
    'Subnets': [
        {
            'SubnetIdentifier': 'string',
            'SubnetAvailabilityZone': {
                'Name': 'string'
            },
            'SubnetStatus': 'string'
        },
    ],
}
],
'PreferredMaintenanceWindow': 'string',
'PendingModifiedValues': {
    'DBInstanceClass': 'string',
    'AllocatedStorage': 123,
    'MasterUserPassword': 'string',
    'Port': 123,
    'BackupRetentionPeriod': 123,
    'MultiAZ': True|False,
    'EngineVersion': 'string',
    'Iops': 123,
    'DBInstanceIdentifier': 'string',
    'StorageType': 'string',
    'CACertificateIdentifier': 'string'
},
'LatestRestorableTime': datetime(2015, 1, 1),
'MultiAZ': True|False,
'EngineVersion': 'string',
'AutoMinorVersionUpgrade': True|False,
'ReadReplicaSourceDBInstanceIdentifier': 'string',
'ReadReplicaDBInstanceIdentifiers': [
    'string',
],
'LicenseModel': 'string',
'Iops': 123,
'OptionGroupMemberships': [
    {
        'OptionGroupName': 'string',
        'Status': 'string'
    },
],
'CharacterSetName': 'string',
'SecondaryAvailabilityZone': 'string',
'PubliclyAccessible': True|False,
'StatusInfos': [
    {
        'StatusType': 'string',
        'Normal': True|False,
        'Status': 'string',
    },
],

```

```
        'Message': 'string'
    },
],
'StorageType': 'string',
'TdeCredentialArn': 'string',
'DbInstancePort': 123,
'DBClusterIdentifier': 'string',
'StorageEncrypted': True|False,
'KmsKeyId': 'string',
'DbiResourceId': 'string',
'CACertificateIdentifier': 'string',
'DomainMemberships': [
    {
        'Domain': 'string',
        'Status': 'string',
        'Connectivity': 'string'
    },
],
'CopyTagsToSnapshot': True|False
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the following actions:

- CreateDBInstance
- DeleteDBInstance
- ModifyDBInstance

This data type is used as a response element in the DescribeDBInstances action.

- **DBInstance** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * CreateDBInstance
- * DeleteDBInstance
- * ModifyDBInstance

This data type is used as a response element in the DescribeDBInstances action.

- * **DBInstanceIdentifier** (*string*) –

Contains a user-supplied database identifier. This identifier is the unique key that identifies a DB instance.

- * **DBInstanceClass** (*string*) –

Contains the name of the compute and memory capacity class of the DB instance.

- * **Engine** (*string*) –

Provides the name of the database engine to be used for this DB instance.

- * **DBInstanceStatus** (*string*) –

Specifies the current state of this database.

- * **MasterUsername** (*string*) –

Contains the master username for the DB instance.

- * **DBName** (*string*) –

The meaning of this parameter differs according to the database

engine you use. For example, this value returns either MySQL or PostgreSQL information when returning values from `CreateDBInstanceReadReplica` since Read Replicas are only supported for MySQL and PostgreSQL.

MySQL, SQL Server, PostgreSQL

Contains the name of the initial database of this instance that was provided at create time, if one was specified when the DB instance was created. This same name is returned for the life of the DB instance.

Type: String

Oracle

Contains the Oracle System ID (SID) of the created DB instance. Not shown when the returned parameters do not apply to an Oracle DB instance.

* **Endpoint** (*dict*) –

Specifies the connection endpoint.

· **Address** (*string*) –

Specifies the DNS address of the DB instance.

· **Port** (*integer*) –

Specifies the port that the database engine is listening on.

* **AllocatedStorage** (*integer*) –

Specifies the allocated storage size specified in gigabytes.

* **InstanceCreateTime** (*datetime*) –

Provides the date and time the DB instance was created.

* **PreferredBackupWindow** (*string*) –

Specifies the daily time range during which automated backups are created if automated backups are enabled, as determined by the `BackupRetentionPeriod`.

* **BackupRetentionPeriod** (*integer*) –

Specifies the number of days for which automatic DB snapshots are retained.

* **DBSecurityGroups** (*list*) –

Provides List of DB security group elements containing only `DBSecurityGroup.Name` and `DBSecurityGroup.Status` subelements.

· (*dict*) –

This data type is used as a response element in the following actions:

- `ModifyDBInstance`
- `RebootDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceToPointInTime`
- **DBSecurityGroupName** (*string*) –

The name of the DB security group.

· **Status** (*string*) –

The status of the DB security group.

* **VpcSecurityGroups** (*list*) –

Provides List of VPC security group elements that the DB instance belongs to.

- *(dict)* –

This data type is used as a response element for queries on VPC security group membership.

- **VpcSecurityGroupId** (*string*) –

The name of the VPC security group.

- **Status** (*string*) –

The status of the VPC security group.

- * **DBParameterGroups** (*list*) –

Provides the list of DB parameter groups applied to this DB instance.

- *(dict)* –

The status of the DB parameter group.

This data type is used as a response element in the following actions:

- CreateDBInstance
- CreateDBInstanceReadReplica
- DeleteDBInstance
- ModifyDBInstance
- RebootDBInstance
- RestoreDBInstanceFromDBSnapshot
- **DBParameterGroupName** (*string*) –

The name of the DP parameter group.

- **ParameterApplyStatus** (*string*) –

The status of parameter updates.

- * **AvailabilityZone** (*string*) –

Specifies the name of the Availability Zone the DB instance is located in.

- * **DBSubnetGroup** (*dict*) –

Specifies information on the subnet group associated with the DB instance, including the name, description, and subnets in the subnet group.

- **DBSubnetGroupName** (*string*) –

Specifies the name of the DB subnet group.

- **DBSubnetGroupDescription** (*string*) –

Provides the description of the DB subnet group.

- **VpcId** (*string*) –

Provides the VpcId of the DB subnet group.

- **SubnetGroupStatus** (*string*) –

Provides the status of the DB subnet group.

- **Subnets** (*list*) –

Contains a list of Subnet elements.

- *(dict)* –

This data type is used as a response element in the DescribeDBSubnetGroups action.

- **SubnetIdentifier** (*string*) –
Specifies the identifier of the subnet.
- **SubnetAvailabilityZone** (*dict*) –
Contains Availability Zone information.

This data type is used as an element in the following data type:
- **OrderableDBInstanceOption**
- **Name** (*string*) –
The name of the availability zone.
- **SubnetStatus** (*string*) –
Specifies the status of the subnet.
- * **PreferredMaintenanceWindow** (*string*) –
Specifies the weekly time range during which system maintenance can occur, in Universal Coordinated Time (UTC).
- * **PendingModifiedValues** (*dict*) –
Specifies that changes to the DB instance are pending. This element is only included when changes are pending. Specific changes are identified by subelements.
 - **DBInstanceClass** (*string*) –
Contains the new DBInstanceClass for the DB instance that will be applied or is in progress.
 - **AllocatedStorage** (*integer*) –
Contains the new AllocatedStorage size for the DB instance that will be applied or is in progress.
 - **MasterUserPassword** (*string*) –
Contains the pending or in-progress change of the master credentials for the DB instance.
 - **Port** (*integer*) –
Specifies the pending port for the DB instance.
 - **BackupRetentionPeriod** (*integer*) –
Specifies the pending number of days for which automated backups are retained.
 - **MultiAZ** (*boolean*) –
Indicates that the Single-AZ DB instance is to change to a Multi-AZ deployment.
 - **EngineVersion** (*string*) –
Indicates the database engine version.
 - **Iops** (*integer*) –
Specifies the new Provisioned IOPS value for the DB instance that will be applied or is being applied.
 - **DBInstanceIdentifier** (*string*) –
Contains the new DBInstanceIdentifier for the DB instance that will be applied or is in progress.
 - **StorageType** (*string*) –
Specifies the storage type to be associated with the DB instance.

- **CACertificateIdentifier** (*string*) –
Specifies the identifier of the CA certificate for the DB instance.
- * **LatestRestorableTime** (*datetime*) –
Specifies the latest time to which a database can be restored with point-in-time restore.
- * **MultiAZ** (*boolean*) –
Specifies if the DB instance is a Multi-AZ deployment.
- * **EngineVersion** (*string*) –
Indicates the database engine version.
- * **AutoMinorVersionUpgrade** (*boolean*) –
Indicates that minor version patches are applied automatically.
- * **ReadReplicaSourceDBInstanceIdentifier** (*string*) –
Contains the identifier of the source DB instance if this DB instance is a Read Replica.
- * **ReadReplicaDBInstanceIdentifiers** (*list*) –
Contains one or more identifiers of the Read Replicas associated with this DB instance.
 - (*string*) –
- * **LicenseModel** (*string*) –
License model information for this DB instance.
- * **Iops** (*integer*) –
Specifies the Provisioned IOPS (I/O operations per second) value.
- * **OptionGroupMemberships** (*list*) –
Provides the list of option group memberships for this DB instance.
 - (*dict*) –
Provides information on the option groups the DB instance is a member of.
 - **OptionGroupName** (*string*) –
The name of the option group that the instance belongs to.
 - **Status** (*string*) –
The status of the DB instance's option group membership (e.g. in-sync, pending, pending-maintenance, applying).
- * **CharacterSetName** (*string*) –
If present, specifies the name of the character set that this instance is associated with.
- * **SecondaryAvailabilityZone** (*string*) –
If present, specifies the name of the secondary Availability Zone for a DB instance with multi-AZ support.
- * **PubliclyAccessible** (*boolean*) –
Specifies the accessibility options for the DB instance. A value of true specifies an Internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. A value of false specifies an internal instance with a DNS name that resolves to a private IP address.

Default: The default behavior varies depending on whether a VPC has been requested or not. The following list shows the default behavior in each case.

- **Default VPC:** true
- **VPC:** false

If no DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be publicly accessible. If a specific DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be private.

* **StatusInfos** (*list*) –

The status of a Read Replica. If the instance is not a Read Replica, this will be blank.

- (*dict*) –

Provides a list of status information for a DB instance.

- **StatusType** (*string*) –

This value is currently “read replication.”

- **Normal** (*boolean*) –

Boolean value that is true if the instance is operating normally, or false if the instance is in an error state.

- **Status** (*string*) –

Status of the DB instance. For a StatusType of read replica, the values can be replicating, error, stopped, or terminated.

- **Message** (*string*) –

Details of the error if there is an error for the instance. If the instance is not in an error state, this value is blank.

* **StorageType** (*string*) –

Specifies the storage type associated with DB instance.

* **TdeCredentialArn** (*string*) –

The ARN from the Key Store with which the instance is associated for TDE encryption.

* **DbInstancePort** (*integer*) –

Specifies the port that the DB instance listens on. If the DB instance is part of a DB cluster, this can be a different port than the DB cluster port.

* **DBClusterIdentifier** (*string*) –

If the DB instance is a member of a DB cluster, contains the name of the DB cluster that the DB instance is a member of.

* **StorageEncrypted** (*boolean*) –

Specifies whether the DB instance is encrypted.

* **KmsKeyId** (*string*) –

If StorageEncrypted is true, the KMS key identifier for the encrypted DB instance.

* **DbiResourceId** (*string*) –

If StorageEncrypted is true, the region-unique, immutable identifier for the encrypted DB instance. This identifier is found

in AWS CloudTrail log entries whenever the KMS key for the DB instance is accessed.

* **CACertificateIdentifier** (*string*) –

The identifier of the CA certificate for this DB instance.

* **DomainMemberships** (*list*) –

The Active Directory Domain membership records associated with the DB instance.

· (*dict*) –

An Active Directory Domain membership record associated with the DB instance.

· **Domain** (*string*) –

The identifier of the Active Directory Domain.

· **Status** (*string*) –

The status of the DB instance's Active Directory Domain membership (e.g. joined, pending-join, failed etc).

· **Connectivity** (*string*) –

The observed connectivity of the Active Directory Domain.

* **CopyTagsToSnapshot** (*boolean*) –

This property is not currently implemented.

purchase_reserved_db_instances_offering (**kwargs)

Purchases a reserved DB instance offering.

Request Syntax

```
response = client.purchase_reserved_db_instances_offering(
    ReservedDBInstancesOfferingId='string',
    ReservedDBInstanceId='string',
    DBInstanceCount=123,
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **ReservedDBInstancesOfferingId** (*string*) – [REQUIRED]

The ID of the Reserved DB instance offering to purchase.

Example: 438012d3-4052-4cc7-b2e3-8d3372e0e706

- **ReservedDBInstanceId** (*string*) – Customer-specified identifier to track this reservation.

Example: myreservationID

- **DBInstanceCount** (*integer*) – The number of instances to reserve.

Default: 1

- **Tags** (*list*) – A list of tags.
 - (*dict*) –

Metadata assigned to an Amazon RDS resource consisting of a key-value pair.

* **Key** (*string*) –

A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

* **Value** (*string*) –

A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

Return type dict

Returns

Response Syntax

```
{
  'ReservedDBInstance': {
    'ReservedDBInstanceId': 'string',
    'ReservedDBInstancesOfferingId': 'string',
    'DBInstanceClass': 'string',
    'StartTime': datetime(2015, 1, 1),
    'Duration': 123,
    'FixedPrice': 123.0,
    'UsagePrice': 123.0,
    'CurrencyCode': 'string',
    'DBInstanceCount': 123,
    'ProductDescription': 'string',
    'OfferingType': 'string',
    'MultiAZ': True|False,
    'State': 'string',
    'RecurringCharges': [
      {
        'RecurringChargeAmount': 123.0,
        'RecurringChargeFrequency': 'string'
      },
    ]
  }
}
```

Response Structure

• (*dict*) –

This data type is used as a response element in the DescribeReservedDBInstances and PurchaseReservedDBInstancesOffering actions.

– **ReservedDBInstance** (*dict*) –

This data type is used as a response element in the DescribeReservedDBInstances and PurchaseReservedDBInstancesOffering actions.

* **ReservedDBInstanceId** (*string*) –

The unique identifier for the reservation.

* **ReservedDBInstancesOfferingId** (*string*) –

The offering identifier.

* **DBInstanceClass** (*string*) –

The DB instance class for the reserved DB instance.

* **StartTime** (*datetime*) –

The time the reservation started.

* **Duration** (*integer*) –

The duration of the reservation in seconds.

* **FixedPrice** (*float*) –

The fixed price charged for this reserved DB instance.

* **UsagePrice** (*float*) –

The hourly price charged for this reserved DB instance.

* **CurrencyCode** (*string*) –

The currency code for the reserved DB instance.

* **DBInstanceCount** (*integer*) –

The number of reserved DB instances.

* **ProductDescription** (*string*) –

The description of the reserved DB instance.

* **OfferingType** (*string*) –

The offering type of this reserved DB instance.

* **MultiAZ** (*boolean*) –

Indicates if the reservation applies to Multi-AZ deployments.

* **State** (*string*) –

The state of the reserved DB instance.

* **RecurringCharges** (*list*) –

The recurring price charged to run this reserved DB instance.

· (*dict*) –

This data type is used as a response element in the DescribeReservedDBInstances and DescribeReservedDBInstancesOfferings actions.

· **RecurringChargeAmount** (*float*) –

The amount of the recurring charge.

· **RecurringChargeFrequency** (*string*) –

The frequency of the recurring charge.

reboot_db_instance (***kwargs*)

Rebooting a DB instance restarts the database engine service. A reboot also applies to the DB instance any modifications to the associated DB parameter group that were pending. Rebooting a DB instance results in a momentary outage of the instance, during which the DB instance status is set to rebooting. If the RDS instance is configured for MultiAZ, it is possible that the reboot will be conducted through a failover. An Amazon RDS event is created when the reboot is completed.

If your DB instance is deployed in multiple Availability Zones, you can force a failover from one AZ to the other during the reboot. You might force a failover to test the availability of your DB instance deployment or to restore operations to the original AZ after a failover occurs.

The time required to reboot is a function of the specific database engine's crash recovery process. To improve the reboot time, we recommend that you reduce database activities as much as possible during the reboot process to reduce rollback activity for in-transit transactions.

Request Syntax

```
response = client.reboot_db_instance(
    DBInstanceIdentifier='string',
    ForceFailover=True|False
)
```

Parameters

- **DBInstanceIdentifier** (*string*) – [REQUIRED]

The DB instance identifier. This parameter is stored as a lowercase string.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **ForceFailover** (*boolean*) – When `true`, the reboot will be conducted through a MultiAZ failover.

Constraint: You cannot specify `true` if the instance is not configured for MultiAZ.

Return type dict

Returns

Response Syntax

```
{
  'DBInstance': {
    'DBInstanceIdentifier': 'string',
    'DBInstanceClass': 'string',
    'Engine': 'string',
    'DBInstanceStatus': 'string',
    'MasterUsername': 'string',
    'DBName': 'string',
    'Endpoint': {
      'Address': 'string',
      'Port': 123
    },
    'AllocatedStorage': 123,
    'InstanceCreateTime': datetime(2015, 1, 1),
    'PreferredBackupWindow': 'string',
    'BackupRetentionPeriod': 123,
    'DBSecurityGroups': [
      {
        'DBSecurityGroupName': 'string',
        'Status': 'string'
      },
    ],
    'VpcSecurityGroups': [
      {
        'VpcSecurityGroupId': 'string',
        'Status': 'string'
      },
    ],
    'DBParameterGroups': [
      {
        'DBParameterGroupName': 'string',
        'ParameterApplyStatus': 'string'
      },
    ],
  },
}
```

```
'AvailabilityZone': 'string',
'DBSubnetGroup': {
    'DBSubnetGroupName': 'string',
    'DBSubnetGroupDescription': 'string',
    'VpcId': 'string',
    'SubnetGroupStatus': 'string',
    'Subnets': [
        {
            'SubnetIdentifier': 'string',
            'SubnetAvailabilityZone': {
                'Name': 'string'
            },
            'SubnetStatus': 'string'
        },
    ]
},
'PreferredMaintenanceWindow': 'string',
'PendingModifiedValues': {
    'DBInstanceClass': 'string',
    'AllocatedStorage': 123,
    'MasterUserPassword': 'string',
    'Port': 123,
    'BackupRetentionPeriod': 123,
    'MultiAZ': True|False,
    'EngineVersion': 'string',
    'Iops': 123,
    'DBInstanceIdentifier': 'string',
    'StorageType': 'string',
    'CACertificateIdentifier': 'string'
},
'LatestRestorableTime': datetime(2015, 1, 1),
'MultiAZ': True|False,
'EngineVersion': 'string',
'AutoMinorVersionUpgrade': True|False,
'ReadReplicaSourceDBInstanceIdentifier': 'string',
'ReadReplicaDBInstanceIdentifiers': [
    'string',
],
'LicenseModel': 'string',
'Iops': 123,
'OptionGroupMemberships': [
    {
        'OptionGroupName': 'string',
        'Status': 'string'
    },
],
'CharacterSetName': 'string',
'SecondaryAvailabilityZone': 'string',
'PubliclyAccessible': True|False,
'StatusInfos': [
    {
        'StatusType': 'string',
        'Normal': True|False,
        'Status': 'string',
        'Message': 'string'
    },
],
'StorageType': 'string',
```



```

'TdeCredentialArn': 'string',
'DbInstancePort': 123,
'DBClusterIdentifier': 'string',
'StorageEncrypted': True|False,
'KmsKeyId': 'string',
'DbiResourceId': 'string',
'CACertificateIdentifier': 'string',
'DomainMemberships': [
    {
        'Domain': 'string',
        'Status': 'string',
        'Connectivity': 'string'
    },
],
'CopyTagsToSnapshot': True|False
}

```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the following actions:

- CreateDBInstance
- DeleteDBInstance
- ModifyDBInstance

This data type is used as a response element in the DescribeDBInstances action.

- **DBInstance** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * CreateDBInstance
- * DeleteDBInstance
- * ModifyDBInstance

This data type is used as a response element in the DescribeDBInstances action.

- * **DBInstanceIdentifier** (*string*) –

Contains a user-supplied database identifier. This identifier is the unique key that identifies a DB instance.

- * **DBInstanceClass** (*string*) –

Contains the name of the compute and memory capacity class of the DB instance.

- * **Engine** (*string*) –

Provides the name of the database engine to be used for this DB instance.

- * **DBInstanceStatus** (*string*) –

Specifies the current state of this database.

- * **MasterUsername** (*string*) –

Contains the master username for the DB instance.

- * **DBName** (*string*) –

The meaning of this parameter differs according to the database engine you use. For example, this value returns either MySQL or PostgreSQL information when returning values from CreateDBInstanceReadReplica since Read Replicas are only supported for MySQL and PostgreSQL.

MySQL, SQL Server, PostgreSQL

Contains the name of the initial database of this instance that was provided at create time, if one was specified when the DB instance was created. This same name is returned for the life of the DB instance.

Type: String

Oracle

Contains the Oracle System ID (SID) of the created DB instance. Not shown when the returned parameters do not apply to an Oracle DB instance.

* **Endpoint** (*dict*) –

Specifies the connection endpoint.

· **Address** (*string*) –

Specifies the DNS address of the DB instance.

· **Port** (*integer*) –

Specifies the port that the database engine is listening on.

* **AllocatedStorage** (*integer*) –

Specifies the allocated storage size specified in gigabytes.

* **InstanceCreateTime** (*datetime*) –

Provides the date and time the DB instance was created.

* **PreferredBackupWindow** (*string*) –

Specifies the daily time range during which automated backups are created if automated backups are enabled, as determined by the `BackupRetentionPeriod`.

* **BackupRetentionPeriod** (*integer*) –

Specifies the number of days for which automatic DB snapshots are retained.

* **DBSecurityGroups** (*list*) –

Provides List of DB security group elements containing only `DBSecurityGroup.Name` and `DBSecurityGroup.Status` subelements.

· (*dict*) –

This data type is used as a response element in the following actions:

- `ModifyDBInstance`
- `RebootDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceToPointInTime`
- **DBSecurityGroupName** (*string*) –

The name of the DB security group.

· **Status** (*string*) –

The status of the DB security group.

* **VpcSecurityGroups** (*list*) –

Provides List of VPC security group elements that the DB instance belongs to.

· (*dict*) –

This data type is used as a response element for queries on VPC security group membership.

- **VpcSecurityGroupId** (*string*) –

The name of the VPC security group.

- **Status** (*string*) –

The status of the VPC security group.

- * **DBParameterGroups** (*list*) –

Provides the list of DB parameter groups applied to this DB instance.

- (*dict*) –

The status of the DB parameter group.

This data type is used as a response element in the following actions:

- CreateDBInstance
- CreateDBInstanceReadReplica
- DeleteDBInstance
- ModifyDBInstance
- RebootDBInstance
- RestoreDBInstanceFromDBSnapshot
- **DBParameterGroupName** (*string*) –

The name of the DP parameter group.

- **ParameterApplyStatus** (*string*) –

The status of parameter updates.

- * **AvailabilityZone** (*string*) –

Specifies the name of the Availability Zone the DB instance is located in.

- * **DBSubnetGroup** (*dict*) –

Specifies information on the subnet group associated with the DB instance, including the name, description, and subnets in the subnet group.

- **DBSubnetGroupName** (*string*) –

Specifies the name of the DB subnet group.

- **DBSubnetGroupDescription** (*string*) –

Provides the description of the DB subnet group.

- **VpcId** (*string*) –

Provides the VpcId of the DB subnet group.

- **SubnetGroupStatus** (*string*) –

Provides the status of the DB subnet group.

- **Subnets** (*list*) –

Contains a list of Subnet elements.

- (*dict*) –

This data type is used as a response element in the DescribeDBSubnetGroups action.

- **SubnetIdentifier** (*string*) –

Specifies the identifier of the subnet.

- **SubnetAvailabilityZone** (*dict*) –

Contains Availability Zone information.

This data type is used as an element in the following data type:

- **OrderableDBInstanceOption**

- **Name** (*string*) –

The name of the availability zone.

- **SubnetStatus** (*string*) –

Specifies the status of the subnet.

- * **PreferredMaintenanceWindow** (*string*) –

Specifies the weekly time range during which system maintenance can occur, in Universal Coordinated Time (UTC).

- * **PendingModifiedValues** (*dict*) –

Specifies that changes to the DB instance are pending. This element is only included when changes are pending. Specific changes are identified by subelements.

- **DBInstanceClass** (*string*) –

Contains the new `DBInstanceClass` for the DB instance that will be applied or is in progress.

- **AllocatedStorage** (*integer*) –

Contains the new `AllocatedStorage` size for the DB instance that will be applied or is in progress.

- **MasterUserPassword** (*string*) –

Contains the pending or in-progress change of the master credentials for the DB instance.

- **Port** (*integer*) –

Specifies the pending port for the DB instance.

- **BackupRetentionPeriod** (*integer*) –

Specifies the pending number of days for which automated backups are retained.

- **MultiAZ** (*boolean*) –

Indicates that the Single-AZ DB instance is to change to a Multi-AZ deployment.

- **EngineVersion** (*string*) –

Indicates the database engine version.

- **Iops** (*integer*) –

Specifies the new Provisioned IOPS value for the DB instance that will be applied or is being applied.

- **DBInstanceIdentifier** (*string*) –

Contains the new `DBInstanceIdentifier` for the DB instance that will be applied or is in progress.

- **StorageType** (*string*) –

Specifies the storage type to be associated with the DB instance.

- **CACertificateIdentifier** (*string*) –

Specifies the identifier of the CA certificate for the DB instance.

* **LatestRestorableTime** (*datetime*) –

Specifies the latest time to which a database can be restored with point-in-time restore.

* **MultiAZ** (*boolean*) –

Specifies if the DB instance is a Multi-AZ deployment.

* **EngineVersion** (*string*) –

Indicates the database engine version.

* **AutoMinorVersionUpgrade** (*boolean*) –

Indicates that minor version patches are applied automatically.

* **ReadReplicaSourceDBInstanceIdentifier** (*string*) –

Contains the identifier of the source DB instance if this DB instance is a Read Replica.

* **ReadReplicaDBInstanceIdentifiers** (*list*) –

Contains one or more identifiers of the Read Replicas associated with this DB instance.

· (*string*) –

* **LicenseModel** (*string*) –

License model information for this DB instance.

* **Iops** (*integer*) –

Specifies the Provisioned IOPS (I/O operations per second) value.

* **OptionGroupMemberships** (*list*) –

Provides the list of option group memberships for this DB instance.

· (*dict*) –

Provides information on the option groups the DB instance is a member of.

· **OptionGroupName** (*string*) –

The name of the option group that the instance belongs to.

· **Status** (*string*) –

The status of the DB instance's option group membership (e.g. in-sync, pending, pending-maintenance, applying).

* **CharacterSetName** (*string*) –

If present, specifies the name of the character set that this instance is associated with.

* **SecondaryAvailabilityZone** (*string*) –

If present, specifies the name of the secondary Availability Zone for a DB instance with multi-AZ support.

* **PubliclyAccessible** (*boolean*) –

Specifies the accessibility options for the DB instance. A value of true specifies an Internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. A value of false specifies an internal instance with a DNS name that resolves to a private IP address.

Default: The default behavior varies depending on whether a VPC has been requested or not. The following list shows the default behavior in each case.

- **Default VPC:** true
- **VPC:** false

If no DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be publicly accessible. If a specific DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be private.

* **StatusInfos** (*list*) –

The status of a Read Replica. If the instance is not a Read Replica, this will be blank.

- (*dict*) –

Provides a list of status information for a DB instance.

- **StatusType** (*string*) –

This value is currently “read replication.”

- **Normal** (*boolean*) –

Boolean value that is true if the instance is operating normally, or false if the instance is in an error state.

- **Status** (*string*) –

Status of the DB instance. For a StatusType of read replica, the values can be replicating, error, stopped, or terminated.

- **Message** (*string*) –

Details of the error if there is an error for the instance. If the instance is not in an error state, this value is blank.

* **StorageType** (*string*) –

Specifies the storage type associated with DB instance.

* **TdeCredentialArn** (*string*) –

The ARN from the Key Store with which the instance is associated for TDE encryption.

* **DbInstancePort** (*integer*) –

Specifies the port that the DB instance listens on. If the DB instance is part of a DB cluster, this can be a different port than the DB cluster port.

* **DBClusterIdentifier** (*string*) –

If the DB instance is a member of a DB cluster, contains the name of the DB cluster that the DB instance is a member of.

* **StorageEncrypted** (*boolean*) –

Specifies whether the DB instance is encrypted.

* **KmsKeyId** (*string*) –

If StorageEncrypted is true, the KMS key identifier for the encrypted DB instance.

* **DbiResourceId** (*string*) –

If StorageEncrypted is true, the region-unique, immutable identifier for the encrypted DB instance. This identifier is found

in AWS CloudTrail log entries whenever the KMS key for the DB instance is accessed.

* **CACertificateIdentifier** (*string*) –

The identifier of the CA certificate for this DB instance.

* **DomainMemberships** (*list*) –

The Active Directory Domain membership records associated with the DB instance.

· (*dict*) –

An Active Directory Domain membership record associated with the DB instance.

· **Domain** (*string*) –

The identifier of the Active Directory Domain.

· **Status** (*string*) –

The status of the DB instance's Active Directory Domain membership (e.g. joined, pending-join, failed etc).

· **Connectivity** (*string*) –

The observed connectivity of the Active Directory Domain.

* **CopyTagsToSnapshot** (*boolean*) –

This property is not currently implemented.

remove_source_identifier_from_subscription (***kwargs*)

Removes a source identifier from an existing RDS event notification subscription.

Request Syntax

```
response = client.remove_source_identifier_from_subscription(
    SubscriptionName='string',
    SourceIdentifier='string'
)
```

Parameters

• **SubscriptionName** (*string*) – [REQUIRED]

The name of the RDS event notification subscription you want to remove a source identifier from.

• **SourceIdentifier** (*string*) – [REQUIRED]

The source identifier to be removed from the subscription, such as the **DB instance identifier** for a DB instance or the name of a security group.

Return type dict

Returns

Response Syntax

```
{
  'EventSubscription': {
    'CustomerAwsId': 'string',
    'CustSubscriptionId': 'string',
    'SnsTopicArn': 'string',
    'Status': 'string',
    'SubscriptionCreationTime': 'string',
    'SourceType': 'string',
    'SourceIdsList': [
      'string',
```

```
    ],  
    'EventCategoriesList': [  
        'string',  
    ],  
    'Enabled': True|False  
}  
}
```

Response Structure

- (*dict*) –

Contains the results of a successful invocation of the DescribeEventSubscriptions action.

- **EventSubscription** (*dict*) –

Contains the results of a successful invocation of the DescribeEventSubscriptions action.

- * **CustomerAwsId** (*string*) –

The AWS customer account associated with the RDS event notification subscription.

- * **CustSubscriptionId** (*string*) –

The RDS event notification subscription Id.

- * **SnsTopicArn** (*string*) –

The topic ARN of the RDS event notification subscription.

- * **Status** (*string*) –

The status of the RDS event notification subscription.

Constraints:

Can be one of the following: creating | modifying | deleting | active | no-permission | topic-not-exist

The status “no-permission” indicates that RDS no longer has permission to post to the SNS topic. The status “topic-not-exist” indicates that the topic was deleted after the subscription was created.

- * **SubscriptionCreationTime** (*string*) –

The time the RDS event notification subscription was created.

- * **SourceType** (*string*) –

The source type for the RDS event notification subscription.

- * **SourceIdsList** (*list*) –

A list of source IDs for the RDS event notification subscription.

· (*string*) –

- * **EventCategoriesList** (*list*) –

A list of event categories for the RDS event notification subscription.

· (*string*) –

- * **Enabled** (*boolean*) –

A Boolean value indicating if the subscription is enabled. True indicates the subscription is enabled.

remove_tags_from_resource (***kwargs*)

Removes metadata tags from an Amazon RDS resource.

For an overview on tagging an Amazon RDS resource, see [Tagging Amazon RDS Resources](#) .

Request Syntax

```
response = client.remove_tags_from_resource(
    ResourceName='string',
    TagKeys=[
        'string',
    ]
)
```

Parameters

- **ResourceName** (*string*) – [REQUIRED]
The Amazon RDS resource the tags will be removed from. This value is an Amazon Resource Name (ARN). For information about creating an ARN, see [Constructing an RDS Amazon Resource Name \(ARN\)](#) .
- **TagKeys** (*list*) – [REQUIRED]
The tag key (name) of the tag to be removed.
– (*string*) –

Returns None

reset_db_cluster_parameter_group (***kwargs*)

Modifies the parameters of a DB cluster parameter group to the default value. To reset specific parameters submit a list of the following: `ParameterName` and `ApplyMethod` . To reset the entire DB cluster parameter group, specify the `DBClusterParameterGroupName` and `ResetAllParameters` parameters.

When resetting the entire group, dynamic parameters are updated immediately and static parameters are set to `pending-reboot` to take effect on the next DB instance restart or `RebootDBInstance` request. You must call `RebootDBInstance` for every DB instance in your DB cluster that you want the updated static parameter to apply to.

For more information on Amazon Aurora, see [Aurora on Amazon RDS](#) in the *Amazon RDS User Guide*.

Request Syntax

```
response = client.reset_db_cluster_parameter_group(
    DBClusterParameterGroupName='string',
    ResetAllParameters=True|False,
    Parameters=[
        {
            'ParameterName': 'string',
            'ParameterValue': 'string',
            'Description': 'string',
            'Source': 'string',
            'ApplyType': 'string',
            'DataType': 'string',
            'AllowedValues': 'string',
            'IsModifiable': True|False,
            'MinimumEngineVersion': 'string',
            'ApplyMethod': 'immediate'|'pending-reboot'
        },
    ]
)
```

Parameters

- **DBClusterParameterGroupName** (*string*) – [REQUIRED]

The name of the DB cluster parameter group to reset.

- **ResetAllParameters** (*boolean*) – A value that is set to `true` to reset all parameters in the DB cluster parameter group to their default values, and `false` otherwise. You cannot use this parameter if there is a list of parameter names specified for the `Parameters` parameter.
- **Parameters** (*list*) – A list of parameter names in the DB cluster parameter group to reset to the default values. You cannot use this parameter if the `ResetAllParameters` parameter is set to `true`.
 - (*dict*) –

This data type is used as a request parameter in the `ModifyDBParameterGroup` and `ResetDBParameterGroup` actions.

This data type is used as a response element in the `DescribeEngineDefaultParameters` and `DescribeDBParameters` actions.

- * **ParameterName** (*string*) –

Specifies the name of the parameter.

- * **ParameterValue** (*string*) –

Specifies the value of the parameter.

- * **Description** (*string*) –

Provides a description of the parameter.

- * **Source** (*string*) –

Indicates the source of the parameter value.

- * **ApplyType** (*string*) –

Specifies the engine specific parameters type.

- * **DataType** (*string*) –

Specifies the valid data type for the parameter.

- * **AllowedValues** (*string*) –

Specifies the valid range of values for the parameter.

- * **IsModifiable** (*boolean*) –

Indicates whether (`true`) or not (`false`) the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.

- * **MinimumEngineVersion** (*string*) –

The earliest engine version to which the parameter can apply.

- * **ApplyMethod** (*string*) –

Indicates when to apply parameter updates.

Return type `dict`

Returns

Response Syntax

```
{
    'DBClusterParameterGroupName': 'string'
}
```

Response Structure

- (*dict*) –

– **DBClusterParameterGroupName** (*string*) –

The name of the DB cluster parameter group.

Constraints:

- * Must be 1 to 255 alphanumeric characters
- * First character must be a letter
- * Cannot end with a hyphen or contain two consecutive hyphens

Note: This value is stored as a lowercase string.

reset_db_parameter_group (***kwargs*)

Modifies the parameters of a DB parameter group to the engine/system default value. To reset specific parameters submit a list of the following: `ParameterName` and `ApplyMethod`. To reset the entire DB parameter group, specify the `DBParameterGroup` name and `ResetAllParameters` parameters. When resetting the entire group, dynamic parameters are updated immediately and static parameters are set to `pending-reboot` to take effect on the next DB instance restart or `RebootDBInstance` request.

Request Syntax

```
response = client.reset_db_parameter_group(
    DBParameterGroupName='string',
    ResetAllParameters=True|False,
    Parameters=[
        {
            'ParameterName': 'string',
            'ParameterValue': 'string',
            'Description': 'string',
            'Source': 'string',
            'ApplyType': 'string',
            'DataType': 'string',
            'AllowedValues': 'string',
            'IsModifiable': True|False,
            'MinimumEngineVersion': 'string',
            'ApplyMethod': 'immediate'|'pending-reboot'
        },
    ],
)
```

Parameters

• **DBParameterGroupName** (*string*) – **[REQUIRED]**

The name of the DB parameter group.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

• **ResetAllParameters** (*boolean*) – Specifies whether (`true`) or not (`false`) to reset all parameters in the DB parameter group to default values.

Default: `true`

• **Parameters** (*list*) – An array of parameter names, values, and the apply method for the parameter update. At least one parameter name, value, and apply method must be supplied; subsequent arguments are optional. A maximum of 20 parameters can be modified in a single request.

MySQL

Valid Values (for Apply method): `immediate` | `pending-reboot`

You can use the `immediate` value with dynamic parameters only. You can use the `pending-reboot` value for both dynamic and static parameters, and changes are applied when DB instance reboots.

Oracle

Valid Values (for Apply method): `pending-reboot`

– (dict) –

This data type is used as a request parameter in the `ModifyDBParameterGroup` and `ResetDBParameterGroup` actions.

This data type is used as a response element in the `DescribeEngineDefaultParameters` and `DescribeDBParameters` actions.

- * **ParameterName** (string) –

Specifies the name of the parameter.

- * **ParameterValue** (string) –

Specifies the value of the parameter.

- * **Description** (string) –

Provides a description of the parameter.

- * **Source** (string) –

Indicates the source of the parameter value.

- * **ApplyType** (string) –

Specifies the engine specific parameters type.

- * **DataType** (string) –

Specifies the valid data type for the parameter.

- * **AllowedValues** (string) –

Specifies the valid range of values for the parameter.

- * **IsModifiable** (boolean) –

Indicates whether (`true`) or not (`false`) the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.

- * **MinimumEngineVersion** (string) –

The earliest engine version to which the parameter can apply.

- * **ApplyMethod** (string) –

Indicates when to apply parameter updates.

Return type dict

Returns

Response Syntax

```
{
    'DBParameterGroupName': 'string'
}
```

Response Structure

- (dict) –

Contains the result of a successful invocation of the `ModifyDBParameterGroup` or `ResetDBParameterGroup` action.

– **DBParameterGroupName** (*string*) –

Provides the name of the DB parameter group.

restore_db_cluster_from_snapshot (***kwargs*)

Creates a new DB cluster from a DB cluster snapshot. The target DB cluster is created from the source DB cluster restore point with the same configuration as the original source DB cluster, except that the new DB cluster is created with the default security group.

For more information on Amazon Aurora, see [Aurora on Amazon RDS](#) in the *Amazon RDS User Guide*.

Request Syntax

```
response = client.restore_db_cluster_from_snapshot(
    AvailabilityZones=[
        'string',
    ],
    DBClusterIdentifier='string',
    SnapshotIdentifier='string',
    Engine='string',
    EngineVersion='string',
    Port=123,
    DBSubnetGroupName='string',
    DatabaseName='string',
    OptionGroupName='string',
    VpcSecurityGroupIds=[
        'string',
    ],
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **AvailabilityZones** (*list*) – Provides the list of EC2 Availability Zones that instances in the restored DB cluster can be created in.
 - (*string*) –
- **DBClusterIdentifier** (*string*) – **[REQUIRED]**

The name of the DB cluster to create from the DB cluster snapshot. This parameter isn't case-sensitive.

Constraints:

- Must contain from 1 to 255 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Example: my-snapshot-id

- **SnapshotIdentifier** (*string*) – **[REQUIRED]**

The identifier for the DB cluster snapshot to restore from.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **Engine** (*string*) – **[REQUIRED]**

The database engine to use for the new DB cluster.

Default: The same as source

Constraint: Must be compatible with the engine of the source

- **EngineVersion** (*string*) – The version of the database engine to use for the new DB cluster.
- **Port** (*integer*) – The port number on which the new DB cluster accepts connections.

Constraints: Value must be 1150–65535

Default: The same port as the original DB cluster.

- **DBSubnetGroupName** (*string*) – The name of the DB subnet group to use for the new DB cluster.
- **DatabaseName** (*string*) – The database name for the restored DB cluster.
- **OptionGroupName** (*string*) – The name of the option group to use for the restored DB cluster.
- **VpcSecurityGroupIds** (*list*) – A list of VPC security groups that the new DB cluster will belong to.
 - (*string*) –
- **Tags** (*list*) – The tags to be assigned to the restored DB cluster.
 - (*dict*) –

Metadata assigned to an Amazon RDS resource consisting of a key-value pair.

* **Key** (*string*) –

A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

* **Value** (*string*) –

A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

Return type dict

Returns

Response Syntax

```
{
  'DBCluster': {
    'AllocatedStorage': 123,
    'AvailabilityZones': [
      'string',
    ],
    'BackupRetentionPeriod': 123,
    'CharacterSetName': 'string',
    'DatabaseName': 'string',
    'DBClusterIdentifier': 'string',
    'DBClusterParameterGroup': 'string',
    'DBSubnetGroup': 'string',
    'Status': 'string',
    'PercentProgress': 'string',
```

```

'EarliestRestorableTime': datetime(2015, 1, 1),
'Endpoint': 'string',
'Engine': 'string',
'EngineVersion': 'string',
'LatestRestorableTime': datetime(2015, 1, 1),
'Port': 123,
'MasterUsername': 'string',
'DBClusterOptionGroupMemberships': [
    {
        'DBClusterOptionGroupName': 'string',
        'Status': 'string'
    },
],
'PreferredBackupWindow': 'string',
'PreferredMaintenanceWindow': 'string',
'DBClusterMembers': [
    {
        'DBInstanceIdentifier': 'string',
        'IsClusterWriter': True|False,
        'DBClusterParameterGroupStatus': 'string'
    },
],
'VpcSecurityGroups': [
    {
        'VpcSecurityGroupId': 'string',
        'Status': 'string'
    },
],
]
}

```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the following actions:

- CreateDBCluster
- DeleteDBCluster
- FailoverDBCluster
- ModifyDBCluster
- RestoreDBClusterFromSnapshot

This data type is used as a response element in the DescribeDBClusters action.

- **DBCluster** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * CreateDBCluster
- * DeleteDBCluster
- * FailoverDBCluster
- * ModifyDBCluster
- * RestoreDBClusterFromSnapshot

This data type is used as a response element in the DescribeDBClusters action.

- * **AllocatedStorage** (*integer*) –

Specifies the allocated storage size in gigabytes (GB).

- * **AvailabilityZones** (*list*) –

Provides the list of EC2 Availability Zones that instances in the DB cluster can be created in.

- *(string)* –
- * **BackupRetentionPeriod** (*integer*) –
Specifies the number of days for which automatic DB snapshots are retained.
- * **CharacterSetName** (*string*) –
If present, specifies the name of the character set that this cluster is associated with.
- * **DatabaseName** (*string*) –
Contains the name of the initial database of this DB cluster that was provided at create time, if one was specified when the DB cluster was created. This same name is returned for the life of the DB cluster.
- * **DBClusterIdentifier** (*string*) –
Contains a user-supplied DB cluster identifier. This identifier is the unique key that identifies a DB cluster.
- * **DBClusterParameterGroup** (*string*) –
Specifies the name of the DB cluster parameter group for the DB cluster.
- * **DBSubnetGroup** (*string*) –
Specifies information on the subnet group associated with the DB cluster, including the name, description, and subnets in the subnet group.
- * **Status** (*string*) –
Specifies the current state of this DB cluster.
- * **PercentProgress** (*string*) –
Specifies the progress of the operation as a percentage.
- * **EarliestRestorableTime** (*datetime*) –
Specifies the earliest time to which a database can be restored with point-in-time restore.
- * **Endpoint** (*string*) –
Specifies the connection endpoint for the primary instance of the DB cluster.
- * **Engine** (*string*) –
Provides the name of the database engine to be used for this DB cluster.
- * **EngineVersion** (*string*) –
Indicates the database engine version.
- * **LatestRestorableTime** (*datetime*) –
Specifies the latest time to which a database can be restored with point-in-time restore.
- * **Port** (*integer*) –
Specifies the port that the database engine is listening on.
- * **MasterUsername** (*string*) –
Contains the master username for the DB cluster.
- * **DBClusterOptionGroupMemberships** (*list*) –
Provides the list of option group memberships for this DB cluster.

- *(dict)* –
Contains status information for a DB cluster option group.
- **DBClusterOptionGroupName** (*string*) –
Specifies the name of the DB cluster option group.
- **Status** (*string*) –
Specifies the status of the DB cluster option group.
- * **PreferredBackupWindow** (*string*) –
Specifies the daily time range during which automated backups are created if automated backups are enabled, as determined by the `BackupRetentionPeriod`.
- * **PreferredMaintenanceWindow** (*string*) –
Specifies the weekly time range during which system maintenance can occur, in Universal Coordinated Time (UTC).
- * **DBClusterMembers** (*list*) –
Provides the list of instances that make up the DB cluster.
 - *(dict)* –
Contains information about an instance that is part of a DB cluster.
 - **DBInstanceIdentifier** (*string*) –
Specifies the instance identifier for this member of the DB cluster.
 - **IsClusterWriter** (*boolean*) –
Value that is `true` if the cluster member is the primary instance for the DB cluster and `false` otherwise.
 - **DBClusterParameterGroupStatus** (*string*) –
Specifies the status of the DB cluster parameter group for this member of the DB cluster.
- * **VpcSecurityGroups** (*list*) –
Provides a list of VPC security groups that the DB cluster belongs to.
 - *(dict)* –
This data type is used as a response element for queries on VPC security group membership.
 - **VpcSecurityGroupId** (*string*) –
The name of the VPC security group.
 - **Status** (*string*) –
The status of the VPC security group.

restore_db_cluster_to_point_in_time (**kwargs)

Restores a DB cluster to an arbitrary point in time. Users can restore to any point in time before `LatestRestorableTime` for up to `BackupRetentionPeriod` days. The target DB cluster is created from the source DB cluster with the same configuration as the original DB cluster, except that the new DB cluster is created with the default DB security group.

For more information on Amazon Aurora, see [Aurora on Amazon RDS](#) in the *Amazon RDS User Guide*.

Request Syntax

```
response = client.restore_db_cluster_to_point_in_time(
    DBClusterIdentifier='string',
    SourceDBClusterIdentifier='string',
    RestoreToTime=datetime(2015, 1, 1),
    UseLatestRestorableTime=True|False,
    Port=123,
    DBSubnetGroupName='string',
    OptionGroupName='string',
    VpcSecurityGroupIds=[
        'string',
    ],
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **DBClusterIdentifier** (*string*) – [REQUIRED]

The name of the new DB cluster to be created.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **SourceDBClusterIdentifier** (*string*) – [REQUIRED]

The identifier of the source DB cluster from which to restore.

Constraints:

- Must be the identifier of an existing database instance
- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **RestoreToTime** (*datetime*) – The date and time to restore the DB cluster to.

Valid Values: Value must be a time in Universal Coordinated Time (UTC) format

Constraints:

- Must be before the latest restorable time for the DB instance
- Cannot be specified if `UseLatestRestorableTime` parameter is true

Example: 2015-03-07T23:45:00Z

- **UseLatestRestorableTime** (*boolean*) – A value that is set to `true` to restore the DB cluster to the latest restorable backup time, and `false` otherwise.

Default: `false`

Constraints: Cannot be specified if `RestoreToTime` parameter is provided.

- **Port** (*integer*) – The port number on which the new DB cluster accepts connections.

Constraints: Value must be 1150–65535

Default: The same port as the original DB cluster.

- **DBSubnetGroupName** (*string*) – The DB subnet group name to use for the new DB cluster.

- **OptionGroupName** (*string*) – The name of the option group for the new DB cluster.
- **VpcSecurityGroupIds** (*list*) – A list of VPC security groups that the new DB cluster belongs to.
 - (*string*) –
- **Tags** (*list*) – A list of tags.
 - (*dict*) –

Metadata assigned to an Amazon RDS resource consisting of a key-value pair.

- * **Key** (*string*) –

A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

- * **Value** (*string*) –

A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

Return type dict

Returns

Response Syntax

```
{
  'DBCluster': {
    'AllocatedStorage': 123,
    'AvailabilityZones': [
      'string',
    ],
    'BackupRetentionPeriod': 123,
    'CharacterSetName': 'string',
    'DatabaseName': 'string',
    'DBClusterIdentifier': 'string',
    'DBClusterParameterGroup': 'string',
    'DBSubnetGroup': 'string',
    'Status': 'string',
    'PercentProgress': 'string',
    'EarliestRestorableTime': datetime(2015, 1, 1),
    'Endpoint': 'string',
    'Engine': 'string',
    'EngineVersion': 'string',
    'LatestRestorableTime': datetime(2015, 1, 1),
    'Port': 123,
    'MasterUsername': 'string',
    'DBClusterOptionGroupMemberships': [
      {
        'DBClusterOptionGroupName': 'string',
        'Status': 'string'
      },
    ],
    'PreferredBackupWindow': 'string',
    'PreferredMaintenanceWindow': 'string',
  },
}
```

```
    'DBClusterMembers': [
        {
            'DBInstanceIdentifier': 'string',
            'IsClusterWriter': True|False,
            'DBClusterParameterGroupStatus': 'string'
        },
    ],
    'VpcSecurityGroups': [
        {
            'VpcSecurityGroupId': 'string',
            'Status': 'string'
        },
    ],
]
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the following actions:

- CreateDBCluster
- DeleteDBCluster
- FailoverDBCluster
- ModifyDBCluster
- RestoreDBClusterFromSnapshot

This data type is used as a response element in the DescribeDBClusters action.

- **DBCluster** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * CreateDBCluster
- * DeleteDBCluster
- * FailoverDBCluster
- * ModifyDBCluster
- * RestoreDBClusterFromSnapshot

This data type is used as a response element in the DescribeDBClusters action.

- * **AllocatedStorage** (*integer*) –

Specifies the allocated storage size in gigabytes (GB).

- * **AvailabilityZones** (*list*) –

Provides the list of EC2 Availability Zones that instances in the DB cluster can be created in.

· (*string*) –

- * **BackupRetentionPeriod** (*integer*) –

Specifies the number of days for which automatic DB snapshots are retained.

- * **CharacterSetName** (*string*) –

If present, specifies the name of the character set that this cluster is associated with.

- * **DatabaseName** (*string*) –

Contains the name of the initial database of this DB cluster that was provided at create time, if one was specified when the DB cluster was created. This same name is returned for the life of the DB cluster.

- * **DBClusterIdentifier** (*string*) –
Contains a user-supplied DB cluster identifier. This identifier is the unique key that identifies a DB cluster.
- * **DBClusterParameterGroup** (*string*) –
Specifies the name of the DB cluster parameter group for the DB cluster.
- * **DBSubnetGroup** (*string*) –
Specifies information on the subnet group associated with the DB cluster, including the name, description, and subnets in the subnet group.
- * **Status** (*string*) –
Specifies the current state of this DB cluster.
- * **PercentProgress** (*string*) –
Specifies the progress of the operation as a percentage.
- * **EarliestRestorableTime** (*datetime*) –
Specifies the earliest time to which a database can be restored with point-in-time restore.
- * **Endpoint** (*string*) –
Specifies the connection endpoint for the primary instance of the DB cluster.
- * **Engine** (*string*) –
Provides the name of the database engine to be used for this DB cluster.
- * **EngineVersion** (*string*) –
Indicates the database engine version.
- * **LatestRestorableTime** (*datetime*) –
Specifies the latest time to which a database can be restored with point-in-time restore.
- * **Port** (*integer*) –
Specifies the port that the database engine is listening on.
- * **MasterUsername** (*string*) –
Contains the master username for the DB cluster.
- * **DBClusterOptionGroupMemberships** (*list*) –
Provides the list of option group memberships for this DB cluster.
 - (*dict*) –
Contains status information for a DB cluster option group.
 - **DBClusterOptionGroupName** (*string*) –
Specifies the name of the DB cluster option group.
 - **Status** (*string*) –
Specifies the status of the DB cluster option group.
- * **PreferredBackupWindow** (*string*) –
Specifies the daily time range during which automated backups are created if automated backups are enabled, as determined by the `BackupRetentionPeriod`.

* **PreferredMaintenanceWindow** (*string*) –

Specifies the weekly time range during which system maintenance can occur, in Universal Coordinated Time (UTC).

* **DBClusterMembers** (*list*) –

Provides the list of instances that make up the DB cluster.

· (*dict*) –

Contains information about an instance that is part of a DB cluster.

· **DBInstanceIdentifier** (*string*) –

Specifies the instance identifier for this member of the DB cluster.

· **IsClusterWriter** (*boolean*) –

Value that is `true` if the cluster member is the primary instance for the DB cluster and `false` otherwise.

· **DBClusterParameterGroupStatus** (*string*) –

Specifies the status of the DB cluster parameter group for this member of the DB cluster.

* **VpcSecurityGroups** (*list*) –

Provides a list of VPC security groups that the DB cluster belongs to.

· (*dict*) –

This data type is used as a response element for queries on VPC security group membership.

· **VpcSecurityGroupId** (*string*) –

The name of the VPC security group.

· **Status** (*string*) –

The status of the VPC security group.

restore_db_instance_from_db_snapshot (***kwargs*)

Creates a new DB instance from a DB snapshot. The target database is created from the source database restore point with the most of original configuration, but in a system chosen availability zone with the default security group, the default subnet group, and the default DB parameter group. By default, the new DB instance is created as a single-AZ deployment except when the instance is a SQL Server instance that has an option group that is associated with mirroring; in this case, the instance becomes a mirrored AZ deployment and not a single-AZ deployment.

If your intent is to replace your original DB instance with the new, restored DB instance, then rename your original DB instance before you call the `RestoreDBInstanceFromDBSnapshot` action. RDS does not allow two DB instances with the same name. Once you have renamed your original DB instance with a different identifier, then you can pass the original name of the DB instance as the `DBInstanceIdentifier` in the call to the `RestoreDBInstanceFromDBSnapshot` action. The result is that you will replace the original DB instance with the DB instance created from the snapshot.

Request Syntax

```
response = client.restore_db_instance_from_db_snapshot(  
    DBInstanceIdentifier='string',  
    DBSnapshotIdentifier='string',  
    DBInstanceClass='string',  
    Port=123,  
    AvailabilityZone='string',
```

```

DBSubnetGroupName='string',
MultiAZ=True|False,
PubliclyAccessible=True|False,
AutoMinorVersionUpgrade=True|False,
LicenseModel='string',
DBName='string',
Engine='string',
Iops=123,
OptionGroupName='string',
Tags=[
    {
        'Key': 'string',
        'Value': 'string'
    },
],
StorageType='string',
TdeCredentialArn='string',
TdeCredentialPassword='string',
VpcSecurityGroupIds=[
    'string',
],
DBSecurityGroups=[
    'string',
],
Domain='string',
CopyTagsToSnapshot=True|False
)

```

Parameters

- **DBInstanceIdentifier** (*string*) – [REQUIRED]

Name of the DB instance to create from the DB snapshot. This parameter isn't case-sensitive.

Constraints:

- Must contain from 1 to 255 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Example: my-snapshot-id

- **DBSnapshotIdentifier** (*string*) – [REQUIRED]

The identifier for the DB snapshot to restore from.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **DBInstanceClass** (*string*) – The compute and memory capacity of the Amazon RDS DB instance.

Valid Values: db.t1.micro | db.m1.small | db.m1.medium | db.m1.large | db.m1.xlarge | db.m2.2xlarge | db.m2.4xlarge | db.m3.medium | db.m3.large | db.m3.xlarge | db.m3.2xlarge | db.r3.large | db.r3.xlarge | db.r3.2xlarge | db.r3.4xlarge | db.r3.8xlarge | db.t2.micro | db.t2.small | db.t2.medium

- **Port** (*integer*) – The port number on which the database accepts connections.

Default: The same port as the original DB instance

Constraints: Value must be 1150–65535

- **AvailabilityZone** (*string*) – The EC2 Availability Zone that the database instance will be created in.

Default: A random, system-chosen Availability Zone.

Constraint: You cannot specify the AvailabilityZone parameter if the MultiAZ parameter is set to `true`.

Example: `us-east-1a`

- **DBSubnetGroupName** (*string*) – The DB subnet group name to use for the new instance.
- **MultiAZ** (*boolean*) – Specifies if the DB instance is a Multi-AZ deployment.

Constraint: You cannot specify the AvailabilityZone parameter if the MultiAZ parameter is set to `true`.

- **PubliclyAccessible** (*boolean*) – Specifies the accessibility options for the DB instance. A value of `true` specifies an Internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. A value of `false` specifies an internal instance with a DNS name that resolves to a private IP address.

Default: The default behavior varies depending on whether a VPC has been requested or not. The following list shows the default behavior in each case.

- **Default VPC:** `true`
- **VPC:** `false`

If no DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be publicly accessible. If a specific DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be private.

- **AutoMinorVersionUpgrade** (*boolean*) – Indicates that minor version upgrades will be applied automatically to the DB instance during the maintenance window.
- **LicenseModel** (*string*) – License model information for the restored DB instance.

Default: Same as source.

Valid values: `license-included` | `bring-your-own-license` | `general-public-license`

- **DBName** (*string*) – The database name for the restored DB instance.

Note: This parameter doesn't apply to the MySQL engine.

- **Engine** (*string*) – The database engine to use for the new instance.

Default: The same as source

Constraint: Must be compatible with the engine of the source

Valid Values: `MySQL` | `oracle-se1` | `oracle-se` | `oracle-ee` | `sqlserver-ee` | `sqlserver-se` | `sqlserver-ex` | `sqlserver-web` | `postgres`

- **Iops** (*integer*) – Specifies the amount of provisioned IOPS for the DB instance, expressed in I/O operations per second. If this parameter is not specified, the IOPS value will be taken from the backup. If this parameter is set to 0, the new instance will be converted to a non-PIOPS instance, which will take additional time, though your DB instance will be available for connections before the conversion starts.

Constraints: Must be an integer greater than 1000.

SQL Server

Setting the IOPS value for the SQL Server database engine is not supported.

- **OptionGroupName** (*string*) – The name of the option group to be used for the restored DB instance.

Permanent options, such as the TDE option for Oracle Advanced Security TDE, cannot be removed from an option group, and that option group cannot be removed from a DB instance once it is associated with a DB instance

- **Tags** (*list*) – A list of tags.
 - (*dict*) –

Metadata assigned to an Amazon RDS resource consisting of a key-value pair.

- * **Key** (*string*) –

A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

- * **Value** (*string*) –

A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “^([\p{L}\p{Z}\p{N}_./=+\\-]*)\$”).

- **StorageType** (*string*) – Specifies the storage type to be associated with the DB instance.

Valid values: `standard` | `gp2` | `io1`

If you specify `io1`, you must also include a value for the `Iops` parameter.

Default: `io1` if the `Iops` parameter is specified; otherwise `standard`

- **TdeCredentialArn** (*string*) – The ARN from the Key Store with which to associate the instance for TDE encryption.
- **TdeCredentialPassword** (*string*) – The password for the given ARN from the Key Store in order to access the device.
- **VpcSecurityGroupIds** (*list*) – A list of EC2 VPC security groups to associate with this DB instance.

Default: The default EC2 VPC security group for the DB subnet group’s VPC.

– (*string*) –

- **DBSecurityGroups** (*list*) – A list of DB security groups to associate with this DB instance.

Default: The default DB security group for the database engine.

– (*string*) –

- **Domain** (*string*) – Specify the Active Directory Domain to restore the instance in.
- **CopyTagsToSnapshot** (*boolean*) – This property is not currently implemented.

Return type `dict`

Returns

Response Syntax

```
{
  'DBInstance': {
    'DBInstanceIdentifier': 'string',
    'DBInstanceClass': 'string',
    'Engine': 'string',
    'DBInstanceStatus': 'string',
    'MasterUsername': 'string',
    'DBName': 'string',
    'Endpoint': {
      'Address': 'string',
      'Port': 123
    },
    'AllocatedStorage': 123,
    'InstanceCreateTime': datetime(2015, 1, 1),
    'PreferredBackupWindow': 'string',
    'BackupRetentionPeriod': 123,
    'DBSecurityGroups': [
      {
        'DBSecurityGroupName': 'string',
        'Status': 'string'
      },
    ],
    'VpcSecurityGroups': [
      {
        'VpcSecurityGroupId': 'string',
        'Status': 'string'
      },
    ],
    'DBParameterGroups': [
      {
        'DBParameterGroupName': 'string',
        'ParameterApplyStatus': 'string'
      },
    ],
    'AvailabilityZone': 'string',
    'DBSubnetGroup': {
      'DBSubnetGroupName': 'string',
      'DBSubnetGroupDescription': 'string',
      'VpcId': 'string',
      'SubnetGroupStatus': 'string',
      'Subnets': [
        {
          'SubnetIdentifier': 'string',
          'SubnetAvailabilityZone': {
            'Name': 'string'
          },
          'SubnetStatus': 'string'
        },
      ],
    },
    'PreferredMaintenanceWindow': 'string',
    'PendingModifiedValues': {
      'DBInstanceClass': 'string',
      'AllocatedStorage': 123,
      'MasterUserPassword': 'string',
      'Port': 123,
      'BackupRetentionPeriod': 123,
      'MultiAZ': True|False,
```

```

        'EngineVersion': 'string',
        'Iops': 123,
        'DBInstanceIdentifier': 'string',
        'StorageType': 'string',
        'CACertificateIdentifier': 'string'
    },
    'LatestRestorableTime': datetime(2015, 1, 1),
    'MultiAZ': True|False,
    'EngineVersion': 'string',
    'AutoMinorVersionUpgrade': True|False,
    'ReadReplicaSourceDBInstanceIdentifier': 'string',
    'ReadReplicaDBInstanceIdentifiers': [
        'string',
    ],
    'LicenseModel': 'string',
    'Iops': 123,
    'OptionGroupMemberships': [
        {
            'OptionGroupName': 'string',
            'Status': 'string'
        },
    ],
    'CharacterSetName': 'string',
    'SecondaryAvailabilityZone': 'string',
    'PubliclyAccessible': True|False,
    'StatusInfos': [
        {
            'StatusType': 'string',
            'Normal': True|False,
            'Status': 'string',
            'Message': 'string'
        },
    ],
    'StorageType': 'string',
    'TdeCredentialArn': 'string',
    'DbInstancePort': 123,
    'DBClusterIdentifier': 'string',
    'StorageEncrypted': True|False,
    'KmsKeyId': 'string',
    'DbiResourceId': 'string',
    'CACertificateIdentifier': 'string',
    'DomainMemberships': [
        {
            'Domain': 'string',
            'Status': 'string',
            'Connectivity': 'string'
        },
    ],
    'CopyTagsToSnapshot': True|False
}
}

```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the following actions:

- CreateDBInstance
- DeleteDBInstance

- ModifyDBInstance

This data type is used as a response element in the DescribeDBInstances action.

- **DBInstance** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * CreateDBInstance
- * DeleteDBInstance
- * ModifyDBInstance

This data type is used as a response element in the DescribeDBInstances action.

- * **DBInstanceIdentifier** (*string*) –

Contains a user-supplied database identifier. This identifier is the unique key that identifies a DB instance.

- * **DBInstanceClass** (*string*) –

Contains the name of the compute and memory capacity class of the DB instance.

- * **Engine** (*string*) –

Provides the name of the database engine to be used for this DB instance.

- * **DBInstanceStatus** (*string*) –

Specifies the current state of this database.

- * **MasterUsername** (*string*) –

Contains the master username for the DB instance.

- * **DBName** (*string*) –

The meaning of this parameter differs according to the database engine you use. For example, this value returns either MySQL or PostgreSQL information when returning values from CreateDBInstanceReadReplica since Read Replicas are only supported for MySQL and PostgreSQL.

MySQL, SQL Server, PostgreSQL

Contains the name of the initial database of this instance that was provided at create time, if one was specified when the DB instance was created. This same name is returned for the life of the DB instance.

Type: String

Oracle

Contains the Oracle System ID (SID) of the created DB instance. Not shown when the returned parameters do not apply to an Oracle DB instance.

- * **Endpoint** (*dict*) –

Specifies the connection endpoint.

- **Address** (*string*) –

Specifies the DNS address of the DB instance.

- **Port** (*integer*) –

Specifies the port that the database engine is listening on.

- * **AllocatedStorage** (*integer*) –

Specifies the allocated storage size specified in gigabytes.

- * **InstanceCreateTime** (*datetime*) –

Provides the date and time the DB instance was created.

* **PreferredBackupWindow** (*string*) –

Specifies the daily time range during which automated backups are created if automated backups are enabled, as determined by the `BackupRetentionPeriod`.

* **BackupRetentionPeriod** (*integer*) –

Specifies the number of days for which automatic DB snapshots are retained.

* **DBSecurityGroups** (*list*) –

Provides List of DB security group elements containing only `DBSecurityGroup.Name` and `DBSecurityGroup.Status` subelements.

· (*dict*) –

This data type is used as a response element in the following actions:

- `ModifyDBInstance`
- `RebootDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceToPointInTime`
- **DBSecurityGroupName** (*string*) –

The name of the DB security group.

- **Status** (*string*) –

The status of the DB security group.

* **VpcSecurityGroups** (*list*) –

Provides List of VPC security group elements that the DB instance belongs to.

· (*dict*) –

This data type is used as a response element for queries on VPC security group membership.

- **VpcSecurityGroupId** (*string*) –

The name of the VPC security group.

- **Status** (*string*) –

The status of the VPC security group.

* **DBParameterGroups** (*list*) –

Provides the list of DB parameter groups applied to this DB instance.

· (*dict*) –

The status of the DB parameter group.

This data type is used as a response element in the following actions:

- `CreateDBInstance`
- `CreateDBInstanceReadReplica`
- `DeleteDBInstance`
- `ModifyDBInstance`
- `RebootDBInstance`
- `RestoreDBInstanceFromDBSnapshot`

- **DBParameterGroupName** (*string*) –
The name of the DP parameter group.
- **ParameterApplyStatus** (*string*) –
The status of parameter updates.
- * **AvailabilityZone** (*string*) –
Specifies the name of the Availability Zone the DB instance is located in.
- * **DBSubnetGroup** (*dict*) –
Specifies information on the subnet group associated with the DB instance, including the name, description, and subnets in the subnet group.
 - **DBSubnetGroupName** (*string*) –
Specifies the name of the DB subnet group.
 - **DBSubnetGroupDescription** (*string*) –
Provides the description of the DB subnet group.
 - **VpcId** (*string*) –
Provides the VpcId of the DB subnet group.
 - **SubnetGroupStatus** (*string*) –
Provides the status of the DB subnet group.
 - **Subnets** (*list*) –
Contains a list of Subnet elements.
 - (*dict*) –
This data type is used as a response element in the DescribeDBSubnetGroups action.
 - **SubnetIdentifier** (*string*) –
Specifies the identifier of the subnet.
 - **SubnetAvailabilityZone** (*dict*) –
Contains Availability Zone information.
This data type is used as an element in the following data type:
 - **OrderableDBInstanceOption**
 - **Name** (*string*) –
The name of the availability zone.
 - **SubnetStatus** (*string*) –
Specifies the status of the subnet.
- * **PreferredMaintenanceWindow** (*string*) –
Specifies the weekly time range during which system maintenance can occur, in Universal Coordinated Time (UTC).
- * **PendingModifiedValues** (*dict*) –
Specifies that changes to the DB instance are pending. This element is only included when changes are pending. Specific changes are identified by subelements.
 - **DBInstanceClass** (*string*) –
Contains the new DBInstanceClass for the DB instance that will be applied or is in progress.

- **AllocatedStorage** (*integer*) –
Contains the new `AllocatedStorage` size for the DB instance that will be applied or is in progress.
- **MasterUserPassword** (*string*) –
Contains the pending or in-progress change of the master credentials for the DB instance.
- **Port** (*integer*) –
Specifies the pending port for the DB instance.
- **BackupRetentionPeriod** (*integer*) –
Specifies the pending number of days for which automated backups are retained.
- **MultiAZ** (*boolean*) –
Indicates that the Single-AZ DB instance is to change to a Multi-AZ deployment.
- **EngineVersion** (*string*) –
Indicates the database engine version.
- **Iops** (*integer*) –
Specifies the new Provisioned IOPS value for the DB instance that will be applied or is being applied.
- **DBInstanceIdentifier** (*string*) –
Contains the new `DBInstanceIdentifier` for the DB instance that will be applied or is in progress.
- **StorageType** (*string*) –
Specifies the storage type to be associated with the DB instance.
- **CACertificateIdentifier** (*string*) –
Specifies the identifier of the CA certificate for the DB instance.
- * **LatestRestorableTime** (*datetime*) –
Specifies the latest time to which a database can be restored with point-in-time restore.
- * **MultiAZ** (*boolean*) –
Specifies if the DB instance is a Multi-AZ deployment.
- * **EngineVersion** (*string*) –
Indicates the database engine version.
- * **AutoMinorVersionUpgrade** (*boolean*) –
Indicates that minor version patches are applied automatically.
- * **ReadReplicaSourceDBInstanceIdentifier** (*string*) –
Contains the identifier of the source DB instance if this DB instance is a Read Replica.
- * **ReadReplicaDBInstanceIdentifiers** (*list*) –
Contains one or more identifiers of the Read Replicas associated with this DB instance.
 - (*string*) –

* **LicenseModel** (*string*) –

License model information for this DB instance.

* **Iops** (*integer*) –

Specifies the Provisioned IOPS (I/O operations per second) value.

* **OptionGroupMemberships** (*list*) –

Provides the list of option group memberships for this DB instance.

· (*dict*) –

Provides information on the option groups the DB instance is a member of.

· **OptionGroupName** (*string*) –

The name of the option group that the instance belongs to.

· **Status** (*string*) –

The status of the DB instance's option group membership (e.g. in-sync, pending, pending-maintenance, applying).

* **CharacterSetName** (*string*) –

If present, specifies the name of the character set that this instance is associated with.

* **SecondaryAvailabilityZone** (*string*) –

If present, specifies the name of the secondary Availability Zone for a DB instance with multi-AZ support.

* **PubliclyAccessible** (*boolean*) –

Specifies the accessibility options for the DB instance. A value of true specifies an Internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. A value of false specifies an internal instance with a DNS name that resolves to a private IP address.

Default: The default behavior varies depending on whether a VPC has been requested or not. The following list shows the default behavior in each case.

· **Default VPC:** true

· **VPC:** false

If no DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be publicly accessible. If a specific DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be private.

* **StatusInfos** (*list*) –

The status of a Read Replica. If the instance is not a Read Replica, this will be blank.

· (*dict*) –

Provides a list of status information for a DB instance.

· **StatusType** (*string*) –

This value is currently “read replication.”

· **Normal** (*boolean*) –

Boolean value that is true if the instance is operating normally, or false if the instance is in an error state.

- **Status** (*string*) –
Status of the DB instance. For a `StatusType` of read replica, the values can be replicating, error, stopped, or terminated.
- **Message** (*string*) –
Details of the error if there is an error for the instance. If the instance is not in an error state, this value is blank.
- * **StorageType** (*string*) –
Specifies the storage type associated with DB instance.
- * **TdeCredentialArn** (*string*) –
The ARN from the Key Store with which the instance is associated for TDE encryption.
- * **DbInstancePort** (*integer*) –
Specifies the port that the DB instance listens on. If the DB instance is part of a DB cluster, this can be a different port than the DB cluster port.
- * **DBClusterIdentifier** (*string*) –
If the DB instance is a member of a DB cluster, contains the name of the DB cluster that the DB instance is a member of.
- * **StorageEncrypted** (*boolean*) –
Specifies whether the DB instance is encrypted.
- * **KmsKeyId** (*string*) –
If `StorageEncrypted` is true, the KMS key identifier for the encrypted DB instance.
- * **DbiResourceId** (*string*) –
If `StorageEncrypted` is true, the region-unique, immutable identifier for the encrypted DB instance. This identifier is found in AWS CloudTrail log entries whenever the KMS key for the DB instance is accessed.
- * **CACertificateIdentifier** (*string*) –
The identifier of the CA certificate for this DB instance.
- * **DomainMemberships** (*list*) –
The Active Directory Domain membership records associated with the DB instance.
 - (*dict*) –
An Active Directory Domain membership record associated with the DB instance.
 - **Domain** (*string*) –
The identifier of the Active Directory Domain.
 - **Status** (*string*) –
The status of the DB instance's Active Directory Domain membership (e.g. joined, pending-join, failed etc).
 - **Connectivity** (*string*) –
The observed connectivity of the Active Directory Domain.
- * **CopyTagsToSnapshot** (*boolean*) –
This property is not currently implemented.

restore_db_instance_to_point_in_time (***kwargs*)

Restores a DB instance to an arbitrary point-in-time. Users can restore to any point in time before the LatestRestorableTime for up to BackupRetentionPeriod days. The target database is created with the most of original configuration, but in a system chosen availability zone with the default security group, the default subnet group, and the default DB parameter group. By default, the new DB instance is created as a single-AZ deployment except when the instance is a SQL Server instance that has an option group that is associated with mirroring; in this case, the instance becomes a mirrored deployment and not a single-AZ deployment.

Request Syntax

```
response = client.restore_db_instance_to_point_in_time(
    SourceDBInstanceIdentifier='string',
    TargetDBInstanceIdentifier='string',
    RestoreTime=datetime(2015, 1, 1),
    UseLatestRestorableTime=True|False,
    DBInstanceClass='string',
    Port=123,
    AvailabilityZone='string',
    DBSubnetGroupName='string',
    MultiAZ=True|False,
    PubliclyAccessible=True|False,
    AutoMinorVersionUpgrade=True|False,
    LicenseModel='string',
    DBName='string',
    Engine='string',
    Iops=123,
    OptionGroupName='string',
    CopyTagsToSnapshot=True|False,
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
    StorageType='string',
    TdeCredentialArn='string',
    TdeCredentialPassword='string',
    VpcSecurityGroupIds=[
        'string',
    ],
    DBSecurityGroups=[
        'string',
    ],
    Domain='string'
)
```

Parameters

- **SourceDBInstanceIdentifier** (*string*) – [REQUIRED]

The identifier of the source DB instance from which to restore.

Constraints:

- Must be the identifier of an existing database instance
 - Must contain from 1 to 63 alphanumeric characters or hyphens
 - First character must be a letter
 - Cannot end with a hyphen or contain two consecutive hyphens
- **TargetDBInstanceIdentifier** (*string*) – [REQUIRED]

The name of the new database instance to be created.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens
- **RestoreTime** (*datetime*) – The date and time to restore from.

Valid Values: Value must be a time in Universal Coordinated Time (UTC) format

Constraints:

- Must be before the latest restorable time for the DB instance
- Cannot be specified if UseLatestRestorableTime parameter is true

Example: 2009-09-07T23:45:00Z

- **UseLatestRestorableTime** (*boolean*) – Specifies whether (`true`) or not (`false`) the DB instance is restored from the latest backup time.

Default: `false`

Constraints: Cannot be specified if RestoreTime parameter is provided.

- **DBInstanceClass** (*string*) – The compute and memory capacity of the Amazon RDS DB instance.

Valid Values: `db.t1.micro` | `db.m1.small` | `db.m1.medium` | `db.m1.large` | `db.m1.xlarge` | `db.m2.2xlarge` | `db.m2.4xlarge` | `db.m3.medium` | `db.m3.large` | `db.m3.xlarge` | `db.m3.2xlarge` | `db.r3.large` | `db.r3.xlarge` | `db.r3.2xlarge` | `db.r3.4xlarge` | `db.r3.8xlarge` | `db.t2.micro` | `db.t2.small` | `db.t2.medium`

Default: The same DBInstanceClass as the original DB instance.

- **Port** (*integer*) – The port number on which the database accepts connections.

Constraints: Value must be 1150–65535

Default: The same port as the original DB instance.

- **AvailabilityZone** (*string*) – The EC2 Availability Zone that the database instance will be created in.

Default: A random, system-chosen Availability Zone.

Constraint: You cannot specify the AvailabilityZone parameter if the MultiAZ parameter is set to `true`.

Example: `us-east-1a`

- **DBSubnetGroupName** (*string*) – The DB subnet group name to use for the new instance.
- **MultiAZ** (*boolean*) – Specifies if the DB instance is a Multi-AZ deployment.

Constraint: You cannot specify the AvailabilityZone parameter if the MultiAZ parameter is set to `true`.

- **PubliclyAccessible** (*boolean*) – Specifies the accessibility options for the DB instance. A value of `true` specifies an Internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. A value of `false` specifies an internal instance with a DNS name that resolves to a private IP address.

Default: The default behavior varies depending on whether a VPC has been requested or not. The following list shows the default behavior in each case.

- **Default VPC:** `true`

– **VPC**: false

If no DB subnet group has been specified as part of the request and the Publicly-Accessible value has not been set, the DB instance will be publicly accessible. If a specific DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be private.

- **AutoMinorVersionUpgrade** (*boolean*) – Indicates that minor version upgrades will be applied automatically to the DB instance during the maintenance window.
- **LicenseModel** (*string*) – License model information for the restored DB instance.

Default: Same as source.

Valid values: `license-included` | `bring-your-own-license` | `general-public-license`

- **DBName** (*string*) – The database name for the restored DB instance.

Note: This parameter is not used for the MySQL engine.

- **Engine** (*string*) – The database engine to use for the new instance.

Default: The same as source

Constraint: Must be compatible with the engine of the source

Valid Values: `MySQL` | `oracle-se1` | `oracle-se` | `oracle-ee` | `sqlserver-ee` | `sqlserver-se` | `sqlserver-ex` | `sqlserver-web` | `postgres`

- **Iops** (*integer*) – The amount of Provisioned IOPS (input/output operations per second) to be initially allocated for the DB instance.

Constraints: Must be an integer greater than 1000.

SQL Server

Setting the IOPS value for the SQL Server database engine is not supported.

- **OptionGroupName** (*string*) – The name of the option group to be used for the restored DB instance.

Permanent options, such as the TDE option for Oracle Advanced Security TDE, cannot be removed from an option group, and that option group cannot be removed from a DB instance once it is associated with a DB instance

- **CopyTagsToSnapshot** (*boolean*) – This property is not currently implemented.
- **Tags** (*list*) – A list of tags.
 - (*dict*) –

Metadata assigned to an Amazon RDS resource consisting of a key-value pair.

* **Key** (*string*) –

A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “`^([\p{L}\p{Z}\p{N}_./=+\\-]*)$`”).

* **Value** (*string*) –

A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with “aws:” or “rds:”. The string can only contain only the set of Unicode letters, digits, white-space, ‘_’, ‘.’, ‘/’, ‘=’, ‘+’, ‘-’ (Java regex: “`^([\p{L}\p{Z}\p{N}_./=+\\-]*)$`”).

- **StorageType** (*string*) – Specifies the storage type to be associated with the DB instance.

Valid values: `standard` | `gp2` | `io1`

If you specify `io1`, you must also include a value for the `Iops` parameter.

Default: `io1` if the `Iops` parameter is specified; otherwise `standard`

- **TdeCredentialArn** (*string*) – The ARN from the Key Store with which to associate the instance for TDE encryption.
- **TdeCredentialPassword** (*string*) – The password for the given ARN from the Key Store in order to access the device.
- **VpcSecurityGroupIds** (*list*) – A list of EC2 VPC security groups to associate with this DB instance.

Default: The default EC2 VPC security group for the DB subnet group's VPC.

– (*string*) –

- **DBSecurityGroups** (*list*) – A list of DB security groups to associate with this DB instance.

Default: The default DB security group for the database engine.

– (*string*) –

- **Domain** (*string*) – Specify the Active Directory Domain to restore the instance in.

Return type dict

Returns

Response Syntax

```
{
  'DBInstance': {
    'DBInstanceIdentifier': 'string',
    'DBInstanceClass': 'string',
    'Engine': 'string',
    'DBInstanceStatus': 'string',
    'MasterUsername': 'string',
    'DBName': 'string',
    'Endpoint': {
      'Address': 'string',
      'Port': 123
    },
    'AllocatedStorage': 123,
    'InstanceCreateTime': datetime(2015, 1, 1),
    'PreferredBackupWindow': 'string',
    'BackupRetentionPeriod': 123,
    'DBSecurityGroups': [
      {
        'DBSecurityGroupName': 'string',
        'Status': 'string'
      },
    ],
    'VpcSecurityGroups': [
      {
        'VpcSecurityGroupId': 'string',
        'Status': 'string'
      },
    ],
    'DBParameterGroups': [
      {
```

```
        'DBParameterGroupName': 'string',
        'ParameterApplyStatus': 'string'
    },
],
'AvailabilityZone': 'string',
'DBSubnetGroup': {
    'DBSubnetGroupName': 'string',
    'DBSubnetGroupDescription': 'string',
    'VpcId': 'string',
    'SubnetGroupStatus': 'string',
    'Subnets': [
        {
            'SubnetIdentifier': 'string',
            'SubnetAvailabilityZone': {
                'Name': 'string'
            },
            'SubnetStatus': 'string'
        },
    ]
},
'PreferredMaintenanceWindow': 'string',
'PendingModifiedValues': {
    'DBInstanceClass': 'string',
    'AllocatedStorage': 123,
    'MasterUserPassword': 'string',
    'Port': 123,
    'BackupRetentionPeriod': 123,
    'MultiAZ': True|False,
    'EngineVersion': 'string',
    'Iops': 123,
    'DBInstanceIdentifier': 'string',
    'StorageType': 'string',
    'CACertificateIdentifier': 'string'
},
'LatestRestorableTime': datetime(2015, 1, 1),
'MultiAZ': True|False,
'EngineVersion': 'string',
'AutoMinorVersionUpgrade': True|False,
'ReadReplicaSourceDBInstanceIdentifier': 'string',
'ReadReplicaDBInstanceIdentifiers': [
    'string',
],
'LicenseModel': 'string',
'Iops': 123,
'OptionGroupMemberships': [
    {
        'OptionGroupName': 'string',
        'Status': 'string'
    },
],
'CharacterSetName': 'string',
'SecondaryAvailabilityZone': 'string',
'PubliclyAccessible': True|False,
'StatusInfos': [
    {
        'StatusType': 'string',
        'Normal': True|False,
        'Status': 'string',
    },
]
```

```

        'Message': 'string'
    },
],
'StorageType': 'string',
'TdeCredentialArn': 'string',
'DbInstancePort': 123,
'DBClusterIdentifier': 'string',
'StorageEncrypted': True|False,
'KmsKeyId': 'string',
'DbiResourceId': 'string',
'CACertificateIdentifier': 'string',
'DomainMemberships': [
    {
        'Domain': 'string',
        'Status': 'string',
        'Connectivity': 'string'
    },
],
'CopyTagsToSnapshot': True|False
}

```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the following actions:

- CreateDBInstance
- DeleteDBInstance
- ModifyDBInstance

This data type is used as a response element in the DescribeDBInstances action.

- **DBInstance** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * CreateDBInstance
- * DeleteDBInstance
- * ModifyDBInstance

This data type is used as a response element in the DescribeDBInstances action.

- * **DBInstanceIdentifier** (*string*) –

Contains a user-supplied database identifier. This identifier is the unique key that identifies a DB instance.

- * **DBInstanceClass** (*string*) –

Contains the name of the compute and memory capacity class of the DB instance.

- * **Engine** (*string*) –

Provides the name of the database engine to be used for this DB instance.

- * **DBInstanceStatus** (*string*) –

Specifies the current state of this database.

- * **MasterUsername** (*string*) –

Contains the master username for the DB instance.

- * **DBName** (*string*) –

The meaning of this parameter differs according to the database

engine you use. For example, this value returns either MySQL or PostgreSQL information when returning values from `CreateDBInstanceReadReplica` since Read Replicas are only supported for MySQL and PostgreSQL.

MySQL, SQL Server, PostgreSQL

Contains the name of the initial database of this instance that was provided at create time, if one was specified when the DB instance was created. This same name is returned for the life of the DB instance.

Type: String

Oracle

Contains the Oracle System ID (SID) of the created DB instance. Not shown when the returned parameters do not apply to an Oracle DB instance.

* **Endpoint** (*dict*) –

Specifies the connection endpoint.

· **Address** (*string*) –

Specifies the DNS address of the DB instance.

· **Port** (*integer*) –

Specifies the port that the database engine is listening on.

* **AllocatedStorage** (*integer*) –

Specifies the allocated storage size specified in gigabytes.

* **InstanceCreateTime** (*datetime*) –

Provides the date and time the DB instance was created.

* **PreferredBackupWindow** (*string*) –

Specifies the daily time range during which automated backups are created if automated backups are enabled, as determined by the `BackupRetentionPeriod`.

* **BackupRetentionPeriod** (*integer*) –

Specifies the number of days for which automatic DB snapshots are retained.

* **DBSecurityGroups** (*list*) –

Provides List of DB security group elements containing only `DBSecurityGroup.Name` and `DBSecurityGroup.Status` subelements.

· (*dict*) –

This data type is used as a response element in the following actions:

- `ModifyDBInstance`
- `RebootDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceToPointInTime`
- **DBSecurityGroupName** (*string*) –

The name of the DB security group.

· **Status** (*string*) –

The status of the DB security group.

* **VpcSecurityGroups** (*list*) –

Provides List of VPC security group elements that the DB instance belongs to.

- *(dict)* –

This data type is used as a response element for queries on VPC security group membership.

- **VpcSecurityGroupId** (*string*) –

The name of the VPC security group.

- **Status** (*string*) –

The status of the VPC security group.

- * **DBParameterGroups** (*list*) –

Provides the list of DB parameter groups applied to this DB instance.

- *(dict)* –

The status of the DB parameter group.

This data type is used as a response element in the following actions:

- CreateDBInstance
- CreateDBInstanceReadReplica
- DeleteDBInstance
- ModifyDBInstance
- RebootDBInstance
- RestoreDBInstanceFromDBSnapshot
- **DBParameterGroupName** (*string*) –

The name of the DP parameter group.

- **ParameterApplyStatus** (*string*) –

The status of parameter updates.

- * **AvailabilityZone** (*string*) –

Specifies the name of the Availability Zone the DB instance is located in.

- * **DBSubnetGroup** (*dict*) –

Specifies information on the subnet group associated with the DB instance, including the name, description, and subnets in the subnet group.

- **DBSubnetGroupName** (*string*) –

Specifies the name of the DB subnet group.

- **DBSubnetGroupDescription** (*string*) –

Provides the description of the DB subnet group.

- **VpcId** (*string*) –

Provides the VpcId of the DB subnet group.

- **SubnetGroupStatus** (*string*) –

Provides the status of the DB subnet group.

- **Subnets** (*list*) –

Contains a list of Subnet elements.

- *(dict)* –

This data type is used as a response element in the DescribeDBSubnetGroups action.

- **SubnetIdentifier** (*string*) –
Specifies the identifier of the subnet.
- **SubnetAvailabilityZone** (*dict*) –
Contains Availability Zone information.

This data type is used as an element in the following data type:
 - OrderableDBInstanceOption
- **Name** (*string*) –
The name of the availability zone.
- **SubnetStatus** (*string*) –
Specifies the status of the subnet.
- * **PreferredMaintenanceWindow** (*string*) –
Specifies the weekly time range during which system maintenance can occur, in Universal Coordinated Time (UTC).
- * **PendingModifiedValues** (*dict*) –
Specifies that changes to the DB instance are pending. This element is only included when changes are pending. Specific changes are identified by subelements.
 - **DBInstanceClass** (*string*) –
Contains the new DBInstanceClass for the DB instance that will be applied or is in progress.
 - **AllocatedStorage** (*integer*) –
Contains the new AllocatedStorage size for the DB instance that will be applied or is in progress.
 - **MasterUserPassword** (*string*) –
Contains the pending or in-progress change of the master credentials for the DB instance.
 - **Port** (*integer*) –
Specifies the pending port for the DB instance.
 - **BackupRetentionPeriod** (*integer*) –
Specifies the pending number of days for which automated backups are retained.
 - **MultiAZ** (*boolean*) –
Indicates that the Single-AZ DB instance is to change to a Multi-AZ deployment.
 - **EngineVersion** (*string*) –
Indicates the database engine version.
 - **Iops** (*integer*) –
Specifies the new Provisioned IOPS value for the DB instance that will be applied or is being applied.
 - **DBInstanceIdentifier** (*string*) –
Contains the new DBInstanceIdentifier for the DB instance that will be applied or is in progress.
 - **StorageType** (*string*) –
Specifies the storage type to be associated with the DB instance.

- **CACertificateIdentifier** (*string*) –
Specifies the identifier of the CA certificate for the DB instance.
- * **LatestRestorableTime** (*datetime*) –
Specifies the latest time to which a database can be restored with point-in-time restore.
- * **MultiAZ** (*boolean*) –
Specifies if the DB instance is a Multi-AZ deployment.
- * **EngineVersion** (*string*) –
Indicates the database engine version.
- * **AutoMinorVersionUpgrade** (*boolean*) –
Indicates that minor version patches are applied automatically.
- * **ReadReplicaSourceDBInstanceIdentifier** (*string*) –
Contains the identifier of the source DB instance if this DB instance is a Read Replica.
- * **ReadReplicaDBInstanceIdentifiers** (*list*) –
Contains one or more identifiers of the Read Replicas associated with this DB instance.
 - (*string*) –
- * **LicenseModel** (*string*) –
License model information for this DB instance.
- * **Iops** (*integer*) –
Specifies the Provisioned IOPS (I/O operations per second) value.
- * **OptionGroupMemberships** (*list*) –
Provides the list of option group memberships for this DB instance.
 - (*dict*) –
Provides information on the option groups the DB instance is a member of.
 - **OptionGroupName** (*string*) –
The name of the option group that the instance belongs to.
 - **Status** (*string*) –
The status of the DB instance's option group membership (e.g. in-sync, pending, pending-maintenance, applying).
- * **CharacterSetName** (*string*) –
If present, specifies the name of the character set that this instance is associated with.
- * **SecondaryAvailabilityZone** (*string*) –
If present, specifies the name of the secondary Availability Zone for a DB instance with multi-AZ support.
- * **PubliclyAccessible** (*boolean*) –
Specifies the accessibility options for the DB instance. A value of true specifies an Internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. A value of false specifies an internal instance with a DNS name that resolves to a private IP address.

Default: The default behavior varies depending on whether a VPC has been requested or not. The following list shows the default behavior in each case.

- **Default VPC:** true
- **VPC:** false

If no DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be publicly accessible. If a specific DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be private.

* **StatusInfos** (*list*) –

The status of a Read Replica. If the instance is not a Read Replica, this will be blank.

- (*dict*) –

Provides a list of status information for a DB instance.

- **StatusType** (*string*) –

This value is currently “read replication.”

- **Normal** (*boolean*) –

Boolean value that is true if the instance is operating normally, or false if the instance is in an error state.

- **Status** (*string*) –

Status of the DB instance. For a StatusType of read replica, the values can be replicating, error, stopped, or terminated.

- **Message** (*string*) –

Details of the error if there is an error for the instance. If the instance is not in an error state, this value is blank.

* **StorageType** (*string*) –

Specifies the storage type associated with DB instance.

* **TdeCredentialArn** (*string*) –

The ARN from the Key Store with which the instance is associated for TDE encryption.

* **DbInstancePort** (*integer*) –

Specifies the port that the DB instance listens on. If the DB instance is part of a DB cluster, this can be a different port than the DB cluster port.

* **DBClusterIdentifier** (*string*) –

If the DB instance is a member of a DB cluster, contains the name of the DB cluster that the DB instance is a member of.

* **StorageEncrypted** (*boolean*) –

Specifies whether the DB instance is encrypted.

* **KmsKeyId** (*string*) –

If StorageEncrypted is true, the KMS key identifier for the encrypted DB instance.

* **DbiResourceId** (*string*) –

If StorageEncrypted is true, the region-unique, immutable identifier for the encrypted DB instance. This identifier is found

in AWS CloudTrail log entries whenever the KMS key for the DB instance is accessed.

* **CACertificateIdentifier** (*string*) –

The identifier of the CA certificate for this DB instance.

* **DomainMemberships** (*list*) –

The Active Directory Domain membership records associated with the DB instance.

· (*dict*) –

An Active Directory Domain membership record associated with the DB instance.

· **Domain** (*string*) –

The identifier of the Active Directory Domain.

· **Status** (*string*) –

The status of the DB instance's Active Directory Domain membership (e.g. joined, pending-join, failed etc).

· **Connectivity** (*string*) –

The observed connectivity of the Active Directory Domain.

* **CopyTagsToSnapshot** (*boolean*) –

This property is not currently implemented.

revoke_db_security_group_ingress (***kwargs*)

Revokes ingress from a DBSecurityGroup for previously authorized IP ranges or EC2 or VPC Security Groups. Required parameters for this API are one of CIDRIP, EC2SecurityGroupId for VPC, or (EC2SecurityGroupOwnerId and either EC2SecurityGroupName or EC2SecurityGroupId).

Request Syntax

```
response = client.revoke_db_security_group_ingress(
    DBSecurityGroupName='string',
    CIDRIP='string',
    EC2SecurityGroupName='string',
    EC2SecurityGroupId='string',
    EC2SecurityGroupOwnerId='string'
)
```

Parameters

• **DBSecurityGroupName** (*string*) – [REQUIRED]

The name of the DB security group to revoke ingress from.

• **CIDRIP** (*string*) – The IP range to revoke access from. Must be a valid CIDR range. If CIDRIP is specified, EC2SecurityGroupName, EC2SecurityGroupId and EC2SecurityGroupOwnerId cannot be provided.

• **EC2SecurityGroupName** (*string*) – The name of the EC2 security group to revoke access from. For VPC DB security groups, EC2SecurityGroupId must be provided. Otherwise, EC2SecurityGroupOwnerId and either EC2SecurityGroupName or EC2SecurityGroupId must be provided.

• **EC2SecurityGroupId** (*string*) – The id of the EC2 security group to revoke access from. For VPC DB security groups, EC2SecurityGroupId must be provided. Otherwise, EC2SecurityGroupOwnerId and either EC2SecurityGroupName or EC2SecurityGroupId must be provided.

• **EC2SecurityGroupOwnerId** (*string*) – The AWS Account Number of the owner of the EC2 security group specified in the EC2SecurityGroupName

parameter. The AWS Access Key ID is not an acceptable value. For VPC DB security groups, `EC2SecurityGroupId` must be provided. Otherwise, `EC2SecurityGroupOwnerId` and either `EC2SecurityGroupName` or `EC2SecurityGroupId` must be provided.

Return type dict

Returns

Response Syntax

```
{
  'DBSecurityGroup': {
    'OwnerId': 'string',
    'DBSecurityGroupName': 'string',
    'DBSecurityGroupDescription': 'string',
    'VpcId': 'string',
    'EC2SecurityGroups': [
      {
        'Status': 'string',
        'EC2SecurityGroupName': 'string',
        'EC2SecurityGroupId': 'string',
        'EC2SecurityGroupOwnerId': 'string'
      },
    ],
    'IPRanges': [
      {
        'Status': 'string',
        'CIDRIP': 'string'
      },
    ],
  }
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the following actions:

- DescribeDBSecurityGroups
- AuthorizeDBSecurityGroupIngress
- CreateDBSecurityGroup
- RevokeDBSecurityGroupIngress

This data type is used as a response element in the DescribeDBSecurityGroups action.

- **DBSecurityGroup** (*dict*) –

Contains the result of a successful invocation of the following actions:

- * DescribeDBSecurityGroups
- * AuthorizeDBSecurityGroupIngress
- * CreateDBSecurityGroup
- * RevokeDBSecurityGroupIngress

This data type is used as a response element in the DescribeDBSecurityGroups action.

- * **OwnerId** (*string*) –

Provides the AWS ID of the owner of a specific DB security group.

- * **DBSecurityGroupName** (*string*) –

Specifies the name of the DB security group.

- * **DBSecurityGroupDescription** (*string*) –

Provides the description of the DB security group.

* **VpcId** (*string*) –

Provides the VpcId of the DB security group.

* **EC2SecurityGroups** (*list*) –

Contains a list of EC2SecurityGroup elements.

· (*dict*) –

This data type is used as a response element in the following actions:

- AuthorizeDBSecurityGroupIngress
- DescribeDBSecurityGroups
- RevokeDBSecurityGroupIngress
- **Status** (*string*) –

Provides the status of the EC2 security group. Status can be “authorizing”, “authorized”, “revoking”, and “revoked”.

· **EC2SecurityGroupName** (*string*) –

Specifies the name of the EC2 security group.

· **EC2SecurityGroupId** (*string*) –

Specifies the id of the EC2 security group.

· **EC2SecurityGroupOwnerId** (*string*) –

Specifies the AWS ID of the owner of the EC2 security group specified in the EC2SecurityGroupName field.

* **IPRanges** (*list*) –

Contains a list of IPRange elements.

· (*dict*) –

This data type is used as a response element in the DescribeDBSecurityGroups action.

· **Status** (*string*) –

Specifies the status of the IP range. Status can be “authorizing”, “authorized”, “revoking”, and “revoked”.

· **CIDRIP** (*string*) –

Specifies the IP range.

Paginators

The available paginators are:

- `RDS.Paginator.DescribeDBEngineVersions`
- `RDS.Paginator.DescribeDBInstances`
- `RDS.Paginator.DescribeDBLogFiles`
- `RDS.Paginator.DescribeDBParameterGroups`
- `RDS.Paginator.DescribeDBParameters`
- `RDS.Paginator.DescribeDBSecurityGroups`
- `RDS.Paginator.DescribeDBSnapshots`
- `RDS.Paginator.DescribeDBSubnetGroups`

- `RDS.Paginator.DescribeEngineDefaultParameters`
- `RDS.Paginator.DescribeEventSubscriptions`
- `RDS.Paginator.DescribeEvents`
- `RDS.Paginator.DescribeOptionGroupOptions`
- `RDS.Paginator.DescribeOptionGroups`
- `RDS.Paginator.DescribeOrderableDBInstanceOptions`
- `RDS.Paginator.DescribeReservedDBInstances`
- `RDS.Paginator.DescribeReservedDBInstancesOfferings`
- `RDS.Paginator.DownloadDBLogFilePortion`

class `RDS.Paginator.DescribeDBEngineVersions`

```
paginator = client.get_paginator('describe_db_engine_versions')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `RDS.Client.describe_db_engine_versions()`.

Request Syntax

```
response_iterator = paginator.paginate(
    Engine='string',
    EngineVersion='string',
    DBParameterGroupFamily='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    DefaultOnly=True|False,
    ListSupportedCharacterSets=True|False,
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **Engine** (*string*) – The database engine to return.
- **EngineVersion** (*string*) – The database engine version to return.

Example: 5.1.49

- **DBParameterGroupFamily** (*string*) – The name of a specific DB parameter group family to return details for.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **Filters** (*list*) – Not currently supported.
 - (*dict*) –
 - * **Name** (*string*) – [REQUIRED]

This parameter is not currently supported.
 - * **Values** (*list*) – [REQUIRED]

This parameter is not currently supported.
- **DefaultOnly** (*boolean*) – Indicates that only the default version of the specified engine or engine and major version combination is returned.
- **ListSupportedCharacterSets** (*boolean*) – If this parameter is specified, and if the requested engine supports the CharacterSetName parameter for CreateDBInstance, the response includes a list of supported character sets for each engine version.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a NextToken will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –

The size of each page.
 - **StartingToken** (*string*) –

A token to specify where to start paginating. This is the NextToken from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'DBEngineVersions': [
    {
      'Engine': 'string',
      'EngineVersion': 'string',
      'DBParameterGroupFamily': 'string',
      'DBEngineDescription': 'string',
      'DBEngineVersionDescription': 'string',
      'DefaultCharacterSet': {
        'CharacterSetName': 'string',
        'CharacterSetDescription': 'string'
      },
      'SupportedCharacterSets': [
        {
          'CharacterSetName': 'string',
          'CharacterSetDescription': 'string'
        }
      ]
    }
  ],
  'NextToken': 'string'
}
```

Response Structure

- *(dict)* –

Contains the result of a successful invocation of the DescribeDBEngineVersions action.

- **DBEngineVersions** (*list*) –

A list of DBEngineVersion elements.

- * *(dict)* –

This data type is used as a response element in the action DescribeDBEngineVersions .

- **Engine** (*string*) –

The name of the database engine.

- **EngineVersion** (*string*) –

The version number of the database engine.

- **DBParameterGroupFamily** (*string*) –

The name of the DB parameter group family for the database engine.

- **DBEngineDescription** (*string*) –

The description of the database engine.

- **DBEngineVersionDescription** (*string*) –

The description of the database engine version.

- **DefaultCharacterSet** (*dict*) –

The default character set for new instances of this engine version, if the CharacterSetName parameter of the CreateDBInstance API is not specified.

- **CharacterSetName** (*string*) –

The name of the character set.

- **CharacterSetDescription** (*string*) –

The description of the character set.

- **SupportedCharacterSets** (*list*) –

A list of the character sets supported by this engine for the CharacterSetName parameter of the CreateDBInstance API.

- *(dict)* –

This data type is used as a response element in the action DescribeDBEngineVersions .

- **CharacterSetName** (*string*) –

The name of the character set.

- **CharacterSetDescription** (*string*) –

The description of the character set.

- **NextToken** (*string*) –

A token to resume pagination.

class RDS.Paginator.**DescribeDBInstances**

```
paginator = client.get_paginator('describe_db_instances')
```

paginate (**kwargs)

Creates an iterator that will paginate through responses from `RDS.Client.describe_db_instances()`.

Request Syntax

```
response_iterator = paginator.paginate(
    DBInstanceIdentifier='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **DBInstanceIdentifier** (*string*) – The user-supplied instance identifier. If this parameter is specified, information from only the specific DB instance is returned. This parameter isn't case-sensitive.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **Filters** (*list*) – This parameter is not currently supported.

– (*dict*) –

* **Name** (*string*) – **[REQUIRED]**

This parameter is not currently supported.

* **Values** (*list*) – **[REQUIRED]**

This parameter is not currently supported.

· (*string*) –

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

– **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

– **PageSize** (*integer*) –

The size of each page.

– **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns**Response Syntax**

```
{
  'DBInstances': [
    {
      'DBInstanceIdentifier': 'string',
      'DBInstanceClass': 'string',
      'Engine': 'string',
      'DBInstanceStatus': 'string',
      'MasterUsername': 'string',
      'DBName': 'string',
      'Endpoint': {
        'Address': 'string',
        'Port': 123
      },
      'AllocatedStorage': 123,
      'InstanceCreateTime': datetime(2015, 1, 1),
      'PreferredBackupWindow': 'string',
      'BackupRetentionPeriod': 123,
      'DBSecurityGroups': [
        {
          'DBSecurityGroupName': 'string',
          'Status': 'string'
        },
      ],
      'VpcSecurityGroups': [
        {
          'VpcSecurityGroupId': 'string',
          'Status': 'string'
        },
      ],
      'DBParameterGroups': [
        {
          'DBParameterGroupName': 'string',
          'ParameterApplyStatus': 'string'
        },
      ],
      'AvailabilityZone': 'string',
      'DBSubnetGroup': {
        'DBSubnetGroupName': 'string',
        'DBSubnetGroupDescription': 'string',
        'VpcId': 'string',
        'SubnetGroupStatus': 'string',
        'Subnets': [
          {
            'SubnetIdentifier': 'string',
            'SubnetAvailabilityZone': {
              'Name': 'string'
            },
            'SubnetStatus': 'string'
          },
        ],
      },
    },
  ],
  'PreferredMaintenanceWindow': 'string',
  'PendingModifiedValues': {
    'DBInstanceClass': 'string',
    'AllocatedStorage': 123,
    'MasterUserPassword': 'string',
    'Port': 123,
    'BackupRetentionPeriod': 123,
```

```

        'MultiAZ': True|False,
        'EngineVersion': 'string',
        'Iops': 123,
        'DBInstanceIdentifier': 'string',
        'StorageType': 'string',
        'CACertificateIdentifier': 'string'
    },
    'LatestRestorableTime': datetime(2015, 1, 1),
    'MultiAZ': True|False,
    'EngineVersion': 'string',
    'AutoMinorVersionUpgrade': True|False,
    'ReadReplicaSourceDBInstanceIdentifier': 'string',
    'ReadReplicaDBInstanceIdentifiers': [
        'string',
    ],
    'LicenseModel': 'string',
    'Iops': 123,
    'OptionGroupMemberships': [
        {
            'OptionGroupName': 'string',
            'Status': 'string'
        },
    ],
    'CharacterSetName': 'string',
    'SecondaryAvailabilityZone': 'string',
    'PubliclyAccessible': True|False,
    'StatusInfos': [
        {
            'StatusType': 'string',
            'Normal': True|False,
            'Status': 'string',
            'Message': 'string'
        },
    ],
    'StorageType': 'string',
    'TdeCredentialArn': 'string',
    'DbInstancePort': 123,
    'DBClusterIdentifier': 'string',
    'StorageEncrypted': True|False,
    'KmsKeyId': 'string',
    'DbiResourceId': 'string',
    'CACertificateIdentifier': 'string',
    'DomainMemberships': [
        {
            'Domain': 'string',
            'Status': 'string',
            'Connectivity': 'string'
        },
    ],
    'CopyTagsToSnapshot': True|False
},
'NextToken': 'string'
}

```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the DescribeDBInstances action.

– **DBInstances** (*list*) –

A list of DBInstance instances.

* (*dict*) –

Contains the result of a successful invocation of the following actions:

- **CreateDBInstance**
- **DeleteDBInstance**
- **ModifyDBInstance**

This data type is used as a response element in the DescribeDBInstances action.

· **DBInstanceIdentifier** (*string*) –

Contains a user-supplied database identifier. This identifier is the unique key that identifies a DB instance.

· **DBInstanceClass** (*string*) –

Contains the name of the compute and memory capacity class of the DB instance.

· **Engine** (*string*) –

Provides the name of the database engine to be used for this DB instance.

· **DBInstanceStatus** (*string*) –

Specifies the current state of this database.

· **MasterUsername** (*string*) –

Contains the master username for the DB instance.

· **DBName** (*string*) –

The meaning of this parameter differs according to the database engine you use. For example, this value returns either MySQL or PostgreSQL information when returning values from CreateDBInstanceReadReplica since Read Replicas are only supported for MySQL and PostgreSQL.

MySQL, SQL Server, PostgreSQL

Contains the name of the initial database of this instance that was provided at create time, if one was specified when the DB instance was created. This same name is returned for the life of the DB instance.

Type: String

Oracle

Contains the Oracle System ID (SID) of the created DB instance. Not shown when the returned parameters do not apply to an Oracle DB instance.

· **Endpoint** (*dict*) –

Specifies the connection endpoint.

· **Address** (*string*) –

Specifies the DNS address of the DB instance.

· **Port** (*integer*) –

Specifies the port that the database engine is listening on.

· **AllocatedStorage** (*integer*) –

Specifies the allocated storage size specified in gigabytes.

- **InstanceCreateTime** (*datetime*) –
Provides the date and time the DB instance was created.
- **PreferredBackupWindow** (*string*) –
Specifies the daily time range during which automated backups are created if automated backups are enabled, as determined by the `BackupRetentionPeriod`.
- **BackupRetentionPeriod** (*integer*) –
Specifies the number of days for which automatic DB snapshots are retained.
- **DBSecurityGroups** (*list*) –
Provides List of DB security group elements containing only `DBSecurityGroup.Name` and `DBSecurityGroup.Status` subelements.
- (*dict*) –
This data type is used as a response element in the following actions:
 - `ModifyDBInstance`
 - `RebootDBInstance`
 - `RestoreDBInstanceFromDBSnapshot`
 - `RestoreDBInstanceToPointInTime`
- **DBSecurityGroupName** (*string*) –
The name of the DB security group.
- **Status** (*string*) –
The status of the DB security group.
- **VpcSecurityGroups** (*list*) –
Provides List of VPC security group elements that the DB instance belongs to.
- (*dict*) –
This data type is used as a response element for queries on VPC security group membership.
- **VpcSecurityGroupId** (*string*) –
The name of the VPC security group.
- **Status** (*string*) –
The status of the VPC security group.
- **DBParameterGroups** (*list*) –
Provides the list of DB parameter groups applied to this DB instance.
- (*dict*) –
The status of the DB parameter group.
This data type is used as a response element in the following actions:
 - `CreateDBInstance`
 - `CreateDBInstanceReadReplica`
 - `DeleteDBInstance`
 - `ModifyDBInstance`
 - `RebootDBInstance`
 - `RestoreDBInstanceFromDBSnapshot`

- **DBParameterGroupName** (*string*) –
The name of the DP parameter group.
- **ParameterApplyStatus** (*string*) –
The status of parameter updates.
- **AvailabilityZone** (*string*) –
Specifies the name of the Availability Zone the DB instance is located in.
- **DBSubnetGroup** (*dict*) –
Specifies information on the subnet group associated with the DB instance, including the name, description, and subnets in the subnet group.
- **DBSubnetGroupName** (*string*) –
Specifies the name of the DB subnet group.
- **DBSubnetGroupDescription** (*string*) –
Provides the description of the DB subnet group.
- **VpcId** (*string*) –
Provides the VpcId of the DB subnet group.
- **SubnetGroupStatus** (*string*) –
Provides the status of the DB subnet group.
- **Subnets** (*list*) –
Contains a list of Subnet elements.
- (*dict*) –
This data type is used as a response element in the DescribeDBSubnetGroups action.
- **SubnetIdentifier** (*string*) –
Specifies the identifier of the subnet.
- **SubnetAvailabilityZone** (*dict*) –
Contains Availability Zone information.
This data type is used as an element in the following data type:
 - OrderableDBInstanceOption
- **Name** (*string*) –
The name of the availability zone.
- **SubnetStatus** (*string*) –
Specifies the status of the subnet.
- **PreferredMaintenanceWindow** (*string*) –
Specifies the weekly time range during which system maintenance can occur, in Universal Coordinated Time (UTC).
- **PendingModifiedValues** (*dict*) –
Specifies that changes to the DB instance are pending. This element is only included when changes are pending. Specific changes are identified by subelements.
- **DBInstanceClass** (*string*) –
Contains the new DBInstanceClass for the DB instance that will be applied or is in progress.

- **AllocatedStorage** (*integer*) –
Contains the new `AllocatedStorage` size for the DB instance that will be applied or is in progress.
- **MasterUserPassword** (*string*) –
Contains the pending or in-progress change of the master credentials for the DB instance.
- **Port** (*integer*) –
Specifies the pending port for the DB instance.
- **BackupRetentionPeriod** (*integer*) –
Specifies the pending number of days for which automated backups are retained.
- **MultiAZ** (*boolean*) –
Indicates that the Single-AZ DB instance is to change to a Multi-AZ deployment.
- **EngineVersion** (*string*) –
Indicates the database engine version.
- **Iops** (*integer*) –
Specifies the new Provisioned IOPS value for the DB instance that will be applied or is being applied.
- **DBInstanceIdentifier** (*string*) –
Contains the new `DBInstanceIdentifier` for the DB instance that will be applied or is in progress.
- **StorageType** (*string*) –
Specifies the storage type to be associated with the DB instance.
- **CACertificateIdentifier** (*string*) –
Specifies the identifier of the CA certificate for the DB instance.
- **LatestRestorableTime** (*datetime*) –
Specifies the latest time to which a database can be restored with point-in-time restore.
- **MultiAZ** (*boolean*) –
Specifies if the DB instance is a Multi-AZ deployment.
- **EngineVersion** (*string*) –
Indicates the database engine version.
- **AutoMinorVersionUpgrade** (*boolean*) –
Indicates that minor version patches are applied automatically.
- **ReadReplicaSourceDBInstanceIdentifier** (*string*) –
Contains the identifier of the source DB instance if this DB instance is a Read Replica.
- **ReadReplicaDBInstanceIdentifiers** (*list*) –
Contains one or more identifiers of the Read Replicas associated with this DB instance.
- (*string*) –

- **LicenseModel** (*string*) –
License model information for this DB instance.
- **Iops** (*integer*) –
Specifies the Provisioned IOPS (I/O operations per second) value.
- **OptionGroupMemberships** (*list*) –
Provides the list of option group memberships for this DB instance.
- (*dict*) –
Provides information on the option groups the DB instance is a member of.
- **OptionGroupName** (*string*) –
The name of the option group that the instance belongs to.
- **Status** (*string*) –
The status of the DB instance's option group membership (e.g. in-sync, pending, pending-maintenance, applying).
- **CharacterSetName** (*string*) –
If present, specifies the name of the character set that this instance is associated with.
- **SecondaryAvailabilityZone** (*string*) –
If present, specifies the name of the secondary Availability Zone for a DB instance with multi-AZ support.
- **PubliclyAccessible** (*boolean*) –
Specifies the accessibility options for the DB instance. A value of true specifies an Internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. A value of false specifies an internal instance with a DNS name that resolves to a private IP address.

Default: The default behavior varies depending on whether a VPC has been requested or not. The following list shows the default behavior in each case.
 - **Default VPC:** true
 - **VPC:** falseIf no DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be publicly accessible. If a specific DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be private.
- **StatusInfos** (*list*) –
The status of a Read Replica. If the instance is not a Read Replica, this will be blank.
- (*dict*) –
Provides a list of status information for a DB instance.
- **StatusType** (*string*) –
This value is currently “read replication.”
- **Normal** (*boolean*) –

Boolean value that is true if the instance is operating normally, or false if the instance is in an error state.

- **Status** (*string*) –

Status of the DB instance. For a `StatusType` of read replica, the values can be replicating, error, stopped, or terminated.

- **Message** (*string*) –

Details of the error if there is an error for the instance. If the instance is not in an error state, this value is blank.

- **StorageType** (*string*) –

Specifies the storage type associated with DB instance.

- **TdeCredentialArn** (*string*) –

The ARN from the Key Store with which the instance is associated for TDE encryption.

- **DbInstancePort** (*integer*) –

Specifies the port that the DB instance listens on. If the DB instance is part of a DB cluster, this can be a different port than the DB cluster port.

- **DBClusterIdentifier** (*string*) –

If the DB instance is a member of a DB cluster, contains the name of the DB cluster that the DB instance is a member of.

- **StorageEncrypted** (*boolean*) –

Specifies whether the DB instance is encrypted.

- **KmsKeyId** (*string*) –

If `StorageEncrypted` is true, the KMS key identifier for the encrypted DB instance.

- **DbiResourceId** (*string*) –

If `StorageEncrypted` is true, the region-unique, immutable identifier for the encrypted DB instance. This identifier is found in AWS CloudTrail log entries whenever the KMS key for the DB instance is accessed.

- **CACertificateIdentifier** (*string*) –

The identifier of the CA certificate for this DB instance.

- **DomainMemberships** (*list*) –

The Active Directory Domain membership records associated with the DB instance.

- (*dict*) –

An Active Directory Domain membership record associated with the DB instance.

- **Domain** (*string*) –

The identifier of the Active Directory Domain.

- **Status** (*string*) –

The status of the DB instance's Active Directory Domain membership (e.g. joined, pending-join, failed etc).

- **Connectivity** (*string*) –

The observed connectivity of the Active Directory Domain.

- **CopyTagsToSnapshot** (*boolean*) –

This property is not currently implemented.

- **NextToken** (*string*) –

A token to resume pagination.

class `RDS.Paginator.DescribeDBLogFiles`

```
paginator = client.get_paginator('describe_db_log_files')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `RDS.Client.describe_db_log_files()`.

Request Syntax

```
response_iterator = paginator.paginate(
    DBInstanceIdentifier='string',
    FilenameContains='string',
    FileLastWritten=123,
    FileSize=123,
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **DBInstanceIdentifier** (*string*) – **[REQUIRED]**

The customer-assigned name of the DB instance that contains the log files you want to list.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **FilenameContains** (*string*) – Filters the available log files for log file names that contain the specified string.

- **FileLastWritten** (*integer*) – Filters the available log files for files written since the specified date, in POSIX timestamp format with milliseconds.

- **FileSize** (*integer*) – Filters the available log files for files larger than the specified size.

- **Filters** (*list*) – This parameter is not currently supported.

- (*dict*) –

- * **Name** (*string*) – **[REQUIRED]**

This parameter is not currently supported.

* **Values** (*list*) – [REQUIRED]

This parameter is not currently supported.

· (*string*) –

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'DescribeDBLogFiles': [
    {
      'LogFileName': 'string',
      'LastWritten': 123,
      'Size': 123
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

The response from a call to `DescribeDBLogFiles`.

- **DescribeDBLogFiles** (*list*) –

The DB log files returned.

- * (*dict*) –

This data type is used as a response element to `DescribeDBLogFiles`.

- **LogFileName** (*string*) –

The name of the log file for the specified DB instance.

- **LastWritten** (*integer*) –

A POSIX timestamp when the last log entry was written.

- **Size** (*integer*) –

The size, in bytes, of the log file for the specified DB instance.

- **NextToken** (*string*) –

A token to resume pagination.

`class RDS.Paginator.DescribeDBParameterGroups`

```
paginator = client.get_paginator('describe_db_parameter_groups')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `RDS.Client.describe_db_parameter_groups()`.

Request Syntax

```
response_iterator = paginator.paginate(
    DBParameterGroupName='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **DBParameterGroupName** (*string*) – The name of a specific DB parameter group to return details for.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **Filters** (*list*) – This parameter is not currently supported.

– (*dict*) –

* **Name** (*string*) – **[REQUIRED]**

This parameter is not currently supported.

* **Values** (*list*) – **[REQUIRED]**

This parameter is not currently supported.

· (*string*) –

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

– **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

– **PageSize** (*integer*) –

The size of each page.

– **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'DBParameterGroups': [
    {
      'DBParameterGroupName': 'string',
      'DBParameterGroupFamily': 'string',
      'Description': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the DescribeDBParameterGroups action.

- **DBParameterGroups** (*list*) –

A list of DBParameterGroup instances.

- * (*dict*) –

Contains the result of a successful invocation of the CreateDBParameterGroup action.

This data type is used as a request parameter in the DeleteDBParameterGroup action, and as a response element in the DescribeDBParameterGroups action.

- **DBParameterGroupName** (*string*) –

Provides the name of the DB parameter group.

- **DBParameterGroupFamily** (*string*) –

Provides the name of the DB parameter group family that this DB parameter group is compatible with.

- **Description** (*string*) –

Provides the customer-specified description for this DB parameter group.

- **NextToken** (*string*) –

A token to resume pagination.

class RDS.Paginator.**DescribeDBParameters**

```
paginator = client.get_paginator('describe_db_parameters')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `RDS.Client.describe_db_parameters()`.

Request Syntax

```
response_iterator = paginator.paginate(
    DBParameterGroupName='string',
    Source='string',
```

```
Filters=[
    {
        'Name': 'string',
        'Values': [
            'string',
        ]
    },
],
PaginationConfig={
    'MaxItems': 123,
    'PageSize': 123,
    'StartingToken': 'string'
}
```

Parameters

- **DBParameterGroupName** (*string*) – **[REQUIRED]**

The name of a specific DB parameter group to return details for.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **Source** (*string*) – The parameter types to return.

Default: All parameter types returned

Valid Values: `user` | `system` | `engine-default`

- **Filters** (*list*) – This parameter is not currently supported.

– (*dict*) –

- * **Name** (*string*) – **[REQUIRED]**

This parameter is not currently supported.

- * **Values** (*list*) – **[REQUIRED]**

This parameter is not currently supported.

· (*string*) –

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
    'Parameters': [
        {
```



```

        'ParameterName': 'string',
        'ParameterValue': 'string',
        'Description': 'string',
        'Source': 'string',
        'ApplyType': 'string',
        'DataType': 'string',
        'AllowedValues': 'string',
        'IsModifiable': True|False,
        'MinimumEngineVersion': 'string',
        'ApplyMethod': 'immediate'|'pending-reboot'
    },
],
'NextToken': 'string'
}

```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the DescribeDBParameters action.

- **Parameters** (*list*) –

A list of Parameter values.

- * (*dict*) –

This data type is used as a request parameter in the ModifyDBParameterGroup and ResetDBParameterGroup actions.

This data type is used as a response element in the DescribeEngineDefaultParameters and DescribeDBParameters actions.

- **ParameterName** (*string*) –

Specifies the name of the parameter.

- **ParameterValue** (*string*) –

Specifies the value of the parameter.

- **Description** (*string*) –

Provides a description of the parameter.

- **Source** (*string*) –

Indicates the source of the parameter value.

- **ApplyType** (*string*) –

Specifies the engine specific parameters type.

- **DataType** (*string*) –

Specifies the valid data type for the parameter.

- **AllowedValues** (*string*) –

Specifies the valid range of values for the parameter.

- **IsModifiable** (*boolean*) –

Indicates whether (`true`) or not (`false`) the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.

- **MinimumEngineVersion** (*string*) –

The earliest engine version to which the parameter can apply.

- **ApplyMethod** (*string*) –

Indicates when to apply parameter updates.

- **NextToken** (*string*) –

A token to resume pagination.

class `RDS.Paginator.DescribeDBSecurityGroups`

```
paginator = client.get_paginator('describe_db_security_groups')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `RDS.Client.describe_db_security_groups()`.

Request Syntax

```
response_iterator = paginator.paginate(
    DBSecurityGroupName='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **DBSecurityGroupName** (*string*) – The name of the DB security group to return details for.
- **Filters** (*list*) – This parameter is not currently supported.
 - (*dict*) –
 - * **Name** (*string*) – **[REQUIRED]**
This parameter is not currently supported.
 - * **Values** (*list*) – **[REQUIRED]**
This parameter is not currently supported.
 - (*string*) –
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'DBSecurityGroups': [
    {
      'OwnerId': 'string',
      'DBSecurityGroupName': 'string',
      'DBSecurityGroupDescription': 'string',
      'VpcId': 'string',
      'EC2SecurityGroups': [
        {
          'Status': 'string',
          'EC2SecurityGroupName': 'string',
          'EC2SecurityGroupId': 'string',
          'EC2SecurityGroupOwnerId': 'string'
        },
      ],
      'IPRanges': [
        {
          'Status': 'string',
          'CIDRIP': 'string'
        },
      ],
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the `DescribeDBSecurityGroups` action.

- **DBSecurityGroups** (*list*) –

A list of `DBSecurityGroup` instances.

- * (*dict*) –

Contains the result of a successful invocation of the following actions:

- `DescribeDBSecurityGroups`
- `AuthorizeDBSecurityGroupIngress`
- `CreateDBSecurityGroup`
- `RevokeDBSecurityGroupIngress`

This data type is used as a response element in the `DescribeDBSecurityGroups` action.

- **OwnerId** (*string*) –

Provides the AWS ID of the owner of a specific DB security group.

- **DBSecurityGroupName** (*string*) –

Specifies the name of the DB security group.

- **DBSecurityGroupDescription** (*string*) –

Provides the description of the DB security group.

- **VpcId** (*string*) –

Provides the VpcId of the DB security group.

- **EC2SecurityGroups** (*list*) –

Contains a list of EC2SecurityGroup elements.

- (*dict*) –

This data type is used as a response element in the following actions:

- AuthorizeDBSecurityGroupIngress
- DescribeDBSecurityGroups
- RevokeDBSecurityGroupIngress
- **Status** (*string*) –

Provides the status of the EC2 security group. Status can be “authorizing”, “authorized”, “revoking”, and “revoked”.

- **EC2SecurityGroupName** (*string*) –

Specifies the name of the EC2 security group.

- **EC2SecurityGroupId** (*string*) –

Specifies the id of the EC2 security group.

- **EC2SecurityGroupOwnerId** (*string*) –

Specifies the AWS ID of the owner of the EC2 security group specified in the EC2SecurityGroupName field.

- **IPRanges** (*list*) –

Contains a list of IPRange elements.

- (*dict*) –

This data type is used as a response element in the DescribeDBSecurityGroups action.

- **Status** (*string*) –

Specifies the status of the IP range. Status can be “authorizing”, “authorized”, “revoking”, and “revoked”.

- **CIDRIP** (*string*) –

Specifies the IP range.

- **NextToken** (*string*) –

A token to resume pagination.

class RDS.Paginator.**DescribeDBSnapshots**

```
paginator = client.get_paginator('describe_db_snapshots')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `RDS.Client.describe_db_snapshots()`.

Request Syntax

```
response_iterator = paginator.paginate(  
    DBInstanceIdentifier='string',  
    DBSnapshotIdentifier='string',  
    SnapshotType='string',
```

```

Filters=[
    {
        'Name': 'string',
        'Values': [
            'string',
        ]
    },
],
PaginationConfig={
    'MaxItems': 123,
    'PageSize': 123,
    'StartingToken': 'string'
}
)

```

Parameters

- **DBInstanceIdentifier** (*string*) – A DB instance identifier to retrieve the list of DB snapshots for. This parameter cannot be used in conjunction with `DBSnapshotIdentifier`. This parameter is not case-sensitive.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **DBSnapshotIdentifier** (*string*) – A specific DB snapshot identifier to describe. This parameter cannot be used in conjunction with `DBInstanceIdentifier`. This value is stored as a lowercase string.

Constraints:

- Must be 1 to 255 alphanumeric characters.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.
- If this is the identifier of an automated snapshot, the `SnapshotType` parameter must also be specified.

- **SnapshotType** (*string*) – The type of snapshots that will be returned. Values can be “automated” or “manual.” If not specified, the returned results will include all snapshots types.

- **Filters** (*list*) – This parameter is not currently supported.

– (dict) –

- * **Name** (*string*) – [REQUIRED]

This parameter is not currently supported.

- * **Values** (*list*) – [REQUIRED]

This parameter is not currently supported.

· (*string*) –

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

– **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

– **PageSize** (*integer*) –

The size of each page.

– **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'DBSnapshots': [
    {
      'DBSnapshotIdentifier': 'string',
      'DBInstanceIdentifier': 'string',
      'SnapshotCreateTime': datetime(2015, 1, 1),
      'Engine': 'string',
      'AllocatedStorage': 123,
      'Status': 'string',
      'Port': 123,
      'AvailabilityZone': 'string',
      'VpcId': 'string',
      'InstanceCreateTime': datetime(2015, 1, 1),
      'MasterUsername': 'string',
      'EngineVersion': 'string',
      'LicenseModel': 'string',
      'SnapshotType': 'string',
      'Iops': 123,
      'OptionGroupName': 'string',
      'PercentProgress': 123,
      'SourceRegion': 'string',
      'SourceDBSnapshotIdentifier': 'string',
      'StorageType': 'string',
      'TdeCredentialArn': 'string',
      'Encrypted': True|False,
      'KmsKeyId': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the `DescribeDBSnapshots` action.

- **DBSnapshots** (*list*) –

A list of `DBSnapshot` instances.

- * (*dict*) –

Contains the result of a successful invocation of the following actions:

- `CreateDBSnapshot`
- `DeleteDBSnapshot`

This data type is used as a response element in the `DescribeDBSnapshots` action.

- **DBSnapshotIdentifier** (*string*) –

Specifies the identifier for the DB snapshot.

- **DBInstanceIdentifier** (*string*) –

Specifies the DB instance identifier of the DB instance this DB snapshot was created from.

- **SnapshotCreateTime** (*datetime*) –

Provides the time when the snapshot was taken, in Universal Coordinated Time (UTC).

- **Engine** (*string*) –

Specifies the name of the database engine.

- **AllocatedStorage** (*integer*) –

Specifies the allocated storage size in gigabytes (GB).

- **Status** (*string*) –

Specifies the status of this DB snapshot.

- **Port** (*integer*) –

Specifies the port that the database engine was listening on at the time of the snapshot.

- **AvailabilityZone** (*string*) –

Specifies the name of the Availability Zone the DB instance was located in at the time of the DB snapshot.

- **VpcId** (*string*) –

Provides the VPC ID associated with the DB snapshot.

- **InstanceCreateTime** (*datetime*) –

Specifies the time when the snapshot was taken, in Universal Coordinated Time (UTC).

- **MasterUsername** (*string*) –

Provides the master username for the DB snapshot.

- **EngineVersion** (*string*) –

Specifies the version of the database engine.

- **LicenseModel** (*string*) –

License model information for the restored DB instance.

- **SnapshotType** (*string*) –

Provides the type of the DB snapshot.

- **Iops** (*integer*) –

Specifies the Provisioned IOPS (I/O operations per second) value of the DB instance at the time of the snapshot.

- **OptionGroupName** (*string*) –

Provides the option group name for the DB snapshot.

- **PercentProgress** (*integer*) –

The percentage of the estimated data that has been transferred.

- **SourceRegion** (*string*) –

The region that the DB snapshot was created in or copied from.

- **SourceDBSnapshotIdentifier** (*string*) –

The DB snapshot Arn that the DB snapshot was copied from. It only has value in case of cross customer or cross region copy.

- **StorageType** (*string*) –
Specifies the storage type associated with DB Snapshot.
- **TdeCredentialArn** (*string*) –
The ARN from the Key Store with which to associate the instance for TDE encryption.
- **Encrypted** (*boolean*) –
Specifies whether the DB snapshot is encrypted.
- **KmsKeyId** (*string*) –
If Encrypted is true, the KMS key identifier for the encrypted DB snapshot.
- **NextToken** (*string*) –
A token to resume pagination.

class RDS.Paginator.**DescribeDBSubnetGroups**

```
paginator = client.get_paginator('describe_db_subnet_groups')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `RDS.Client.describe_db_subnet_groups()`.

Request Syntax

```
response_iterator = paginator.paginate(
    DBSubnetGroupName='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **DBSubnetGroupName** (*string*) – The name of the DB subnet group to return details for.
- **Filters** (*list*) – This parameter is not currently supported.
 - (*dict*) –
 - * **Name** (*string*) – **[REQUIRED]**

This parameter is not currently supported.
 - * **Values** (*list*) – **[REQUIRED]**

This parameter is not currently supported.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'DBSubnetGroups': [
    {
      'DBSubnetGroupName': 'string',
      'DBSubnetGroupDescription': 'string',
      'VpcId': 'string',
      'SubnetGroupStatus': 'string',
      'Subnets': [
        {
          'SubnetIdentifier': 'string',
          'SubnetAvailabilityZone': {
            'Name': 'string'
          },
          'SubnetStatus': 'string'
        },
        ...
      ],
      ...
    },
    ...
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the `DescribeDBSubnetGroups` action.

- **DBSubnetGroups** (*list*) –

A list of `DBSubnetGroup` instances.

- * (*dict*) –

Contains the result of a successful invocation of the following actions:

- `CreateDBSubnetGroup`
- `ModifyDBSubnetGroup`
- `DescribeDBSubnetGroups`
- `DeleteDBSubnetGroup`

This data type is used as a response element in the `DescribeDBSubnetGroups` action.

- **DBSubnetGroupName** (*string*) –

Specifies the name of the DB subnet group.

- **DBSubnetGroupDescription** (*string*) –
Provides the description of the DB subnet group.
- **VpcId** (*string*) –
Provides the VpcId of the DB subnet group.
- **SubnetGroupStatus** (*string*) –
Provides the status of the DB subnet group.
- **Subnets** (*list*) –
Contains a list of Subnet elements.
- (*dict*) –
This data type is used as a response element in the DescribeDBSubnetGroups action.
- **SubnetIdentifier** (*string*) –
Specifies the identifier of the subnet.
- **SubnetAvailabilityZone** (*dict*) –
Contains Availability Zone information.
This data type is used as an element in the following data type:
 - OrderableDBInstanceOption
 - **Name** (*string*) –
The name of the availability zone.
 - **SubnetStatus** (*string*) –
Specifies the status of the subnet.
- **NextToken** (*string*) –
A token to resume pagination.

class RDS.Paginator.**DescribeEngineDefaultParameters**

```
paginator = client.get_paginator('describe_engine_default_parameters')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `RDS.Client.describe_engine_default_parameters()`.

Request Syntax

```
response_iterator = paginator.paginate(  
    DBParameterGroupFamily='string',  
    Filters=[  
        {  
            'Name': 'string',  
            'Values': [  
                'string',  
            ]  
        },  
    ],  
    PaginationConfig={  
        'MaxItems': 123,  
        'PageSize': 123,  
        'StartingToken': 'string'  
    }  
)
```

Parameters

- **DBParameterGroupFamily** (*string*) – [REQUIRED]
The name of the DB parameter group family.
- **Filters** (*list*) – Not currently supported.
 - (*dict*) –
 - * **Name** (*string*) – [REQUIRED]
This parameter is not currently supported.
 - * **Values** (*list*) – [REQUIRED]
This parameter is not currently supported.
 - (*string*) –
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a NextToken will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the NextToken from a previous response.

Return type dict**Returns****Response Syntax**

```
{
  'EngineDefaults': {
    'DBParameterGroupFamily': 'string',
    'Marker': 'string',
    'Parameters': [
      {
        'ParameterName': 'string',
        'ParameterValue': 'string',
        'Description': 'string',
        'Source': 'string',
        'ApplyType': 'string',
        'DataType': 'string',
        'AllowedValues': 'string',
        'IsModifiable': True|False,
        'MinimumEngineVersion': 'string',
        'ApplyMethod': 'immediate'|'pending-reboot'
      },
    ]
  },
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the DescribeEngineDefaultParameters action.

– **EngineDefaults** (*dict*) –

Contains the result of a successful invocation of the DescribeEngineDefaultParameters action.

* **DBParameterGroupFamily** (*string*) –

Specifies the name of the DB parameter group family that the engine default parameters apply to.

* **Marker** (*string*) –

An optional pagination token provided by a previous EngineDefaults request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

* **Parameters** (*list*) –

Contains a list of engine default parameters.

· (*dict*) –

This data type is used as a request parameter in the ModifyDBParameterGroup and ResetDBParameterGroup actions.

This data type is used as a response element in the DescribeEngineDefaultParameters and DescribeDBParameters actions.

· **ParameterName** (*string*) –

Specifies the name of the parameter.

· **ParameterValue** (*string*) –

Specifies the value of the parameter.

· **Description** (*string*) –

Provides a description of the parameter.

· **Source** (*string*) –

Indicates the source of the parameter value.

· **ApplyType** (*string*) –

Specifies the engine specific parameters type.

· **DataType** (*string*) –

Specifies the valid data type for the parameter.

· **AllowedValues** (*string*) –

Specifies the valid range of values for the parameter.

· **IsModifiable** (*boolean*) –

Indicates whether (`true`) or not (`false`) the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.

· **MinimumEngineVersion** (*string*) –

The earliest engine version to which the parameter can apply.

· **ApplyMethod** (*string*) –

Indicates when to apply parameter updates.

– **NextToken** (*string*) –

A token to resume pagination.

class `RDS.Paginator.DescribeEventSubscriptions`

```
paginator = client.get_paginator('describe_event_subscriptions')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `RDS.Client.describe_event_subscriptions()`.

Request Syntax

```
response_iterator = paginator.paginate(
    SubscriptionName='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **SubscriptionName** (*string*) – The name of the RDS event notification subscription you want to describe.
- **Filters** (*list*) – This parameter is not currently supported.
 - (*dict*) –
 - * **Name** (*string*) – **[REQUIRED]**
This parameter is not currently supported.
 - * **Values** (*list*) – **[REQUIRED]**
This parameter is not currently supported.
 - (*string*) –
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
  'EventSubscriptionsList': [
    {
      'CustomerAwsId': 'string',
      'CustSubscriptionId': 'string',
      'SnsTopicArn': 'string',
      'Status': 'string',
      'SubscriptionCreationTime': 'string',
      'SourceType': 'string',
      'SourceIdsList': [
        'string',
      ],
      'EventCategoriesList': [
        'string',
      ],
      'Enabled': True|False
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Data returned by the **DescribeEventSubscriptions** action.

- **EventSubscriptionsList** (*list*) –

A list of EventSubscriptions data types.

- * (*dict*) –

Contains the results of a successful invocation of the **DescribeEventSubscriptions** action.

- **CustomerAwsId** (*string*) –

The AWS customer account associated with the RDS event notification subscription.

- **CustSubscriptionId** (*string*) –

The RDS event notification subscription Id.

- **SnsTopicArn** (*string*) –

The topic ARN of the RDS event notification subscription.

- **Status** (*string*) –

The status of the RDS event notification subscription.

Constraints:

Can be one of the following: creating | modifying | deleting | active | no-permission | topic-not-exist

The status “no-permission” indicates that RDS no longer has permission to post to the SNS topic. The status “topic-not-exist” indicates that the topic was deleted after the subscription was created.

- **SubscriptionCreationTime** (*string*) –

The time the RDS event notification subscription was created.

- **SourceType** (*string*) –

The source type for the RDS event notification subscription.

- **SourceIdsList** (*list*) –
A list of source IDs for the RDS event notification subscription.
- (*string*) –
- **EventCategoriesList** (*list*) –
A list of event categories for the RDS event notification subscription.
- (*string*) –
- **Enabled** (*boolean*) –
A Boolean value indicating if the subscription is enabled. True indicates the subscription is enabled.
- **NextToken** (*string*) –
A token to resume pagination.

class `RDS.Paginator.DescribeEvents`

```
paginator = client.get_paginator('describe_events')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `RDS.Client.describe_events()`.

Request Syntax

```
response_iterator = paginator.paginate(
    SourceIdentifier='string',
    SourceType='db-instance'|'db-parameter-group'|'db-security-group'|'db-snapshot',
    StartTime=datetime(2015, 1, 1),
    EndTime=datetime(2015, 1, 1),
    Duration=123,
    EventCategories=[
        'string',
    ],
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **SourceIdentifier** (*string*) – The identifier of the event source for which events will be returned. If not specified, then all sources are included in the response.

Constraints:

- If `SourceIdentifier` is supplied, `SourceType` must also be provided.
- If the source type is `DBInstance`, then a `DBInstanceIdentifier` must be supplied.

- If the source type is `DBSecurityGroup`, a `DBSecurityGroupName` must be supplied.
- If the source type is `DBParameterGroup`, a `DBParameterGroupName` must be supplied.
- If the source type is `DBSnapshot`, a `DBSnapshotIdentifier` must be supplied.
- Cannot end with a hyphen or contain two consecutive hyphens.
- **SourceType** (*string*) – The event source to retrieve events for. If no value is specified, all events are returned.
- **StartTime** (*datetime*) – The beginning of the time interval to retrieve events for, specified in ISO 8601 format. For more information about ISO 8601, go to the [ISO8601 Wikipedia page](#).

Example: 2009-07-08T18:00Z

- **EndTime** (*datetime*) – The end of the time interval for which to retrieve events, specified in ISO 8601 format. For more information about ISO 8601, go to the [ISO8601 Wikipedia page](#).

Example: 2009-07-08T18:00Z

- **Duration** (*integer*) – The number of minutes to retrieve events for.

Default: 60

- **EventCategories** (*list*) – A list of event categories that trigger notifications for a event notification subscription.

- (*string*) –

- **Filters** (*list*) – This parameter is not currently supported.

- (*dict*) –

- * **Name** (*string*) – [REQUIRED]

- This parameter is not currently supported.

- * **Values** (*list*) – [REQUIRED]

- This parameter is not currently supported.

- (*string*) –

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

- The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

- The size of each page.

- **StartingToken** (*string*) –

- A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Events': [
    {
      'SourceIdentifier': 'string',
      'SourceType': 'db-instance'|'db-parameter-group'|'db-security-group'|
      'Message': 'string',
```



```

        'EventCategories': [
            'string',
        ],
        'Date': datetime(2015, 1, 1)
    },
],
'NextToken': 'string'
}

```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the DescribeEvents action.

- **Events** (*list*) –

A list of Event instances.

- * (*dict*) –

This data type is used as a response element in the DescribeEvents action.

- **SourceIdentifier** (*string*) –

Provides the identifier for the source of the event.

- **SourceType** (*string*) –

Specifies the source type for this event.

- **Message** (*string*) –

Provides the text of this event.

- **EventCategories** (*list*) –

Specifies the category for the event.

- (*string*) –

- **Date** (*datetime*) –

Specifies the date and time of the event.

- **NextToken** (*string*) –

A token to resume pagination.

class RDS.Paginator.**DescribeOptionGroupOptions**

```
paginator = client.get_paginator('describe_option_group_options')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `RDS.Client.describe_option_group_options()`.

Request Syntax

```

response_iterator = paginator.paginate(
    EngineName='string',
    MajorEngineVersion='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        }
    ]
)

```

```
    },  
],  
PaginationConfig={  
    'MaxItems': 123,  
    'PageSize': 123,  
    'StartingToken': 'string'  
}  
)
```

Parameters

- **EngineName** (*string*) – [REQUIRED]

A required parameter. Options available for the given engine name will be described.

- **MajorEngineVersion** (*string*) – If specified, filters the results to include only options for the specified major engine version.

- **Filters** (*list*) – This parameter is not currently supported.

- (*dict*) –

- * **Name** (*string*) – [REQUIRED]

- This parameter is not currently supported.

- * **Values** (*list*) – [REQUIRED]

- This parameter is not currently supported.

- (*string*) –

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

- The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

- The size of each page.

- **StartingToken** (*string*) –

- A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{  
    'OptionGroupOptions': [  
        {  
            'Name': 'string',  
            'Description': 'string',  
            'EngineName': 'string',  
            'MajorEngineVersion': 'string',  
            'MinimumRequiredMinorEngineVersion': 'string',  
            'PortRequired': True|False,  
            'DefaultPort': 123,  
            'OptionsDependedOn': [  
                'string',  
            ],  
            'Persistent': True|False,  
            'Permanent': True|False,  
        },  
    ],  
}
```

```

'OptionGroupOptionSettings': [
    {
        'SettingName': 'string',
        'SettingDescription': 'string',
        'DefaultValue': 'string',
        'ApplyType': 'string',
        'AllowedValues': 'string',
        'IsModifiable': True|False
    },
],
},
'NextToken': 'string'
}

```

Response Structure

- (*dict*) –

- **OptionGroupOptions** (*list*) –

List of available option group options.

- * (*dict*) –

Available option.

- **Name** (*string*) –

The name of the option.

- **Description** (*string*) –

The description of the option.

- **EngineName** (*string*) –

The name of the engine that this option can be applied to.

- **MajorEngineVersion** (*string*) –

Indicates the major engine version that the option is available for.

- **MinimumRequiredMinorEngineVersion** (*string*) –

The minimum required engine version for the option to be applied.

- **PortRequired** (*boolean*) –

Specifies whether the option requires a port.

- **DefaultPort** (*integer*) –

If the option requires a port, specifies the default port for the option.

- **OptionsDependedOn** (*list*) –

List of all options that are prerequisites for this option.

- (*string*) –

- **Persistent** (*boolean*) –

A persistent option cannot be removed from the option group once the option group is used, but this option can be removed from the db instance while modifying the related data and assigning another option group without this option.

- **Permanent** (*boolean*) –

A permanent option cannot be removed from the option group once the option group is used, and it cannot be removed from

the db instance after assigning an option group with this permanent option.

- **OptionGroupOptionSettings** (*list*) –

Specifies the option settings that are available (and the default value) for each option in an option group.

- (*dict*) –

Option group option settings are used to display settings available for each option with their default values and other information. These values are used with the DescribeOptionGroupOptions action.

- **SettingName** (*string*) –

The name of the option group option.

- **SettingDescription** (*string*) –

The description of the option group option.

- **DefaultValue** (*string*) –

The default value for the option group option.

- **ApplyType** (*string*) –

The DB engine specific parameter type for the option group option.

- **AllowedValues** (*string*) –

Indicates the acceptable values for the option group option.

- **IsModifiable** (*boolean*) –

Boolean value where true indicates that this option group option can be changed from the default value.

- **NextToken** (*string*) –

A token to resume pagination.

class RDS.Paginator.**DescribeOptionGroups**

```
paginator = client.get_paginator('describe_option_groups')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `RDS.Client.describe_option_groups()`.

Request Syntax

```
response_iterator = paginator.paginate(
    OptionGroupName='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    EngineName='string',
    MajorEngineVersion='string',
    PaginationConfig={
```

```

        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)

```

Parameters

- **OptionGroupName** (*string*) – The name of the option group to describe. Cannot be supplied together with EngineName or MajorEngineVersion.
- **Filters** (*list*) – This parameter is not currently supported.
 - (*dict*) –
 - * **Name** (*string*) – **[REQUIRED]**
This parameter is not currently supported.
 - * **Values** (*list*) – **[REQUIRED]**
This parameter is not currently supported.
 - (*string*) –
- **EngineName** (*string*) – Filters the list of option groups to only include groups associated with a specific database engine.
- **MajorEngineVersion** (*string*) – Filters the list of option groups to only include groups associated with a specific database engine version. If specified, then EngineName must also be specified.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a NextToken will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the NextToken from a previous response.

Return type dict

Returns

Response Syntax

```

{
  'OptionGroupsList': [
    {
      'OptionGroupName': 'string',
      'OptionGroupDescription': 'string',
      'EngineName': 'string',
      'MajorEngineVersion': 'string',
      'Options': [
        {
          'OptionName': 'string',
          'OptionDescription': 'string',
          'Persistent': True|False,
          'Permanent': True|False,
          'Port': 123,
          'OptionSettings': [
            {

```

```
        'Name': 'string',
        'Value': 'string',
        'DefaultValue': 'string',
        'Description': 'string',
        'ApplyType': 'string',
        'DataType': 'string',
        'AllowedValues': 'string',
        'IsModifiable': True|False,
        'IsCollection': True|False
    },
],
'DBSecurityGroupMemberships': [
    {
        'DBSecurityGroupName': 'string',
        'Status': 'string'
    },
],
'VpcSecurityGroupMemberships': [
    {
        'VpcSecurityGroupId': 'string',
        'Status': 'string'
    },
],
],
'AllowsVpcAndNonVpcInstanceMemberships': True|False,
'VpcId': 'string'
},
],
'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

List of option groups.

- **OptionGroupsList** (*list*) –

List of option groups.

- * (*dict*) –

- **OptionGroupName** (*string*) –

Specifies the name of the option group.

- **OptionGroupDescription** (*string*) –

Provides a description of the option group.

- **EngineName** (*string*) –

Indicates the name of the engine that this option group can be applied to.

- **MajorEngineVersion** (*string*) –

Indicates the major engine version associated with this option group.

- **Options** (*list*) –

Indicates what options are available in the option group.

- (*dict*) –

Option details.

- **OptionName** (*string*) –
The name of the option.
- **OptionDescription** (*string*) –
The description of the option.
- **Persistent** (*boolean*) –
Indicate if this option is persistent.
- **Permanent** (*boolean*) –
Indicate if this option is permanent.
- **Port** (*integer*) –
If required, the port configured for this option to use.
- **OptionSettings** (*list*) –
The option settings for this option.
- (*dict*) –
Option settings are the actual settings being applied or configured for that option. It is used when you modify an option group or describe option groups. For example, the `NATIVE_NETWORK_ENCRYPTION` option has a setting called `SQLNET.ENCRYPTION_SERVER` that can have several different values.
- **Name** (*string*) –
The name of the option that has settings that you can set.
- **Value** (*string*) –
The current value of the option setting.
- **DefaultValue** (*string*) –
The default value of the option setting.
- **Description** (*string*) –
The description of the option setting.
- **ApplyType** (*string*) –
The DB engine specific parameter type.
- **DataType** (*string*) –
The data type of the option setting.
- **AllowedValues** (*string*) –
The allowed values of the option setting.
- **IsModifiable** (*boolean*) –
A Boolean value that, when true, indicates the option setting can be modified from the default.
- **IsCollection** (*boolean*) –
Indicates if the option setting is part of a collection.
- **DBSecurityGroupMemberships** (*list*) –
If the option requires access to a port, then this DB security group allows access to the port.
- (*dict*) –
This data type is used as a response element in the following actions:
 - `ModifyDBInstance`

- `RebootDBInstance`
 - `RestoreDBInstanceFromDBSnapshot`
 - `RestoreDBInstanceToPointInTime`
 - **`DBSecurityGroupName`** (*string*) –
The name of the DB security group.
 - **`Status`** (*string*) –
The status of the DB security group.
 - **`VpcSecurityGroupMemberships`** (*list*) –
If the option requires access to a port, then this VPC security group allows access to the port.
 - (*dict*) –
This data type is used as a response element for queries on VPC security group membership.
 - **`VpcSecurityGroupId`** (*string*) –
The name of the VPC security group.
 - **`Status`** (*string*) –
The status of the VPC security group.
 - **`AllowsVpcAndNonVpcInstanceMemberships`** (*boolean*) –
Indicates whether this option group can be applied to both VPC and non-VPC instances. The value `true` indicates the option group can be applied to both VPC and non-VPC instances.
 - **`VpcId`** (*string*) –
If **`AllowsVpcAndNonVpcInstanceMemberships`** is `false`, this field is blank. If **`AllowsVpcAndNonVpcInstanceMemberships`** is `true` and this field is blank, then this option group can be applied to both VPC and non-VPC instances. If this field contains a value, then this option group can only be applied to instances that are in the VPC indicated by this field.
- **`NextToken`** (*string*) –
A token to resume pagination.

class `RDS.Paginator.DescribeOrderableDBInstanceOptions`

```
paginator = client.get_paginator('describe_orderable_db_instance_options')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `RDS.Client.describe_orderable_db_instance_options()`.

Request Syntax

```
response_iterator = paginator.paginate(  
    Engine='string',  
    EngineVersion='string',  
    DBInstanceClass='string',  
    LicenseModel='string',  
    Vpc=True|False,  
    Filters=[  
        {
```



```

        'Name': 'string',
        'Values': [
            'string',
        ]
    },
],
PaginationConfig={
    'MaxItems': 123,
    'PageSize': 123,
    'StartingToken': 'string'
}
)

```

Parameters

- **Engine** (*string*) – [REQUIRED]

The name of the engine to retrieve DB instance options for.

- **EngineVersion** (*string*) – The engine version filter value. Specify this parameter to show only the available offerings matching the specified engine version.
- **DBInstanceClass** (*string*) – The DB instance class filter value. Specify this parameter to show only the available offerings matching the specified DB instance class.
- **LicenseModel** (*string*) – The license model filter value. Specify this parameter to show only the available offerings matching the specified license model.
- **Vpc** (*boolean*) – The VPC filter value. Specify this parameter to show only the available VPC or non-VPC offerings.
- **Filters** (*list*) – This parameter is not currently supported.

- (*dict*) –

- * **Name** (*string*) – [REQUIRED]

- This parameter is not currently supported.

- * **Values** (*list*) – [REQUIRED]

- This parameter is not currently supported.

- (*string*) –

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```

{
    'OrderableDBInstanceOptions': [
        {
            'Engine': 'string',

```

```
        'EngineVersion': 'string',
        'DBInstanceClass': 'string',
        'LicenseModel': 'string',
        'AvailabilityZones': [
            {
                'Name': 'string'
            },
        ],
        'MultiAZCapable': True|False,
        'ReadReplicaCapable': True|False,
        'Vpc': True|False,
        'SupportsStorageEncryption': True|False,
        'StorageType': 'string',
        'SupportsIops': True|False
    },
],
'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the DescribeOrderableDBInstanceOptions action.

- **OrderableDBInstanceOptions** (*list*) –

An OrderableDBInstanceOption structure containing information about orderable options for the DB instance.

- * (*dict*) –

Contains a list of available options for a DB instance

This data type is used as a response element in the DescribeOrderableDBInstanceOptions action.

- **Engine** (*string*) –

The engine type of the orderable DB instance.

- **EngineVersion** (*string*) –

The engine version of the orderable DB instance.

- **DBInstanceClass** (*string*) –

The DB instance class for the orderable DB instance.

- **LicenseModel** (*string*) –

The license model for the orderable DB instance.

- **AvailabilityZones** (*list*) –

A list of Availability Zones for the orderable DB instance.

- (*dict*) –

Contains Availability Zone information.

This data type is used as an element in the following data type:

- OrderableDBInstanceOption

- **Name** (*string*) –

The name of the availability zone.

- **MultiAZCapable** (*boolean*) –

Indicates whether this orderable DB instance is multi-AZ capable.

- **ReadReplicaCapable** (*boolean*) –
Indicates whether this orderable DB instance can have a Read Replica.
- **Vpc** (*boolean*) –
Indicates whether this is a VPC orderable DB instance.
- **SupportsStorageEncryption** (*boolean*) –
Indicates whether this orderable DB instance supports encrypted storage.
- **StorageType** (*string*) –
Indicates the storage type for this orderable DB instance.
- **SupportsIops** (*boolean*) –
Indicates whether this orderable DB instance supports provisioned IOPS.

– **NextToken** (*string*) –

A token to resume pagination.

class RDS.Paginator.**DescribeReservedDBInstances**

```
paginator = client.get_paginator('describe_reserved_db_instances')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `RDS.Client.describe_reserved_db_instances()`.

Request Syntax

```
response_iterator = paginator.paginate(
    ReservedDBInstanceId='string',
    ReservedDBInstancesOfferingId='string',
    DBInstanceClass='string',
    Duration='string',
    ProductDescription='string',
    OfferingType='string',
    MultiAZ=True|False,
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **ReservedDBInstanceId** (*string*) – The reserved DB instance identifier filter value. Specify this parameter to show only the reservation that matches the specified reservation ID.

- **ReservedDBInstancesOfferingId** (*string*) – The offering identifier filter value. Specify this parameter to show only purchased reservations matching the specified offering identifier.
- **DBInstanceClass** (*string*) – The DB instance class filter value. Specify this parameter to show only those reservations matching the specified DB instances class.
- **Duration** (*string*) – The duration filter value, specified in years or seconds. Specify this parameter to show only reservations for this duration.

Valid Values: 1 | 3 | 31536000 | 94608000

- **ProductDescription** (*string*) – The product description filter value. Specify this parameter to show only those reservations matching the specified product description.
- **OfferingType** (*string*) – The offering type filter value. Specify this parameter to show only the available offerings matching the specified offering type.

Valid Values: "Partial Upfront" | "All Upfront" | "No Upfront"

- **MultiAZ** (*boolean*) – The Multi-AZ filter value. Specify this parameter to show only those reservations matching the specified Multi-AZ parameter.
- **Filters** (*list*) – This parameter is not currently supported.

- (*dict*) –

- * **Name** (*string*) – [REQUIRED]

- This parameter is not currently supported.

- * **Values** (*list*) – [REQUIRED]

- This parameter is not currently supported.

- (*string*) –

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

- The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

- The size of each page.

- **StartingToken** (*string*) –

- A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'ReservedDBInstances': [
    {
      'ReservedDBInstanceId': 'string',
      'ReservedDBInstancesOfferingId': 'string',
      'DBInstanceClass': 'string',
      'StartTime': datetime(2015, 1, 1),
      'Duration': 123,
      'FixedPrice': 123.0,
      'UsagePrice': 123.0,
```

```

        'CurrencyCode': 'string',
        'DBInstanceCount': 123,
        'ProductDescription': 'string',
        'OfferingType': 'string',
        'MultiAZ': True|False,
        'State': 'string',
        'RecurringCharges': [
            {
                'RecurringChargeAmount': 123.0,
                'RecurringChargeFrequency': 'string'
            },
        ]
    },
    'NextToken': 'string'
}

```

Response Structure

- *(dict)* –

Contains the result of a successful invocation of the DescribeReservedDBInstances action.

- **ReservedDBInstances** (*list*) –

A list of reserved DB instances.

- * *(dict)* –

This data type is used as a response element in the DescribeReservedDBInstances and PurchaseReservedDBInstancesOffering actions.

- **ReservedDBInstanceId** (*string*) –
The unique identifier for the reservation.
- **ReservedDBInstancesOfferingId** (*string*) –
The offering identifier.
- **DBInstanceClass** (*string*) –
The DB instance class for the reserved DB instance.
- **StartTime** (*datetime*) –
The time the reservation started.
- **Duration** (*integer*) –
The duration of the reservation in seconds.
- **FixedPrice** (*float*) –
The fixed price charged for this reserved DB instance.
- **UsagePrice** (*float*) –
The hourly price charged for this reserved DB instance.
- **CurrencyCode** (*string*) –
The currency code for the reserved DB instance.
- **DBInstanceCount** (*integer*) –
The number of reserved DB instances.
- **ProductDescription** (*string*) –
The description of the reserved DB instance.
- **OfferingType** (*string*) –

The offering type of this reserved DB instance.

- **MultiAZ** (*boolean*) –

Indicates if the reservation applies to Multi-AZ deployments.

- **State** (*string*) –

The state of the reserved DB instance.

- **RecurringCharges** (*list*) –

The recurring price charged to run this reserved DB instance.

- (*dict*) –

This data type is used as a response element in the DescribeReservedDBInstances and DescribeReservedDBInstancesOfferings actions.

- **RecurringChargeAmount** (*float*) –

The amount of the recurring charge.

- **RecurringChargeFrequency** (*string*) –

The frequency of the recurring charge.

- **NextToken** (*string*) –

A token to resume pagination.

class RDS.Paginator.**DescribeReservedDBInstancesOfferings**

```
paginator = client.get_paginator('describe_reserved_db_instances_offerings')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `RDS.Client.describe_reserved_db_instances_offerings()`.

Request Syntax

```
response_iterator = paginator.paginate(
    ReservedDBInstancesOfferingId='string',
    DBInstanceClass='string',
    Duration='string',
    ProductDescription='string',
    OfferingType='string',
    MultiAZ=True|False,
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **ReservedDBInstancesOfferingId** (*string*) – The offering identifier filter value. Specify this parameter to show only the available offering that matches the specified reservation identifier.

Example: 438012d3-4052-4cc7-b2e3-8d3372e0e706

- **DBInstanceClass** (*string*) – The DB instance class filter value. Specify this parameter to show only the available offerings matching the specified DB instance class.
- **Duration** (*string*) – Duration filter value, specified in years or seconds. Specify this parameter to show only reservations for this duration.

Valid Values: 1 | 3 | 31536000 | 94608000

- **ProductDescription** (*string*) – Product description filter value. Specify this parameter to show only the available offerings matching the specified product description.
- **OfferingType** (*string*) – The offering type filter value. Specify this parameter to show only the available offerings matching the specified offering type.

Valid Values: "Partial Upfront" | "All Upfront" | "No Upfront"

- **MultiAZ** (*boolean*) – The Multi-AZ filter value. Specify this parameter to show only the available offerings matching the specified Multi-AZ parameter.
- **Filters** (*list*) – This parameter is not currently supported.

- (*dict*) –

- * **Name** (*string*) – [REQUIRED]

- This parameter is not currently supported.

- * **Values** (*list*) – [REQUIRED]

- This parameter is not currently supported.

- (*string*) –

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'ReservedDBInstancesOfferings': [
    {
      'ReservedDBInstancesOfferingId': 'string',
      'DBInstanceClass': 'string',
      'Duration': 123,
      'FixedPrice': 123.0,
      'UsagePrice': 123.0,
      'CurrencyCode': 'string',
```

```
        'ProductDescription': 'string',
        'OfferingType': 'string',
        'MultiAZ': True|False,
        'RecurringCharges': [
            {
                'RecurringChargeAmount': 123.0,
                'RecurringChargeFrequency': 'string'
            },
        ],
    },
],
'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the result of a successful invocation of the DescribeReservedDBInstancesOfferings action.

- **ReservedDBInstancesOfferings** (*list*) –

A list of reserved DB instance offerings.

- * (*dict*) –

This data type is used as a response element in the DescribeReservedDBInstancesOfferings action.

- **ReservedDBInstancesOfferingId** (*string*) –

The offering identifier.

- **DBInstanceClass** (*string*) –

The DB instance class for the reserved DB instance.

- **Duration** (*integer*) –

The duration of the offering in seconds.

- **FixedPrice** (*float*) –

The fixed price charged for this offering.

- **UsagePrice** (*float*) –

The hourly price charged for this offering.

- **CurrencyCode** (*string*) –

The currency code for the reserved DB instance offering.

- **ProductDescription** (*string*) –

The database engine used by the offering.

- **OfferingType** (*string*) –

The offering type.

- **MultiAZ** (*boolean*) –

Indicates if the offering applies to Multi-AZ deployments.

- **RecurringCharges** (*list*) –

The recurring price charged to run this reserved DB instance.

- (*dict*) –

This data type is used as a response element in the DescribeReservedDBInstances and DescribeReservedDBInstancesOfferings actions.

- **RecurringChargeAmount** (*float*) –
The amount of the recurring charge.
- **RecurringChargeFrequency** (*string*) –
The frequency of the recurring charge.
- **NextToken** (*string*) –
A token to resume pagination.

class `RDS.Paginator.DownloadDBLogFilePortion`

```
paginator = client.get_paginator('download_db_log_file_portion')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `RDS.Client.download_db_log_file_portion()`.

Request Syntax

```
response_iterator = paginator.paginate(
    DBInstanceIdentifier='string',
    LogFileName='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **DBInstanceIdentifier** (*string*) – **[REQUIRED]**

The customer-assigned name of the DB instance that contains the log files you want to list.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **LogFileName** (*string*) – **[REQUIRED]**

The name of the log file to be downloaded.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
    'LogFileData': 'string',
    'AdditionalDataPending': True|False,
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

This data type is used as a response element to `DownloadDBLogFilePortion`.

- **LogFileData** (*string*) –

Entries from the specified log file.

- **AdditionalDataPending** (*boolean*) –

Boolean value that if true, indicates there is more data to be downloaded.

- **NextToken** (*string*) –

A token to resume pagination.

Waiters

The available waiters are:

- `RDS.Waiter.DBInstanceAvailable`
- `RDS.Waiter.DBInstanceDeleted`
- `RDS.Waiter.DBSnapshotCompleted`

class `RDS.Waiter.DBInstanceAvailable`

```
waiter = client.get_waiter('db_instance_available')
```

wait (***kwargs*)

Polls `RDS.Client.describe_db_instances()` every 30 seconds until a successful state is reached. An error is returned after 60 failed checks.

Request Syntax

```
waiter.wait(
    DBInstanceIdentifier='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **DBInstanceIdentifier** (*string*) – The user-supplied instance identifier. If this parameter is specified, information from only the specific DB instance is returned. This parameter isn't case-sensitive.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens
- **Filters** (*list*) – This parameter is not currently supported.
 - (*dict*) –
 - * **Name** (*string*) – [REQUIRED]
This parameter is not currently supported.
 - * **Values** (*list*) – [REQUIRED]
This parameter is not currently supported.
 - (*string*) –
- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: Minimum 20, maximum 100.

- **Marker** (*string*) – An optional pagination token provided by a previous `DescribeDBInstances` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Returns None

class `RDS.Waiter.DBInstanceDeleted`

```
waiter = client.get_waiter('db_instance_deleted')
```

wait (***kwargs*)

Polls `RDS.Client.describe_db_instances()` every 30 seconds until a successful state is reached. An error is returned after 60 failed checks.

Request Syntax

```
waiter.wait(
    DBInstanceIdentifier='string',
    Filters=[
        {
            'Name': 'string',
            'Values': [
                'string',
            ]
        },
    ],
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **DBInstanceIdentifier** (*string*) – The user-supplied instance identifier. If this parameter is specified, information from only the specific DB instance is returned. This parameter isn't case-sensitive.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
 - First character must be a letter
 - Cannot end with a hyphen or contain two consecutive hyphens
- **Filters** (*list*) – This parameter is not currently supported.
 - (*dict*) –
 - * **Name** (*string*) – [REQUIRED]
This parameter is not currently supported.
 - * **Values** (*list*) – [REQUIRED]
This parameter is not currently supported.
 - (*string*) –

- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: Minimum 20, maximum 100.

- **Marker** (*string*) – An optional pagination token provided by a previous `DescribeDBInstances` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Returns None

class `RDS.Waiter.DBSnapshotCompleted`

```
waiter = client.get_waiter('db_snapshot_completed')
```

wait (***kwargs*)

Polls `RDS.Client.describe_db_snapshots()` every 15 seconds until a successful state is reached. An error is returned after 40 failed checks.

Request Syntax

```
waiter.wait(  
    DBInstanceIdentifier='string',  
    DBSnapshotIdentifier='string',  
    SnapshotType='string',  
    Filters=[  
        {  
            'Name': 'string',  
            'Values': [  
                'string',  
            ]  
        },  
    ],  
    MaxRecords=123,  
    Marker='string'  
)
```

Parameters

- **DBInstanceIdentifier** (*string*) – A DB instance identifier to retrieve the list of DB snapshots for. This parameter cannot be used in conjunction with `DBSnapshotIdentifier`. This parameter is not case-sensitive.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **DBSnapshotIdentifier** (*string*) – A specific DB snapshot identifier to describe. This parameter cannot be used in conjunction with `DBInstanceIdentifier`. This value is stored as a lowercase string.

Constraints:

- Must be 1 to 255 alphanumeric characters.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.
- If this is the identifier of an automated snapshot, the `SnapshotType` parameter must also be specified.

- **SnapshotType** (*string*) – The type of snapshots that will be returned. Values can be “automated” or “manual.” If not specified, the returned results will include all snapshots types.

- **Filters** (*list*) – This parameter is not currently supported.

– (*dict*) –

* **Name** (*string*) – [REQUIRED]

This parameter is not currently supported.

* **Values** (*list*) – [REQUIRED]

This parameter is not currently supported.

· (*string*) –

- **MaxRecords** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: Minimum 20, maximum 100.

- **Marker** (*string*) – An optional pagination token provided by a previous `DescribeDBSnapshots` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

Returns None

Redshift

Table of Contents

- Redshift
 - Client
 - Paginators
 - Waiters

Client

`class Redshift.Client`

A low-level client representing Amazon Redshift:

```
import boto3

client = boto3.client('redshift')
```

These are the available methods:

- `authorize_cluster_security_group_ingress()`
- `authorize_snapshot_access()`
- `can_paginate()`
- `copy_cluster_snapshot()`
- `create_cluster()`
- `create_cluster_parameter_group()`
- `create_cluster_security_group()`
- `create_cluster_snapshot()`
- `create_cluster_subnet_group()`
- `create_event_subscription()`
- `create_hsm_client_certificate()`
- `create_hsm_configuration()`
- `create_snapshot_copy_grant()`
- `create_tags()`
- `delete_cluster()`
- `delete_cluster_parameter_group()`
- `delete_cluster_security_group()`
- `delete_cluster_snapshot()`
- `delete_cluster_subnet_group()`
- `delete_event_subscription()`
- `delete_hsm_client_certificate()`
- `delete_hsm_configuration()`
- `delete_snapshot_copy_grant()`
- `delete_tags()`
- `describe_cluster_parameter_groups()`
- `describe_cluster_parameters()`
- `describe_cluster_security_groups()`
- `describe_cluster_snapshots()`
- `describe_cluster_subnet_groups()`
- `describe_cluster_versions()`
- `describe_clusters()`
- `describe_default_cluster_parameters()`
- `describe_event_categories()`
- `describe_event_subscriptions()`
- `describe_events()`
- `describe_hsm_client_certificates()`
- `describe_hsm_configurations()`
- `describe_logging_status()`
- `describe_orderable_cluster_options()`
- `describe_reserved_node_offerings()`
- `describe_reserved_nodes()`
- `describe_resize()`
- `describe_snapshot_copy_grants()`
- `describe_tags()`

- `disable_logging()`
- `disable_snapshot_copy()`
- `enable_logging()`
- `enable_snapshot_copy()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_waiter()`
- `modify_cluster()`
- `modify_cluster_parameter_group()`
- `modify_cluster_subnet_group()`
- `modify_event_subscription()`
- `modify_snapshot_copy_retention_period()`
- `purchase_reserved_node_offering()`
- `reboot_cluster()`
- `reset_cluster_parameter_group()`
- `restore_from_cluster_snapshot()`
- `revoke_cluster_security_group_ingress()`
- `revoke_snapshot_access()`
- `rotate_encryption_key()`

authorize_cluster_security_group_ingress (***kwargs*)

Adds an inbound (ingress) rule to an Amazon Redshift security group. Depending on whether the application accessing your cluster is running on the Internet or an EC2 instance, you can authorize inbound access to either a Classless Interdomain Routing (CIDR) IP address range or an EC2 security group. You can add as many as 20 ingress rules to an Amazon Redshift security group.

Note: The EC2 security group must be defined in the AWS region where the cluster resides.

For an overview of CIDR blocks, see the Wikipedia article on [Classless Inter-Domain Routing](#).

You must also associate the security group with a cluster so that clients running on these IP addresses or the EC2 instance are authorized to connect to the cluster. For information about managing security groups, go to [Working with Security Groups](#) in the *Amazon Redshift Cluster Management Guide*.

Request Syntax

```
response = client.authorize_cluster_security_group_ingress(
    ClusterSecurityGroupName='string',
    CIDRIP='string',
    EC2SecurityGroupName='string',
    EC2SecurityGroupOwnerId='string'
)
```

Parameters

- **ClusterSecurityGroupName** (*string*) – **[REQUIRED]**
The name of the security group to which the ingress rule is added.
- **CIDRIP** (*string*) – The IP range to be added the Amazon Redshift security group.
- **EC2SecurityGroupName** (*string*) – The EC2 security group to be added the Amazon Redshift security group.
- **EC2SecurityGroupOwnerId** (*string*) – The AWS account number of the owner of the security group specified by the *EC2SecurityGroupName* parameter. The AWS Access Key ID is not an acceptable value.

Example: 111122223333

Return type dict

Returns

Response Syntax

```
{
  'ClusterSecurityGroup': {
    'ClusterSecurityGroupName': 'string',
    'Description': 'string',
    'EC2SecurityGroups': [
      {
        'Status': 'string',
        'EC2SecurityGroupName': 'string',
        'EC2SecurityGroupOwnerId': 'string',
        'Tags': [
          {
            'Key': 'string',
            'Value': 'string'
          }
        ]
      }
    ],
  },
  'IPRanges': [
    {
      'Status': 'string',
      'CIDRIP': 'string',
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        }
      ]
    }
  ],
  'Tags': [
    {
      'Key': 'string',
      'Value': 'string'
    }
  ]
}
```

Response Structure

- (*dict*) –

Describes a security group.

- **ClusterSecurityGroup** (*dict*) –

Describes a security group.

- * **ClusterSecurityGroupName** (*string*) –

The name of the cluster security group to which the operation was applied.

- * **Description** (*string*) –

A description of the security group.

- * **EC2SecurityGroups** (*list*) –

A list of EC2 security groups that are permitted to access clusters associated with this cluster security group.

- *(dict)* –
Describes an Amazon EC2 security group.
- **Status** (*string*) –
The status of the EC2 security group.
- **EC2SecurityGroupName** (*string*) –
The name of the EC2 Security Group.
- **EC2SecurityGroupOwnerId** (*string*) –
The AWS ID of the owner of the EC2 security group specified in the `EC2SecurityGroupName` field.
- **Tags** (*list*) –
The list of tags for the EC2 security group.
- *(dict)* –
A tag consisting of a name/value pair for a resource.
- **Key** (*string*) –
The key, or name, for the resource tag.
- **Value** (*string*) –
The value for the resource tag.
- * **IPRanges** (*list*) –
A list of IP ranges (CIDR blocks) that are permitted to access clusters associated with this cluster security group.
- *(dict)* –
Describes an IP range used in a security group.
- **Status** (*string*) –
The status of the IP range, for example, “authorized”.
- **CIDRIP** (*string*) –
The IP range in Classless Inter-Domain Routing (CIDR) notation.
- **Tags** (*list*) –
The list of tags for the IP range.
- *(dict)* –
A tag consisting of a name/value pair for a resource.
- **Key** (*string*) –
The key, or name, for the resource tag.
- **Value** (*string*) –
The value for the resource tag.
- * **Tags** (*list*) –
The list of tags for the cluster security group.
- *(dict)* –
A tag consisting of a name/value pair for a resource.
- **Key** (*string*) –
The key, or name, for the resource tag.
- **Value** (*string*) –
The value for the resource tag.

authorize_snapshot_access (**kwargs)

Authorizes the specified AWS customer account to restore the specified snapshot.

For more information about working with snapshots, go to [Amazon Redshift Snapshots](#) in the *Amazon Redshift Cluster Management Guide*.

Request Syntax

```
response = client.authorize_snapshot_access(
    SnapshotIdentifier='string',
    SnapshotClusterIdentifier='string',
    AccountWithRestoreAccess='string'
)
```

Parameters

- **SnapshotIdentifier** (*string*) – [REQUIRED]

The identifier of the snapshot the account is authorized to restore.

- **SnapshotClusterIdentifier** (*string*) – The identifier of the cluster the snapshot was created from. This parameter is required if your IAM user has a policy containing a snapshot resource element that specifies anything other than * for the cluster name.
- **AccountWithRestoreAccess** (*string*) – [REQUIRED]

The identifier of the AWS customer account authorized to restore the specified snapshot.

Return type dict

Returns**Response Syntax**

```
{
  'Snapshot': {
    'SnapshotIdentifier': 'string',
    'ClusterIdentifier': 'string',
    'SnapshotCreateTime': datetime(2015, 1, 1),
    'Status': 'string',
    'Port': 123,
    'AvailabilityZone': 'string',
    'ClusterCreateTime': datetime(2015, 1, 1),
    'MasterUsername': 'string',
    'ClusterVersion': 'string',
    'SnapshotType': 'string',
    'NodeType': 'string',
    'NumberOfNodes': 123,
    'DBName': 'string',
    'VpcId': 'string',
    'Encrypted': True|False,
    'KmsKeyId': 'string',
    'EncryptedWithHSM': True|False,
    'AccountsWithRestoreAccess': [
      {
        'AccountId': 'string'
      },
    ],
    'OwnerAccount': 'string',
    'TotalBackupSizeInMegabytes': 123.0,
    'ActualIncrementalBackupSizeInMegabytes': 123.0,
    'BackupProgressInMegabytes': 123.0,
```

```

'CurrentBackupRateInMegabytesPerSecond': 123.0,
'EstimatedSecondsToCompletion': 123,
'ElapsedTimeInSeconds': 123,
'SourceRegion': 'string',
'Tags': [
    {
        'Key': 'string',
        'Value': 'string'
    },
],
'RestorableNodeTypes': [
    'string',
]
}

```

Response Structure

- (*dict*) –

Describes a snapshot.

- **Snapshot** (*dict*) –

Describes a snapshot.

- * **SnapshotIdentifier** (*string*) –

The snapshot identifier that is provided in the request.

- * **ClusterIdentifier** (*string*) –

The identifier of the cluster for which the snapshot was taken.

- * **SnapshotCreateTime** (*datetime*) –

The time (UTC) when Amazon Redshift began the snapshot. A snapshot contains a copy of the cluster data as of this exact time.

- * **Status** (*string*) –

The snapshot status. The value of the status depends on the API operation used.

- CreateClusterSnapshot and CopyClusterSnapshot returns status as “creating”.
- DescribeClusterSnapshots returns status as “creating”, “available”, “final snapshot”, or “failed”.
- DeleteClusterSnapshot returns status as “deleted”.

- * **Port** (*integer*) –

The port that the cluster is listening on.

- * **AvailabilityZone** (*string*) –

The Availability Zone in which the cluster was created.

- * **ClusterCreateTime** (*datetime*) –

The time (UTC) when the cluster was originally created.

- * **MasterUsername** (*string*) –

The master user name for the cluster.

- * **ClusterVersion** (*string*) –

The version ID of the Amazon Redshift engine that is running on the cluster.

- * **SnapshotType** (*string*) –

The snapshot type. Snapshots created using `CreateClusterSnapshot` and `CopyClusterSnapshot` will be of type “manual”.

* **NodeType** (*string*) –

The node type of the nodes in the cluster.

* **NumberOfNodes** (*integer*) –

The number of nodes in the cluster.

* **DBName** (*string*) –

The name of the database that was created when the cluster was created.

* **VpcId** (*string*) –

The VPC identifier of the cluster if the snapshot is from a cluster in a VPC. Otherwise, this field is not in the output.

* **Encrypted** (*boolean*) –

If `true`, the data in the snapshot is encrypted at rest.

* **KmsKeyId** (*string*) –

The AWS Key Management Service (KMS) key ID of the encryption key that was used to encrypt data in the cluster from which the snapshot was taken.

* **EncryptedWithHSM** (*boolean*) –

A boolean that indicates whether the snapshot data is encrypted using the HSM keys of the source cluster. `true` indicates that the data is encrypted using HSM keys.

* **AccountsWithRestoreAccess** (*list*) –

A list of the AWS customer accounts authorized to restore the snapshot. Returns `null` if no accounts are authorized. Visible only to the snapshot owner.

· (*dict*) –

Describes an AWS customer account authorized to restore a snapshot.

· **AccountId** (*string*) –

The identifier of an AWS customer account authorized to restore a snapshot.

* **OwnerAccount** (*string*) –

For manual snapshots, the AWS customer account used to create or copy the snapshot. For automatic snapshots, the owner of the cluster. The owner can perform all snapshot actions, such as sharing a manual snapshot.

* **TotalBackupSizeInMegaBytes** (*float*) –

The size of the complete set of backup data that would be used to restore the cluster.

* **ActualIncrementalBackupSizeInMegaBytes** (*float*) –

The size of the incremental backup.

* **BackupProgressInMegaBytes** (*float*) –

The number of megabytes that have been transferred to the snapshot backup.

* **CurrentBackupRateInMegaBytesPerSecond** (*float*) –

The number of megabytes per second being transferred to the snapshot backup. Returns 0 for a completed backup.

* **EstimatedSecondsToCompletion** (*integer*) –

The estimate of the time remaining before the snapshot backup will complete. Returns 0 for a completed backup.

* **ElapsedTimeInSeconds** (*integer*) –

The amount of time an in-progress snapshot backup has been running, or the amount of time it took a completed backup to finish.

* **SourceRegion** (*string*) –

The source region from which the snapshot was copied.

* **Tags** (*list*) –

The list of tags for the cluster snapshot.

· (*dict*) –

A tag consisting of a name/value pair for a resource.

· **Key** (*string*) –

The key, or name, for the resource tag.

· **Value** (*string*) –

The value for the resource tag.

* **RestorableNodeTypes** (*list*) –

The list of node types that this cluster snapshot is able to restore into.

· (*string*) –

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

copy_cluster_snapshot (***kwargs*)

Copies the specified automated cluster snapshot to a new manual cluster snapshot. The source must be an automated snapshot and it must be in the available state.

When you delete a cluster, Amazon Redshift deletes any automated snapshots of the cluster. Also, when the retention period of the snapshot expires, Amazon Redshift automatically deletes it. If you want to keep an automated snapshot for a longer period, you can make a manual copy of the snapshot. Manual snapshots are retained until you delete them.

For more information about working with snapshots, go to [Amazon Redshift Snapshots](#) in the *Amazon Redshift Cluster Management Guide*.

Request Syntax

```
response = client.copy_cluster_snapshot(
    SourceSnapshotIdentifier='string',
    SourceSnapshotClusterIdentifier='string',
    TargetSnapshotIdentifier='string'
)
```

Parameters

- **SourceSnapshotIdentifier** (*string*) – [REQUIRED]

The identifier for the source snapshot.

Constraints:

- Must be the identifier for a valid automated snapshot whose state is available.

- **SourceSnapshotClusterIdentifier** (*string*) – The identifier of the cluster the source snapshot was created from. This parameter is required if your IAM user has a policy containing a snapshot resource element that specifies anything other than * for the cluster name.

Constraints:

- Must be the identifier for a valid cluster.

- **TargetSnapshotIdentifier** (*string*) – [REQUIRED]

The identifier given to the new manual snapshot.

Constraints:

- Cannot be null, empty, or blank.
- Must contain from 1 to 255 alphanumeric characters or hyphens.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.
- Must be unique for the AWS account that is making the request.

Return type dict

Returns

Response Syntax

```
{
  'Snapshot': {
    'SnapshotIdentifier': 'string',
    'ClusterIdentifier': 'string',
    'SnapshotCreateTime': datetime(2015, 1, 1),
    'Status': 'string',
    'Port': 123,
    'AvailabilityZone': 'string',
    'ClusterCreateTime': datetime(2015, 1, 1),
    'MasterUsername': 'string',
    'ClusterVersion': 'string',
    'SnapshotType': 'string',
    'NodeType': 'string',
    'NumberOfNodes': 123,
    'DBName': 'string',
    'VpcId': 'string',
    'Encrypted': True|False,
    'KmsKeyId': 'string',
    'EncryptedWithHSM': True|False,
    'AccountsWithRestoreAccess': [
      {
        'AccountId': 'string'
      },
    ],
    'OwnerAccount': 'string',
    'TotalBackupSizeInMegaBytes': 123.0,
    'ActualIncrementalBackupSizeInMegaBytes': 123.0,
    'BackupProgressInMegaBytes': 123.0,
    'CurrentBackupRateInMegaBytesPerSecond': 123.0,
    'EstimatedSecondsToCompletion': 123,
```

```

'ElapsedTimeInSeconds': 123,
'SourceRegion': 'string',
'Tags': [
    {
        'Key': 'string',
        'Value': 'string'
    },
],
'RestorableNodeTypes': [
    'string',
]
}
}

```

Response Structure

- (*dict*) –

Describes a snapshot.

- **Snapshot** (*dict*) –

Describes a snapshot.

- * **SnapshotIdentifier** (*string*) –

The snapshot identifier that is provided in the request.

- * **ClusterIdentifier** (*string*) –

The identifier of the cluster for which the snapshot was taken.

- * **SnapshotCreateTime** (*datetime*) –

The time (UTC) when Amazon Redshift began the snapshot. A snapshot contains a copy of the cluster data as of this exact time.

- * **Status** (*string*) –

The snapshot status. The value of the status depends on the API operation used.

- CreateClusterSnapshot and CopyClusterSnapshot returns status as “creating”.
- DescribeClusterSnapshots returns status as “creating”, “available”, “final snapshot”, or “failed”.
- DeleteClusterSnapshot returns status as “deleted”.

- * **Port** (*integer*) –

The port that the cluster is listening on.

- * **AvailabilityZone** (*string*) –

The Availability Zone in which the cluster was created.

- * **ClusterCreateTime** (*datetime*) –

The time (UTC) when the cluster was originally created.

- * **MasterUsername** (*string*) –

The master user name for the cluster.

- * **ClusterVersion** (*string*) –

The version ID of the Amazon Redshift engine that is running on the cluster.

- * **SnapshotType** (*string*) –

The snapshot type. Snapshots created using CreateClusterSnapshot and CopyClusterSnapshot will be of type “manual”.

- * **NodeType** (*string*) –

The node type of the nodes in the cluster.

- * **NumberOfNodes** (*integer*) –

The number of nodes in the cluster.

- * **DBName** (*string*) –

The name of the database that was created when the cluster was created.

- * **VpcId** (*string*) –

The VPC identifier of the cluster if the snapshot is from a cluster in a VPC. Otherwise, this field is not in the output.

- * **Encrypted** (*boolean*) –

If `true`, the data in the snapshot is encrypted at rest.

- * **KmsKeyId** (*string*) –

The AWS Key Management Service (KMS) key ID of the encryption key that was used to encrypt data in the cluster from which the snapshot was taken.

- * **EncryptedWithHSM** (*boolean*) –

A boolean that indicates whether the snapshot data is encrypted using the HSM keys of the source cluster. `true` indicates that the data is encrypted using HSM keys.

- * **AccountsWithRestoreAccess** (*list*) –

A list of the AWS customer accounts authorized to restore the snapshot. Returns `null` if no accounts are authorized. Visible only to the snapshot owner.

- (*dict*) –

- Describes an AWS customer account authorized to restore a snapshot.

- **AccountId** (*string*) –

- The identifier of an AWS customer account authorized to restore a snapshot.

- * **OwnerAccount** (*string*) –

For manual snapshots, the AWS customer account used to create or copy the snapshot. For automatic snapshots, the owner of the cluster. The owner can perform all snapshot actions, such as sharing a manual snapshot.

- * **TotalBackupSizeInMegaBytes** (*float*) –

The size of the complete set of backup data that would be used to restore the cluster.

- * **ActualIncrementalBackupSizeInMegaBytes** (*float*) –

The size of the incremental backup.

- * **BackupProgressInMegaBytes** (*float*) –

The number of megabytes that have been transferred to the snapshot backup.

- * **CurrentBackupRateInMegaBytesPerSecond** (*float*) –

The number of megabytes per second being transferred to the snapshot backup. Returns 0 for a completed backup.

- * **EstimatedSecondsToCompletion** (*integer*) –
The estimate of the time remaining before the snapshot backup will complete. Returns 0 for a completed backup.
- * **ElapsedTimeInSeconds** (*integer*) –
The amount of time an in-progress snapshot backup has been running, or the amount of time it took a completed backup to finish.
- * **SourceRegion** (*string*) –
The source region from which the snapshot was copied.
- * **Tags** (*list*) –
The list of tags for the cluster snapshot.
 - (*dict*) –
A tag consisting of a name/value pair for a resource.
 - **Key** (*string*) –
The key, or name, for the resource tag.
 - **Value** (*string*) –
The value for the resource tag.
- * **RestorableNodeTypes** (*list*) –
The list of node types that this cluster snapshot is able to restore into.
 - (*string*) –

create_cluster (***kwargs*)

Creates a new cluster. To create the cluster in virtual private cloud (VPC), you must provide cluster subnet group name. If you don't provide a cluster subnet group name or the cluster security group parameter, Amazon Redshift creates a non-VPC cluster, it associates the default cluster security group with the cluster. For more information about managing clusters, go to [Amazon Redshift Clusters](#) in the *Amazon Redshift Cluster Management Guide*.

Request Syntax

```
response = client.create_cluster(
    DBName='string',
    ClusterIdentifier='string',
    ClusterType='string',
    NodeType='string',
    MasterUsername='string',
    MasterUserPassword='string',
    ClusterSecurityGroups=[
        'string',
    ],
    VpcSecurityGroupIds=[
        'string',
    ],
    ClusterSubnetGroupName='string',
    AvailabilityZone='string',
    PreferredMaintenanceWindow='string',
    ClusterParameterGroupName='string',
    AutomatedSnapshotRetentionPeriod=123,
    Port=123,
    ClusterVersion='string',
    AllowVersionUpgrade=True|False,
    NumberOfNodes=123,
```

```
PubliclyAccessible=True|False,
Encrypted=True|False,
HsmClientCertificateIdentifier='string',
HsmConfigurationIdentifier='string',
ElasticIp='string',
Tags=[
    {
        'Key': 'string',
        'Value': 'string'
    },
],
KmsKeyId='string'
)
```

Parameters

- **DBName** (*string*) – The name of the first database to be created when the cluster is created.

To create additional databases after the cluster is created, connect to the cluster with a SQL client and use SQL commands to create a database. For more information, go to [Create a Database](#) in the Amazon Redshift Database Developer Guide.

Default: dev

Constraints:

- Must contain 1 to 64 alphanumeric characters.
- Must contain only lowercase letters.
- Cannot be a word that is reserved by the service. A list of reserved words can be found in [Reserved Words](#) in the Amazon Redshift Database Developer Guide.

- **ClusterIdentifier** (*string*) – **[REQUIRED]**

A unique identifier for the cluster. You use this identifier to refer to the cluster for any subsequent cluster operations such as deleting or modifying. The identifier also appears in the Amazon Redshift console.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens.
- Alphabetic characters must be lowercase.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.
- Must be unique for all clusters within an AWS account.

Example: myexamplecluster

- **ClusterType** (*string*) – The type of the cluster. When cluster type is specified as
 - `single-node`, the **NumberOfNodes** parameter is not required.
 - `multi-node`, the **NumberOfNodes** parameter is required.

Valid Values: `multi-node` | `single-node`

Default: `multi-node`

- **NodeType** (*string*) – **[REQUIRED]**

The node type to be provisioned for the cluster. For information about node types, go to [Working with Clusters](#) in the *Amazon Redshift Cluster Management Guide*.

Valid Values: `ds1.xlarge` | `ds1.8xlarge` | `ds2.xlarge` | `ds2.8xlarge` | `dc1.large` | `dc1.8xlarge`.

- **MasterUsername** (*string*) – [REQUIRED]

The user name associated with the master user account for the cluster that is being created.

Constraints:

- Must be 1 - 128 alphanumeric characters.
- First character must be a letter.
- Cannot be a reserved word. A list of reserved words can be found in [Reserved Words](#) in the Amazon Redshift Database Developer Guide.

- **MasterUserPassword** (*string*) – [REQUIRED]

The password associated with the master user account for the cluster that is being created.

Constraints:

- Must be between 8 and 64 characters in length.
- Must contain at least one uppercase letter.
- Must contain at least one lowercase letter.
- Must contain one number.
- Can be any printable ASCII character (ASCII code 33 to 126) except ‘ (single quote), ” (double quote), , /, @, or space.

- **ClusterSecurityGroups** (*list*) – A list of security groups to be associated with this cluster.

Default: The default cluster security group for Amazon Redshift.

– (*string*) –

- **VpcSecurityGroupIds** (*list*) – A list of Virtual Private Cloud (VPC) security groups to be associated with the cluster.

Default: The default VPC security group is associated with the cluster.

– (*string*) –

- **ClusterSubnetGroupName** (*string*) – The name of a cluster subnet group to be associated with this cluster.

If this parameter is not provided the resulting cluster will be deployed outside virtual private cloud (VPC).

- **AvailabilityZone** (*string*) – The EC2 Availability Zone (AZ) in which you want Amazon Redshift to provision the cluster. For example, if you have several EC2 instances running in a specific Availability Zone, then you might want the cluster to be provisioned in the same zone in order to decrease network latency.

Default: A random, system-chosen Availability Zone in the region that is specified by the endpoint.

Example: `us-east-1d`

Constraint: The specified Availability Zone must be in the same region as the current endpoint.

- **PreferredMaintenanceWindow** (*string*) – The weekly time range (in UTC) during which automated cluster maintenance can occur.

Format: `ddd:hh24:mi-ddd:hh24:mi`

Default: A 30-minute window selected at random from an 8-hour block of time per region, occurring on a random day of the week. For more information about the time blocks for each region, see [Maintenance Windows](#) in Amazon Redshift Cluster Management Guide.

Valid Days: Mon | Tue | Wed | Thu | Fri | Sat | Sun

Constraints: Minimum 30-minute window.

- **ClusterParameterGroupName** (*string*) – The name of the parameter group to be associated with this cluster.

Default: The default Amazon Redshift cluster parameter group. For information about the default parameter group, go to [Working with Amazon Redshift Parameter Groups](#)

Constraints:

- Must be 1 to 255 alphanumeric characters or hyphens.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.
- **AutomatedSnapshotRetentionPeriod** (*integer*) – The number of days that automated snapshots are retained. If the value is 0, automated snapshots are disabled. Even if automated snapshots are disabled, you can still create manual snapshots when you want with `CreateClusterSnapshot` .

Default: 1

Constraints: Must be a value from 0 to 35.

- **Port** (*integer*) – The port number on which the cluster accepts incoming connections.

The cluster is accessible only via the JDBC and ODBC connection strings. Part of the connection string requires the port on which the cluster will listen for incoming connections.

Default: 5439

Valid Values: 1150–65535

- **ClusterVersion** (*string*) – The version of the Amazon Redshift engine software that you want to deploy on the cluster.

The version selected runs on all the nodes in the cluster.

Constraints: Only version 1.0 is currently available.

Example: 1.0

- **AllowVersionUpgrade** (*boolean*) – If `true` , major version upgrades can be applied during the maintenance window to the Amazon Redshift engine that is running on the cluster.

When a new major version of the Amazon Redshift engine is released, you can request that the service automatically apply upgrades during the maintenance window to the Amazon Redshift engine that is running on your cluster.

Default: `true`

- **NumberOfNodes** (*integer*) – The number of compute nodes in the cluster. This parameter is required when the **ClusterType** parameter is specified as `multi-node` .

For information about determining how many nodes you need, go to [Working with Clusters](#) in the *Amazon Redshift Cluster Management Guide* .

If you don't specify this parameter, you get a single-node cluster. When requesting a multi-node cluster, you must specify the number of nodes that you want in the cluster.

Default: 1

Constraints: Value must be at least 1 and no more than 100.

- **PubliclyAccessible** (*boolean*) – If `true`, the cluster can be accessed from a public network.
- **Encrypted** (*boolean*) – If `true`, the data in the cluster is encrypted at rest.

Default: `false`

- **HsmClientCertificateIdentifier** (*string*) – Specifies the name of the HSM client certificate the Amazon Redshift cluster uses to retrieve the data encryption keys stored in an HSM.
- **HsmConfigurationIdentifier** (*string*) – Specifies the name of the HSM configuration that contains the information the Amazon Redshift cluster can use to retrieve and store keys in an HSM.
- **ElasticIp** (*string*) – The Elastic IP (EIP) address for the cluster.

Constraints: The cluster must be provisioned in EC2-VPC and publicly-accessible through an Internet gateway. For more information about provisioning clusters in EC2-VPC, go to [Supported Platforms to Launch Your Cluster](#) in the Amazon Redshift Cluster Management Guide.

- **Tags** (*list*) – A list of tag instances.
 - (*dict*) –

A tag consisting of a name/value pair for a resource.

* **Key** (*string*) –

The key, or name, for the resource tag.

* **Value** (*string*) –

The value for the resource tag.

- **KmsKeyId** (*string*) – The AWS Key Management Service (KMS) key ID of the encryption key that you want to use to encrypt data in the cluster.

Return type `dict`

Returns

Response Syntax

```
{
  'Cluster': {
    'ClusterIdentifier': 'string',
    'NodeType': 'string',
    'ClusterStatus': 'string',
    'ModifyStatus': 'string',
    'MasterUsername': 'string',
    'DBName': 'string',
    'Endpoint': {
      'Address': 'string',
      'Port': 123
    },
  },
  'ClusterCreateTime': datetime(2015, 1, 1),
  'AutomatedSnapshotRetentionPeriod': 123,
  'ClusterSecurityGroups': [
    {
      'ClusterSecurityGroupName': 'string',
      'Status': 'string'
    },
  ],
  'VpcSecurityGroups': [
    {
      'VpcSecurityGroupId': 'string',
      'Status': 'string'
    },
  ],
}
```

```
],
'ClusterParameterGroups': [
    {
        'ParameterGroupName': 'string',
        'ParameterApplyStatus': 'string',
        'ClusterParameterStatusList': [
            {
                'ParameterName': 'string',
                'ParameterApplyStatus': 'string',
                'ParameterApplyErrorDescription': 'string'
            },
        ]
    },
],
'ClusterSubnetGroupName': 'string',
'VpcId': 'string',
'AvailabilityZone': 'string',
'PreferredMaintenanceWindow': 'string',
'PendingModifiedValues': {
    'MasterUserPassword': 'string',
    'NodeType': 'string',
    'NumberOfNodes': 123,
    'ClusterType': 'string',
    'ClusterVersion': 'string',
    'AutomatedSnapshotRetentionPeriod': 123,
    'ClusterIdentifier': 'string'
},
'ClusterVersion': 'string',
'AllowVersionUpgrade': True|False,
'NumberOfNodes': 123,
'PubliclyAccessible': True|False,
'Encrypted': True|False,
'RestoreStatus': {
    'Status': 'string',
    'CurrentRestoreRateInMegaBytesPerSecond': 123.0,
    'SnapshotSizeInMegaBytes': 123,
    'ProgressInMegaBytes': 123,
    'ElapsedTimeInSeconds': 123,
    'EstimatedTimeToCompletionInSeconds': 123
},
'HsmStatus': {
    'HsmClientCertificateIdentifier': 'string',
    'HsmConfigurationIdentifier': 'string',
    'Status': 'string'
},
'ClusterSnapshotCopyStatus': {
    'DestinationRegion': 'string',
    'RetentionPeriod': 123,
    'SnapshotCopyGrantName': 'string'
},
'ClusterPublicKey': 'string',
'ClusterNodes': [
    {
        'NodeRole': 'string',
        'PrivateIpAddress': 'string',
        'PublicIpAddress': 'string'
    },
],
```

```

'ElasticIpStatus': {
    'ElasticIp': 'string',
    'Status': 'string'
},
'ClusterRevisionNumber': 'string',
'Tags': [
    {
        'Key': 'string',
        'Value': 'string'
    },
],
'KmsKeyId': 'string'
}

```

Response Structure

- (*dict*) –

Describes a cluster.

- **Cluster** (*dict*) –

Describes a cluster.

- * **ClusterIdentifier** (*string*) –

The unique identifier of the cluster.

- * **NodeType** (*string*) –

The node type for the nodes in the cluster.

- * **ClusterStatus** (*string*) –

The current state of this cluster. Possible values include `available`, `creating`, `deleting`, `rebooting`, `renaming`, and `resizing`.

- * **ModifyStatus** (*string*) –

The status of a modify operation, if any, initiated for the cluster.

- * **MasterUsername** (*string*) –

The master user name for the cluster. This name is used to connect to the database that is specified in **DBName**.

- * **DBName** (*string*) –

The name of the initial database that was created when the cluster was created. This same name is returned for the life of the cluster. If an initial database was not specified, a database named “dev” was created by default.

- * **Endpoint** (*dict*) –

The connection endpoint.

- **Address** (*string*) –

The DNS address of the Cluster.

- **Port** (*integer*) –

The port that the database engine is listening on.

- * **ClusterCreateTime** (*datetime*) –

The date and time that the cluster was created.

- * **AutomatedSnapshotRetentionPeriod** (*integer*) –

The number of days that automatic cluster snapshots are retained.

* **ClusterSecurityGroups** (*list*) –

A list of cluster security group that are associated with the cluster. Each security group is represented by an element that contains `ClusterSecurityGroup.Name` and `ClusterSecurityGroup.Status` subelements.

Cluster security groups are used when the cluster is not created in a VPC. Clusters that are created in a VPC use VPC security groups, which are listed by the **VpcSecurityGroups** parameter.

- (*dict*) –

- Describes a security group.

- **ClusterSecurityGroupName** (*string*) –

- The name of the cluster security group.

- **Status** (*string*) –

- The status of the cluster security group.

* **VpcSecurityGroups** (*list*) –

A list of Virtual Private Cloud (VPC) security groups that are associated with the cluster. This parameter is returned only if the cluster is in a VPC.

- (*dict*) –

- Describes the members of a VPC security group.

- **VpcSecurityGroupId** (*string*) –

- **Status** (*string*) –

* **ClusterParameterGroups** (*list*) –

The list of cluster parameter groups that are associated with this cluster. Each parameter group in the list is returned with its status.

- (*dict*) –

- Describes the status of a parameter group.

- **ParameterGroupName** (*string*) –

- The name of the cluster parameter group.

- **ParameterApplyStatus** (*string*) –

- The status of parameter updates.

- **ClusterParameterStatusList** (*list*) –

- The list of parameter statuses.

For more information about parameters and parameter groups, go to [Amazon Redshift Parameter Groups](#) in the *Amazon Redshift Cluster Management Guide*.

- (*dict*) –

- Describes the status of a parameter group.

- **ParameterName** (*string*) –

- The name of the parameter.

- **ParameterApplyStatus** (*string*) –

- The status of the parameter that indicates whether the parameter is in sync with the database, waiting for a cluster reboot, or encountered an error when being applied.

- The following are possible statuses and descriptions.

- `in-sync` : The parameter value is in sync with the database.
- `pending-reboot` : The parameter value will be applied after the cluster reboots.
- `applying` : The parameter value is being applied to the database.
- `invalid-parameter` : Cannot apply the parameter value because it has an invalid value or syntax.
- `apply-deferred` : The parameter contains static property changes. The changes are deferred until the cluster reboots.
- `apply-error` : Cannot connect to the cluster. The parameter change will be applied after the cluster reboots.
- `unknown-error` : Cannot apply the parameter change right now. The change will be applied after the cluster reboots.
- **ParameterApplyErrorDescription** (*string*) –

The error that prevented the parameter from being applied to the database.

- * **ClusterSubnetGroupName** (*string*) –

The name of the subnet group that is associated with the cluster. This parameter is valid only when the cluster is in a VPC.

- * **VpcId** (*string*) –

The identifier of the VPC the cluster is in, if the cluster is in a VPC.

- * **AvailabilityZone** (*string*) –

The name of the Availability Zone in which the cluster is located.

- * **PreferredMaintenanceWindow** (*string*) –

The weekly time range (in UTC) during which system maintenance can occur.

- * **PendingModifiedValues** (*dict*) –

If present, changes to the cluster are pending. Specific pending changes are identified by subelements.

- **MasterUserPassword** (*string*) –

The pending or in-progress change of the master user password for the cluster.

- **NodeType** (*string*) –

The pending or in-progress change of the cluster's node type.

- **NumberOfNodes** (*integer*) –

The pending or in-progress change of the number of nodes in the cluster.

- **ClusterType** (*string*) –

The pending or in-progress change of the cluster type.

- **ClusterVersion** (*string*) –

The pending or in-progress change of the service version.

- **AutomatedSnapshotRetentionPeriod** (*integer*) –

The pending or in-progress change of the automated snapshot retention period.

- **ClusterIdentifier** (*string*) –

The pending or in-progress change of the new identifier for the cluster.

* **ClusterVersion** (*string*) –

The version ID of the Amazon Redshift engine that is running on the cluster.

* **AllowVersionUpgrade** (*boolean*) –

If `true`, major version upgrades will be applied automatically to the cluster during the maintenance window.

* **NumberOfNodes** (*integer*) –

The number of compute nodes in the cluster.

* **PubliclyAccessible** (*boolean*) –

If `true`, the cluster can be accessed from a public network.

* **Encrypted** (*boolean*) –

If `true`, data in the cluster is encrypted at rest.

* **RestoreStatus** (*dict*) –

Describes the status of a cluster restore action. Returns null if the cluster was not created by restoring a snapshot.

· **Status** (*string*) –

The status of the restore action. Returns starting, restoring, completed, or failed.

· **CurrentRestoreRateInMegaBytesPerSecond** (*float*) –

The number of megabytes per second being transferred from the backup storage. Returns the average rate for a completed backup.

· **SnapshotSizeInMegaBytes** (*integer*) –

The size of the set of snapshot data used to restore the cluster.

· **ProgressInMegaBytes** (*integer*) –

The number of megabytes that have been transferred from snapshot storage.

· **ElapsedTimeInSeconds** (*integer*) –

The amount of time an in-progress restore has been running, or the amount of time it took a completed restore to finish.

· **EstimatedTimeToCompletionInSeconds** (*integer*) –

The estimate of the time remaining before the restore will complete. Returns 0 for a completed restore.

* **HsmStatus** (*dict*) –

Reports whether the Amazon Redshift cluster has finished applying any HSM settings changes specified in a modify cluster command.

Values: active, applying

· **HsmClientCertificateIdentifier** (*string*) –

Specifies the name of the HSM client certificate the Amazon Redshift cluster uses to retrieve the data encryption keys stored in an HSM.

· **HsmConfigurationIdentifier** (*string*) –

Specifies the name of the HSM configuration that contains the information the Amazon Redshift cluster can use to retrieve

and store keys in an HSM.

- **Status** (*string*) –

Reports whether the Amazon Redshift cluster has finished applying any HSM settings changes specified in a modify cluster command.

Values: active, applying

- * **ClusterSnapshotCopyStatus** (*dict*) –

Returns the destination region and retention period that are configured for cross-region snapshot copy.

- **DestinationRegion** (*string*) –

The destination region that snapshots are automatically copied to when cross-region snapshot copy is enabled.

- **RetentionPeriod** (*integer*) –

The number of days that automated snapshots are retained in the destination region after they are copied from a source region.

- **SnapshotCopyGrantName** (*string*) –

The name of the snapshot copy grant.

- * **ClusterPublicKey** (*string*) –

The public key for the cluster.

- * **ClusterNodes** (*list*) –

The nodes in a cluster.

- (*dict*) –

The identifier of a node in a cluster.

- **NodeRole** (*string*) –

Whether the node is a leader node or a compute node.

- **PrivateIPAddress** (*string*) –

The private IP address of a node within a cluster.

- **PublicIPAddress** (*string*) –

The public IP address of a node within a cluster.

- * **ElasticIpStatus** (*dict*) –

Describes the status of the elastic IP (EIP) address.

- **ElasticIp** (*string*) –

The elastic IP (EIP) address for the cluster.

- **Status** (*string*) –

Describes the status of the elastic IP (EIP) address.

- * **ClusterRevisionNumber** (*string*) –

The specific revision number of the database in the cluster.

- * **Tags** (*list*) –

The list of tags for the cluster.

- (*dict*) –

A tag consisting of a name/value pair for a resource.

- **Key** (*string*) –

The key, or name, for the resource tag.

- **Value** (*string*) –

The value for the resource tag.

- * **KmsKeyId** (*string*) –

The AWS Key Management Service (KMS) key ID of the encryption key used to encrypt data in the cluster.

create_cluster_parameter_group (***kwargs*)

Creates an Amazon Redshift parameter group.

Creating parameter groups is independent of creating clusters. You can associate a cluster with a parameter group when you create the cluster. You can also associate an existing cluster with a parameter group after the cluster is created by using `ModifyCluster`.

Parameters in the parameter group define specific behavior that applies to the databases you create on the cluster. For more information about parameters and parameter groups, go to [Amazon Redshift Parameter Groups](#) in the *Amazon Redshift Cluster Management Guide*.

Request Syntax

```
response = client.create_cluster_parameter_group(
    ParameterGroupName='string',
    ParameterGroupFamily='string',
    Description='string',
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **ParameterGroupName** (*string*) – [REQUIRED]

The name of the cluster parameter group.

Constraints:

- Must be 1 to 255 alphanumeric characters or hyphens
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.
- Must be unique withing your AWS account.

Note: This value is stored as a lower-case string.

- **ParameterGroupFamily** (*string*) – [REQUIRED]

The Amazon Redshift engine version to which the cluster parameter group applies. The cluster engine version determines the set of parameters.

To get a list of valid parameter group family names, you can call `DescribeClusterParameterGroups`. By default, Amazon Redshift returns a list of all the parameter groups that are owned by your AWS account, including the default parameter groups for each Amazon Redshift engine version. The parameter group family names associated with the default parameter groups provide you the valid values. For example, a valid family name is “redshift-1.0”.

- **Description** (*string*) – [REQUIRED]

A description of the parameter group.

- **Tags** (*list*) – A list of tag instances.

– (*dict*) –

A tag consisting of a name/value pair for a resource.

* **Key** (*string*) –

The key, or name, for the resource tag.

* **Value** (*string*) –

The value for the resource tag.

Return type `dict`

Returns

Response Syntax

```
{
  'ClusterParameterGroup': {
    'ParameterGroupName': 'string',
    'ParameterGroupFamily': 'string',
    'Description': 'string',
    'Tags': [
      {
        'Key': 'string',
        'Value': 'string'
      },
    ],
  }
}
```

Response Structure

• (*dict*) –

Describes a parameter group.

– **ClusterParameterGroup** (*dict*) –

Describes a parameter group.

* **ParameterGroupName** (*string*) –

The name of the cluster parameter group.

* **ParameterGroupFamily** (*string*) –

The name of the cluster parameter group family that this cluster parameter group is compatible with.

* **Description** (*string*) –

The description of the parameter group.

* **Tags** (*list*) –

The list of tags for the cluster parameter group.

· (*dict*) –

A tag consisting of a name/value pair for a resource.

· **Key** (*string*) –

The key, or name, for the resource tag.

· **Value** (*string*) –

The value for the resource tag.

create_cluster_security_group (***kwargs*)

Creates a new Amazon Redshift security group. You use security groups to control access to non-VPC clusters.

For information about managing security groups, go to [Amazon Redshift Cluster Security Groups](#) in the *Amazon Redshift Cluster Management Guide* .

Request Syntax

```
response = client.create_cluster_security_group(
    ClusterSecurityGroupName='string',
    Description='string',
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **ClusterSecurityGroupName** (*string*) – **[REQUIRED]**

The name for the security group. Amazon Redshift stores the value as a lower-case string.

Constraints:

- Must contain no more than 255 alphanumeric characters or hyphens.
- Must not be “Default”.
- Must be unique for all security groups that are created by your AWS account.

Example: examplesecuritygroup

- **Description** (*string*) – **[REQUIRED]**

A description for the security group.

- **Tags** (*list*) – A list of tag instances.
 - (*dict*) –

A tag consisting of a name/value pair for a resource.

* **Key** (*string*) –

The key, or name, for the resource tag.

* **Value** (*string*) –

The value for the resource tag.

Return type dict

Returns

Response Syntax

```
{
    'ClusterSecurityGroup': {
        'ClusterSecurityGroupName': 'string',
        'Description': 'string',
        'EC2SecurityGroups': [
            {
                'Status': 'string',
                'EC2SecurityGroupName': 'string',
                'EC2SecurityGroupOwnerId': 'string',
                'Tags': [
                    {
                        'Key': 'string',
                        'Value': 'string'
                    },
                ]
            },
        ]
    }
}
```

```

        ]
    },
],
'IPRanges': [
    {
        'Status': 'string',
        'CIDRIP': 'string',
        'Tags': [
            {
                'Key': 'string',
                'Value': 'string'
            },
        ]
    },
],
},
],
'Tags': [
    {
        'Key': 'string',
        'Value': 'string'
    },
],
]
}

```

Response Structure

- (*dict*) –

Describes a security group.

- **ClusterSecurityGroup** (*dict*) –

Describes a security group.

- * **ClusterSecurityGroupName** (*string*) –

The name of the cluster security group to which the operation was applied.

- * **Description** (*string*) –

A description of the security group.

- * **EC2SecurityGroups** (*list*) –

A list of EC2 security groups that are permitted to access clusters associated with this cluster security group.

- (*dict*) –

Describes an Amazon EC2 security group.

- **Status** (*string*) –

The status of the EC2 security group.

- **EC2SecurityGroupName** (*string*) –

The name of the EC2 Security Group.

- **EC2SecurityGroupOwnerId** (*string*) –

The AWS ID of the owner of the EC2 security group specified in the `EC2SecurityGroupName` field.

- **Tags** (*list*) –

The list of tags for the EC2 security group.

- (*dict*) –

A tag consisting of a name/value pair for a resource.

- **Key** (*string*) –
The key, or name, for the resource tag.
- **Value** (*string*) –
The value for the resource tag.
- * **IPRanges** (*list*) –
A list of IP ranges (CIDR blocks) that are permitted to access clusters associated with this cluster security group.
 - (*dict*) –
Describes an IP range used in a security group.
 - **Status** (*string*) –
The status of the IP range, for example, “authorized”.
 - **CIDRIP** (*string*) –
The IP range in Classless Inter-Domain Routing (CIDR) notation.
 - **Tags** (*list*) –
The list of tags for the IP range.
 - (*dict*) –
A tag consisting of a name/value pair for a resource.
 - **Key** (*string*) –
The key, or name, for the resource tag.
 - **Value** (*string*) –
The value for the resource tag.
- * **Tags** (*list*) –
The list of tags for the cluster security group.
 - (*dict*) –
A tag consisting of a name/value pair for a resource.
 - **Key** (*string*) –
The key, or name, for the resource tag.
 - **Value** (*string*) –
The value for the resource tag.

create_cluster_snapshot (**kwargs)

Creates a manual snapshot of the specified cluster. The cluster must be in the available state.

For more information about working with snapshots, go to [Amazon Redshift Snapshots](#) in the *Amazon Redshift Cluster Management Guide*.

Request Syntax

```
response = client.create_cluster_snapshot(  
    SnapshotIdentifier='string',  
    ClusterIdentifier='string',  
    Tags=[  
        {  
            'Key': 'string',  
            'Value': 'string'  
        },  
    ],  
)
```


Parameters

- **SnapshotIdentifier** (*string*) – **[REQUIRED]**

A unique identifier for the snapshot that you are requesting. This identifier must be unique for all snapshots within the AWS account.

Constraints:

- Cannot be null, empty, or blank
- Must contain from 1 to 255 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Example: my-snapshot-id

- **ClusterIdentifier** (*string*) – **[REQUIRED]**

The cluster identifier for which you want a snapshot.

- **Tags** (*list*) – A list of tag instances.
- (*dict*) –

A tag consisting of a name/value pair for a resource.

* **Key** (*string*) –

The key, or name, for the resource tag.

* **Value** (*string*) –

The value for the resource tag.

Return type dict

Returns

Response Syntax

```
{
  'Snapshot': {
    'SnapshotIdentifier': 'string',
    'ClusterIdentifier': 'string',
    'SnapshotCreateTime': datetime(2015, 1, 1),
    'Status': 'string',
    'Port': 123,
    'AvailabilityZone': 'string',
    'ClusterCreateTime': datetime(2015, 1, 1),
    'MasterUsername': 'string',
    'ClusterVersion': 'string',
    'SnapshotType': 'string',
    'NodeType': 'string',
    'NumberOfNodes': 123,
    'DBName': 'string',
    'VpcId': 'string',
    'Encrypted': True|False,
    'KmsKeyId': 'string',
    'EncryptedWithHSM': True|False,
    'AccountsWithRestoreAccess': [
      {
        'AccountId': 'string'
      },
    ],
    'OwnerAccount': 'string',
    'TotalBackupSizeInMegaBytes': 123.0,
    'ActualIncrementalBackupSizeInMegaBytes': 123.0,
    'BackupProgressInMegaBytes': 123.0,
    'CurrentBackupRateInMegaBytesPerSecond': 123.0,
    'EstimatedSecondsToCompletion': 123,
```

```
'ElapsedTimeInSeconds': 123,
'SourceRegion': 'string',
'Tags': [
    {
        'Key': 'string',
        'Value': 'string'
    },
],
'RestorableNodeTypes': [
    'string',
]
}
```

Response Structure

- (*dict*) –

Describes a snapshot.

- **Snapshot** (*dict*) –

Describes a snapshot.

- * **SnapshotIdentifier** (*string*) –

The snapshot identifier that is provided in the request.

- * **ClusterIdentifier** (*string*) –

The identifier of the cluster for which the snapshot was taken.

- * **SnapshotCreateTime** (*datetime*) –

The time (UTC) when Amazon Redshift began the snapshot. A snapshot contains a copy of the cluster data as of this exact time.

- * **Status** (*string*) –

The snapshot status. The value of the status depends on the API operation used.

- CreateClusterSnapshot and CopyClusterSnapshot returns status as “creating”.
- DescribeClusterSnapshots returns status as “creating”, “available”, “final snapshot”, or “failed”.
- DeleteClusterSnapshot returns status as “deleted”.

- * **Port** (*integer*) –

The port that the cluster is listening on.

- * **AvailabilityZone** (*string*) –

The Availability Zone in which the cluster was created.

- * **ClusterCreateTime** (*datetime*) –

The time (UTC) when the cluster was originally created.

- * **MasterUsername** (*string*) –

The master user name for the cluster.

- * **ClusterVersion** (*string*) –

The version ID of the Amazon Redshift engine that is running on the cluster.

- * **SnapshotType** (*string*) –

The snapshot type. Snapshots created using CreateClusterSnapshot and CopyClusterSnapshot will be of type “manual”.

- * **NodeType** (*string*) –
The node type of the nodes in the cluster.
- * **NumberOfNodes** (*integer*) –
The number of nodes in the cluster.
- * **DBName** (*string*) –
The name of the database that was created when the cluster was created.
- * **VpcId** (*string*) –
The VPC identifier of the cluster if the snapshot is from a cluster in a VPC. Otherwise, this field is not in the output.
- * **Encrypted** (*boolean*) –
If `true`, the data in the snapshot is encrypted at rest.
- * **KmsKeyId** (*string*) –
The AWS Key Management Service (KMS) key ID of the encryption key that was used to encrypt data in the cluster from which the snapshot was taken.
- * **EncryptedWithHSM** (*boolean*) –
A boolean that indicates whether the snapshot data is encrypted using the HSM keys of the source cluster. `true` indicates that the data is encrypted using HSM keys.
- * **AccountsWithRestoreAccess** (*list*) –
A list of the AWS customer accounts authorized to restore the snapshot. Returns `null` if no accounts are authorized. Visible only to the snapshot owner.
 - (*dict*) –
Describes an AWS customer account authorized to restore a snapshot.
 - **AccountId** (*string*) –
The identifier of an AWS customer account authorized to restore a snapshot.
- * **OwnerAccount** (*string*) –
For manual snapshots, the AWS customer account used to create or copy the snapshot. For automatic snapshots, the owner of the cluster. The owner can perform all snapshot actions, such as sharing a manual snapshot.
- * **TotalBackupSizeInMegaBytes** (*float*) –
The size of the complete set of backup data that would be used to restore the cluster.
- * **ActualIncrementalBackupSizeInMegaBytes** (*float*) –
The size of the incremental backup.
- * **BackupProgressInMegaBytes** (*float*) –
The number of megabytes that have been transferred to the snapshot backup.
- * **CurrentBackupRateInMegaBytesPerSecond** (*float*) –
The number of megabytes per second being transferred to the snapshot backup. Returns 0 for a completed backup.

- * **EstimatedSecondsToCompletion** (*integer*) –
The estimate of the time remaining before the snapshot backup will complete. Returns 0 for a completed backup.
- * **ElapsedTimeInSeconds** (*integer*) –
The amount of time an in-progress snapshot backup has been running, or the amount of time it took a completed backup to finish.
- * **SourceRegion** (*string*) –
The source region from which the snapshot was copied.
- * **Tags** (*list*) –
The list of tags for the cluster snapshot.
 - (*dict*) –
A tag consisting of a name/value pair for a resource.
 - **Key** (*string*) –
The key, or name, for the resource tag.
 - **Value** (*string*) –
The value for the resource tag.
- * **RestorableNodeTypes** (*list*) –
The list of node types that this cluster snapshot is able to restore into.
 - (*string*) –

create_cluster_subnet_group (***kwargs*)

Creates a new Amazon Redshift subnet group. You must provide a list of one or more subnets in your existing Amazon Virtual Private Cloud (Amazon VPC) when creating Amazon Redshift subnet group.

For information about subnet groups, go to [Amazon Redshift Cluster Subnet Groups](#) in the *Amazon Redshift Cluster Management Guide*.

Request Syntax

```
response = client.create_cluster_subnet_group(
    ClusterSubnetGroupName='string',
    Description='string',
    SubnetIds=[
        'string',
    ],
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **ClusterSubnetGroupName** (*string*) – [REQUIRED]

The name for the subnet group. Amazon Redshift stores the value as a lowercase string.

Constraints:

- Must contain no more than 255 alphanumeric characters or hyphens.
- Must not be “Default”.

- Must be unique for all subnet groups that are created by your AWS account.

Example: `examplesubnetgroup`

- **Description** (*string*) – **[REQUIRED]**

A description for the subnet group.

- **SubnetIds** (*list*) – **[REQUIRED]**

An array of VPC subnet IDs. A maximum of 20 subnets can be modified in a single request.

- (*string*) –

- **Tags** (*list*) – A list of tag instances.

- (*dict*) –

A tag consisting of a name/value pair for a resource.

- * **Key** (*string*) –

The key, or name, for the resource tag.

- * **Value** (*string*) –

The value for the resource tag.

Return type dict

Returns

Response Syntax

```
{
  'ClusterSubnetGroup': {
    'ClusterSubnetGroupName': 'string',
    'Description': 'string',
    'VpcId': 'string',
    'SubnetGroupStatus': 'string',
    'Subnets': [
      {
        'SubnetIdentifier': 'string',
        'SubnetAvailabilityZone': {
          'Name': 'string'
        },
        'SubnetStatus': 'string'
      },
    ],
    'Tags': [
      {
        'Key': 'string',
        'Value': 'string'
      },
    ],
  ]
}
```

Response Structure

- (*dict*) –

Describes a subnet group.

- **ClusterSubnetGroup** (*dict*) –

Describes a subnet group.

- * **ClusterSubnetGroupName** (*string*) –

The name of the cluster subnet group.

- * **Description** (*string*) –

The description of the cluster subnet group.

- * **VpcId** (*string*) –

The VPC ID of the cluster subnet group.

- * **SubnetGroupStatus** (*string*) –

The status of the cluster subnet group. Possible values are `Complete`, `Incomplete` and `Invalid`.

- * **Subnets** (*list*) –

A list of the VPC Subnet elements.

- (*dict*) –

Describes a subnet.

- **SubnetIdentifier** (*string*) –

The identifier of the subnet.

- **SubnetAvailabilityZone** (*dict*) –

Describes an availability zone.

- **Name** (*string*) –

The name of the availability zone.

- **SubnetStatus** (*string*) –

The status of the subnet.

- * **Tags** (*list*) –

The list of tags for the cluster subnet group.

- (*dict*) –

A tag consisting of a name/value pair for a resource.

- **Key** (*string*) –

The key, or name, for the resource tag.

- **Value** (*string*) –

The value for the resource tag.

create_event_subscription (***kwargs*)

Creates an Amazon Redshift event notification subscription. This action requires an ARN (Amazon Resource Name) of an Amazon SNS topic created by either the Amazon Redshift console, the Amazon SNS console, or the Amazon SNS API. To obtain an ARN with Amazon SNS, you must create a topic in Amazon SNS and subscribe to the topic. The ARN is displayed in the SNS console.

You can specify the source type, and lists of Amazon Redshift source IDs, event categories, and event severities. Notifications will be sent for all events you want that match those criteria. For example, you can specify source type = cluster, source ID = my-cluster-1 and mycluster2, event categories = Availability, Backup, and severity = ERROR. The subscription will only send notifications for those ERROR events in the Availability and Backup categories for the specified clusters.

If you specify both the source type and source IDs, such as source type = cluster and source identifier = my-cluster-1, notifications will be sent for all the cluster events for my-cluster-1. If you specify a source type but do not specify a source identifier, you will receive notice of the events for the objects of that type in your AWS account. If you do not specify either the `SourceType` nor the `SourceIdentifier`, you will be notified of events generated from all Amazon Redshift sources belonging to your AWS account. You must specify a source type if you specify a source ID.

Request Syntax

```

response = client.create_event_subscription(
    SubscriptionName='string',
    SnsTopicArn='string',
    SourceType='string',
    SourceIds=[
        'string',
    ],
    EventCategories=[
        'string',
    ],
    Severity='string',
    Enabled=True|False,
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)

```

Parameters

- **SubscriptionName** (*string*) – [REQUIRED]

The name of the event subscription to be created.

Constraints:

- Cannot be null, empty, or blank.
- Must contain from 1 to 255 alphanumeric characters or hyphens.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.

- **SnsTopicArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the Amazon SNS topic used to transmit the event notifications. The ARN is created by Amazon SNS when you create a topic and subscribe to it.

- **SourceType** (*string*) – The type of source that will be generating the events. For example, if you want to be notified of events generated by a cluster, you would set this parameter to cluster. If this value is not specified, events are returned for all Amazon Redshift objects in your AWS account. You must specify a source type in order to specify source IDs.

Valid values: cluster, cluster-parameter-group, cluster-security-group, and cluster-snapshot.

- **SourceIds** (*list*) – A list of one or more identifiers of Amazon Redshift source objects. All of the objects must be of the same type as was specified in the source type parameter. The event subscription will return only events generated by the specified objects. If not specified, then events are returned for all objects within the source type specified.

Example: my-cluster-1, my-cluster-2

Example: my-snapshot-20131010

– (*string*) –

- **EventCategories** (*list*) – Specifies the Amazon Redshift event categories to be published by the event notification subscription.

Values: Configuration, Management, Monitoring, Security

– (*string*) –

- **Severity** (*string*) – Specifies the Amazon Redshift event severity to be published by the event notification subscription.

Values: ERROR, INFO

- **Enabled** (*boolean*) – A Boolean value; set to `true` to activate the subscription, set to `false` to create the subscription but not active it.
- **Tags** (*list*) – A list of tag instances.
 - (*dict*) –

A tag consisting of a name/value pair for a resource.

- * **Key** (*string*) –

The key, or name, for the resource tag.

- * **Value** (*string*) –

The value for the resource tag.

Return type dict

Returns

Response Syntax

```
{
  'EventSubscription': {
    'CustomerAwsId': 'string',
    'CustSubscriptionId': 'string',
    'SnsTopicArn': 'string',
    'Status': 'string',
    'SubscriptionCreationTime': datetime(2015, 1, 1),
    'SourceType': 'string',
    'SourceIdsList': [
      'string',
    ],
    'EventCategoriesList': [
      'string',
    ],
    'Severity': 'string',
    'Enabled': True|False,
    'Tags': [
      {
        'Key': 'string',
        'Value': 'string'
      },
    ],
  },
}
```

Response Structure

- (*dict*) –
 - **EventSubscription** (*dict*) –
 - * **CustomerAwsId** (*string*) –

The AWS customer account associated with the Amazon Redshift event notification subscription.
 - * **CustSubscriptionId** (*string*) –

The name of the Amazon Redshift event notification subscription.
 - * **SnsTopicArn** (*string*) –

The Amazon Resource Name (ARN) of the Amazon SNS topic used by the event notification subscription.

* **Status** (*string*) –

The status of the Amazon Redshift event notification subscription.

Constraints:

- Can be one of the following: active | no-permission | topic-not-exist
- The status “no-permission” indicates that Amazon Redshift no longer has permission to post to the Amazon SNS topic. The status “topic-not-exist” indicates that the topic was deleted after the subscription was created.

* **SubscriptionCreationTime** (*datetime*) –

The date and time the Amazon Redshift event notification subscription was created.

* **SourceType** (*string*) –

The source type of the events returned the Amazon Redshift event notification, such as cluster, or cluster-snapshot.

* **SourceIdsList** (*list*) –

A list of the sources that publish events to the Amazon Redshift event notification subscription.

- (*string*) –

* **EventCategoriesList** (*list*) –

The list of Amazon Redshift event categories specified in the event notification subscription.

Values: Configuration, Management, Monitoring, Security

- (*string*) –

* **Severity** (*string*) –

The event severity specified in the Amazon Redshift event notification subscription.

Values: ERROR, INFO

* **Enabled** (*boolean*) –

A Boolean value indicating whether the subscription is enabled. `true` indicates the subscription is enabled.

* **Tags** (*list*) –

The list of tags for the event subscription.

- (*dict*) –

A tag consisting of a name/value pair for a resource.

- **Key** (*string*) –

The key, or name, for the resource tag.

- **Value** (*string*) –

The value for the resource tag.

create_hsm_client_certificate (**kwargs)

Creates an HSM client certificate that an Amazon Redshift cluster will use to connect to the client’s HSM in order to store and retrieve the keys used to encrypt the cluster databases.

The command returns a public key, which you must store in the HSM. In addition to creating the HSM certificate, you must create an Amazon Redshift HSM configuration that provides a cluster the information needed to store and use encryption keys in the HSM. For more information, go to [Hardware Security Modules](#) in the Amazon Redshift Cluster Management Guide.

Request Syntax

```
response = client.create_hsm_client_certificate(
    HsmClientCertificateIdentifier='string',
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **HsmClientCertificateIdentifier** (*string*) – [REQUIRED]

The identifier to be assigned to the new HSM client certificate that the cluster will use to connect to the HSM to use the database encryption keys.

- **Tags** (*list*) – A list of tag instances.

– (*dict*) –

A tag consisting of a name/value pair for a resource.

* **Key** (*string*) –

The key, or name, for the resource tag.

* **Value** (*string*) –

The value for the resource tag.

Return type dict

Returns

Response Syntax

```
{
  'HsmClientCertificate': {
    'HsmClientCertificateIdentifier': 'string',
    'HsmClientCertificatePublicKey': 'string',
    'Tags': [
      {
        'Key': 'string',
        'Value': 'string'
      },
    ],
  },
}
```

Response Structure

- (*dict*) –

Returns information about an HSM client certificate. The certificate is stored in a secure Hardware Storage Module (HSM), and used by the Amazon Redshift cluster to encrypt data files.

– **HsmClientCertificate** (*dict*) –

Returns information about an HSM client certificate. The certificate is stored in a secure Hardware Storage Module (HSM), and used by the Amazon Redshift cluster to encrypt data files.

* **HsmClientCertificateIdentifier** (*string*) –

The identifier of the HSM client certificate.

* **HsmClientCertificatePublicKey** (*string*) –

The public key that the Amazon Redshift cluster will use to connect to the HSM. You must register the public key in the HSM.

* **Tags** (*list*) –

The list of tags for the HSM client certificate.

· (*dict*) –

A tag consisting of a name/value pair for a resource.

· **Key** (*string*) –

The key, or name, for the resource tag.

· **Value** (*string*) –

The value for the resource tag.

create_hsm_configuration (***kwargs*)

Creates an HSM configuration that contains the information required by an Amazon Redshift cluster to store and use database encryption keys in a Hardware Security Module (HSM). After creating the HSM configuration, you can specify it as a parameter when creating a cluster. The cluster will then store its encryption keys in the HSM.

In addition to creating an HSM configuration, you must also create an HSM client certificate. For more information, go to [Hardware Security Modules](#) in the Amazon Redshift Cluster Management Guide.

Request Syntax

```
response = client.create_hsm_configuration(
    HsmConfigurationIdentifier='string',
    Description='string',
    HsmIpAddress='string',
    HsmPartitionName='string',
    HsmPartitionPassword='string',
    HsmServerPublicCertificate='string',
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
)
```

Parameters

• **HsmConfigurationIdentifier** (*string*) – **[REQUIRED]**

The identifier to be assigned to the new Amazon Redshift HSM configuration.

• **Description** (*string*) – **[REQUIRED]**

A text description of the HSM configuration to be created.

• **HsmIpAddress** (*string*) – **[REQUIRED]**

The IP address that the Amazon Redshift cluster must use to access the HSM.

• **HsmPartitionName** (*string*) – **[REQUIRED]**

The name of the partition in the HSM where the Amazon Redshift clusters will store their database encryption keys.

• **HsmPartitionPassword** (*string*) – **[REQUIRED]**

The password required to access the HSM partition.

- **HsmServerPublicCertificate** (*string*) – [REQUIRED]

The HSMs public certificate file. When using Cloud HSM, the file name is server.pem.

- **Tags** (*list*) – A list of tag instances.
 - (*dict*) –

A tag consisting of a name/value pair for a resource.

- * **Key** (*string*) –

The key, or name, for the resource tag.

- * **Value** (*string*) –

The value for the resource tag.

Return type dict

Returns

Response Syntax

```
{
  'HsmConfiguration': {
    'HsmConfigurationIdentifier': 'string',
    'Description': 'string',
    'HsmIpAddress': 'string',
    'HsmPartitionName': 'string',
    'Tags': [
      {
        'Key': 'string',
        'Value': 'string'
      },
    ]
  }
}
```

Response Structure

- (*dict*) –

Returns information about an HSM configuration, which is an object that describes to Amazon Redshift clusters the information they require to connect to an HSM where they can store database encryption keys.

- **HsmConfiguration** (*dict*) –

Returns information about an HSM configuration, which is an object that describes to Amazon Redshift clusters the information they require to connect to an HSM where they can store database encryption keys.

- * **HsmConfigurationIdentifier** (*string*) –

The name of the Amazon Redshift HSM configuration.

- * **Description** (*string*) –

A text description of the HSM configuration.

- * **HsmIpAddress** (*string*) –

The IP address that the Amazon Redshift cluster must use to access the HSM.

- * **HsmPartitionName** (*string*) –

The name of the partition in the HSM where the Amazon Redshift clusters will store their database encryption keys.

* **Tags** (*list*) –

The list of tags for the HSM configuration.

· (*dict*) –

A tag consisting of a name/value pair for a resource.

· **Key** (*string*) –

The key, or name, for the resource tag.

· **Value** (*string*) –

The value for the resource tag.

create_snapshot_copy_grant (***kwargs*)

Creates a snapshot copy grant that permits Amazon Redshift to use a customer master key (CMK) from AWS Key Management Service (AWS KMS) to encrypt copied snapshots in a destination region.

For more information about managing snapshot copy grants, go to [Amazon Redshift Database Encryption](#) in the *Amazon Redshift Cluster Management Guide*.

Request Syntax

```
response = client.create_snapshot_copy_grant(
    SnapshotCopyGrantName='string',
    KmsKeyId='string',
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

• **SnapshotCopyGrantName** (*string*) – **[REQUIRED]**

The name of the snapshot copy grant. This name must be unique in the region for the AWS account.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens.
- Alphabetic characters must be lowercase.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.
- Must be unique for all clusters within an AWS account.

• **KmsKeyId** (*string*) – The unique identifier of the customer master key (CMK) to which to grant Amazon Redshift permission. If no key is specified, the default key is used.

• **Tags** (*list*) – A list of tag instances.

– (*dict*) –

A tag consisting of a name/value pair for a resource.

* **Key** (*string*) –

The key, or name, for the resource tag.

* **Value** (*string*) –

The value for the resource tag.

Return type dict

Returns

Response Syntax

```
{
  'SnapshotCopyGrant': {
    'SnapshotCopyGrantName': 'string',
    'KmsKeyId': 'string',
    'Tags': [
      {
        'Key': 'string',
        'Value': 'string'
      },
    ],
  }
}
```

Response Structure

- *(dict)* –

The snapshot copy grant that grants Amazon Redshift permission to encrypt copied snapshots with the specified customer master key (CMK) from AWS KMS in the destination region.

For more information about managing snapshot copy grants, go to [Amazon Redshift Database Encryption](#) in the *Amazon Redshift Cluster Management Guide*.

- **SnapshotCopyGrant** (*dict*) –

The snapshot copy grant that grants Amazon Redshift permission to encrypt copied snapshots with the specified customer master key (CMK) from AWS KMS in the destination region.

For more information about managing snapshot copy grants, go to [Amazon Redshift Database Encryption](#) in the *Amazon Redshift Cluster Management Guide*.

- * **SnapshotCopyGrantName** (*string*) –

The name of the snapshot copy grant.

- * **KmsKeyId** (*string*) –

The unique identifier of the customer master key (CMK) in AWS KMS to which Amazon Redshift is granted permission.

- * **Tags** (*list*) –

A list of tag instances.

- (*dict*) –

A tag consisting of a name/value pair for a resource.

- **Key** (*string*) –

The key, or name, for the resource tag.

- **Value** (*string*) –

The value for the resource tag.

create_tags (***kwargs*)

Adds one or more tags to a specified resource.

A resource can have up to 10 tags. If you try to create more than 10 tags for a resource, you will receive an error and the attempt will fail.

If you specify a key that already exists for the resource, the value for that key will be updated with the new value.

Request Syntax

```
response = client.create_tags(
    ResourceName='string',
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Parameters

- **ResourceName** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) to which you want to add the tag or tags. For example, `arn:aws:redshift:us-east-1:123456789:cluster:t1`.

- **Tags** (*list*) – [REQUIRED]

One or more name/value pairs to add as tags to the specified resource. Each tag name is passed in with the parameter `Key` and the corresponding value is passed in with the parameter `Value`. The `Key` and `Value` parameters are separated by a comma (,). Separate multiple tags with a space. For example, `--tags "Key"="owner", "Value"="admin" "Key"="environment", "Value"="test" "Key"="version", "Value"="1.0"`.

– (*dict*) –

A tag consisting of a name/value pair for a resource.

* **Key** (*string*) –

The key, or name, for the resource tag.

* **Value** (*string*) –

The value for the resource tag.

Returns None

delete_cluster (***kwargs*)

Deletes a previously provisioned cluster. A successful response from the web service indicates that the request was received correctly. Use `DescribeClusters` to monitor the status of the deletion. The delete operation cannot be canceled or reverted once submitted. For more information about managing clusters, go to [Amazon Redshift Clusters](#) in the *Amazon Redshift Cluster Management Guide*.

If you want to shut down the cluster and retain it for future use, set `SkipFinalClusterSnapshot` to `false` and specify a name for `FinalClusterSnapshotIdentifier`. You can later restore this snapshot to resume using the cluster. If a final cluster snapshot is requested, the status of the cluster will be “final-snapshot” while the snapshot is being taken, then it’s “deleting” once Amazon Redshift begins deleting the cluster.

For more information about managing clusters, go to [Amazon Redshift Clusters](#) in the *Amazon Redshift Cluster Management Guide*.

Request Syntax

```
response = client.delete_cluster(
    ClusterIdentifier='string',
    SkipFinalClusterSnapshot=True|False,
```

```
FinalClusterSnapshotIdentifier='string'  
)
```

Parameters

- **ClusterIdentifier** (*string*) – [REQUIRED]

The identifier of the cluster to be deleted.

Constraints:

- Must contain lowercase characters.
 - Must contain from 1 to 63 alphanumeric characters or hyphens.
 - First character must be a letter.
 - Cannot end with a hyphen or contain two consecutive hyphens.
- **SkipFinalClusterSnapshot** (*boolean*) – Determines whether a final snapshot of the cluster is created before Amazon Redshift deletes the cluster. If `true`, a final cluster snapshot is not created. If `false`, a final cluster snapshot is created before the cluster is deleted.

Note: The *FinalClusterSnapshotIdentifier* parameter must be specified if *SkipFinalClusterSnapshot* is `false`.

Default: `false`

- **FinalClusterSnapshotIdentifier** (*string*) – The identifier of the final snapshot that is to be created immediately before deleting the cluster. If this parameter is provided, *SkipFinalClusterSnapshot* must be `false`.

Constraints:

- Must be 1 to 255 alphanumeric characters.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.

Return type dict

Returns

Response Syntax

```
{  
  'Cluster': {  
    'ClusterIdentifier': 'string',  
    'NodeType': 'string',  
    'ClusterStatus': 'string',  
    'ModifyStatus': 'string',  
    'MasterUsername': 'string',  
    'DBName': 'string',  
    'Endpoint': {  
      'Address': 'string',  
      'Port': 123  
    },  
    'ClusterCreateTime': datetime(2015, 1, 1),  
    'AutomatedSnapshotRetentionPeriod': 123,  
    'ClusterSecurityGroups': [  
      {  
        'ClusterSecurityGroupName': 'string',  
        'Status': 'string'  
      },  
    ],  
    'VpcSecurityGroups': [  
      {  

```



```

        'VpcSecurityGroupId': 'string',
        'Status': 'string'
    },
],
'ClusterParameterGroups': [
    {
        'ParameterGroupName': 'string',
        'ParameterApplyStatus': 'string',
        'ClusterParameterStatusList': [
            {
                'ParameterName': 'string',
                'ParameterApplyStatus': 'string',
                'ParameterApplyErrorDescription': 'string'
            },
        ]
    },
],
'ClusterSubnetGroupName': 'string',
'VpcId': 'string',
'AvailabilityZone': 'string',
'PreferredMaintenanceWindow': 'string',
'PendingModifiedValues': {
    'MasterUserPassword': 'string',
    'NodeType': 'string',
    'NumberOfNodes': 123,
    'ClusterType': 'string',
    'ClusterVersion': 'string',
    'AutomatedSnapshotRetentionPeriod': 123,
    'ClusterIdentifier': 'string'
},
'ClusterVersion': 'string',
'AllowVersionUpgrade': True|False,
'NumberOfNodes': 123,
'PubliclyAccessible': True|False,
'Encrypted': True|False,
'RestoreStatus': {
    'Status': 'string',
    'CurrentRestoreRateInMegaBytesPerSecond': 123.0,
    'SnapshotSizeInMegaBytes': 123,
    'ProgressInMegaBytes': 123,
    'ElapsedTimeInSeconds': 123,
    'EstimatedTimeToCompletionInSeconds': 123
},
'HsmStatus': {
    'HsmClientCertificateIdentifier': 'string',
    'HsmConfigurationIdentifier': 'string',
    'Status': 'string'
},
'ClusterSnapshotCopyStatus': {
    'DestinationRegion': 'string',
    'RetentionPeriod': 123,
    'SnapshotCopyGrantName': 'string'
},
'ClusterPublicKey': 'string',
'ClusterNodes': [
    {
        'NodeRole': 'string',
        'PrivateIpAddress': 'string',

```

```
        'PublicIPAddress': 'string'
    },
],
'ElasticIpStatus': {
    'ElasticIp': 'string',
    'Status': 'string'
},
'ClusterRevisionNumber': 'string',
'Tags': [
    {
        'Key': 'string',
        'Value': 'string'
    },
],
'KmsKeyId': 'string'
}
```

Response Structure

- (*dict*) –

Describes a cluster.

- **Cluster** (*dict*) –

Describes a cluster.

- * **ClusterIdentifier** (*string*) –

The unique identifier of the cluster.

- * **NodeType** (*string*) –

The node type for the nodes in the cluster.

- * **ClusterStatus** (*string*) –

The current state of this cluster. Possible values include `available`, `creating`, `deleting`, `rebooting`, `renaming`, and `resizing`.

- * **ModifyStatus** (*string*) –

The status of a modify operation, if any, initiated for the cluster.

- * **MasterUsername** (*string*) –

The master user name for the cluster. This name is used to connect to the database that is specified in **DBName**.

- * **DBName** (*string*) –

The name of the initial database that was created when the cluster was created. This same name is returned for the life of the cluster. If an initial database was not specified, a database named “dev” was created by default.

- * **Endpoint** (*dict*) –

The connection endpoint.

- **Address** (*string*) –

The DNS address of the Cluster.

- **Port** (*integer*) –

The port that the database engine is listening on.

- * **ClusterCreateTime** (*datetime*) –

The date and time that the cluster was created.

* **AutomatedSnapshotRetentionPeriod** (*integer*) –

The number of days that automatic cluster snapshots are retained.

* **ClusterSecurityGroups** (*list*) –

A list of cluster security group that are associated with the cluster. Each security group is represented by an element that contains `ClusterSecurityGroupName` and `ClusterSecurityGroup.Status` subelements.

Cluster security groups are used when the cluster is not created in a VPC. Clusters that are created in a VPC use VPC security groups, which are listed by the **VpcSecurityGroups** parameter.

· (*dict*) –

Describes a security group.

· **ClusterSecurityGroupName** (*string*) –

The name of the cluster security group.

· **Status** (*string*) –

The status of the cluster security group.

* **VpcSecurityGroups** (*list*) –

A list of Virtual Private Cloud (VPC) security groups that are associated with the cluster. This parameter is returned only if the cluster is in a VPC.

· (*dict*) –

Describes the members of a VPC security group.

· **VpcSecurityGroupId** (*string*) –

· **Status** (*string*) –

* **ClusterParameterGroups** (*list*) –

The list of cluster parameter groups that are associated with this cluster. Each parameter group in the list is returned with its status.

· (*dict*) –

Describes the status of a parameter group.

· **ParameterGroupName** (*string*) –

The name of the cluster parameter group.

· **ParameterApplyStatus** (*string*) –

The status of parameter updates.

· **ClusterParameterStatusList** (*list*) –

The list of parameter statuses.

For more information about parameters and parameter groups, go to [Amazon Redshift Parameter Groups](#) in the *Amazon Redshift Cluster Management Guide*.

· (*dict*) –

Describes the status of a parameter group.

· **ParameterName** (*string*) –

The name of the parameter.

· **ParameterApplyStatus** (*string*) –

The status of the parameter that indicates whether the parameter is in sync with the database, waiting for a cluster reboot,

or encountered an error when being applied.

The following are possible statuses and descriptions.

- `in-sync` : The parameter value is in sync with the database.
- `pending-reboot` : The parameter value will be applied after the cluster reboots.
- `applying` : The parameter value is being applied to the database.
- `invalid-parameter` : Cannot apply the parameter value because it has an invalid value or syntax.
- `apply-deferred` : The parameter contains static property changes. The changes are deferred until the cluster reboots.
- `apply-error` : Cannot connect to the cluster. The parameter change will be applied after the cluster reboots.
- `unknown-error` : Cannot apply the parameter change right now. The change will be applied after the cluster reboots.
- **ParameterApplyErrorDescription** (*string*) –

The error that prevented the parameter from being applied to the database.

- * **ClusterSubnetGroupName** (*string*) –

The name of the subnet group that is associated with the cluster. This parameter is valid only when the cluster is in a VPC.

- * **VpcId** (*string*) –

The identifier of the VPC the cluster is in, if the cluster is in a VPC.

- * **AvailabilityZone** (*string*) –

The name of the Availability Zone in which the cluster is located.

- * **PreferredMaintenanceWindow** (*string*) –

The weekly time range (in UTC) during which system maintenance can occur.

- * **PendingModifiedValues** (*dict*) –

If present, changes to the cluster are pending. Specific pending changes are identified by subelements.

- **MasterUserPassword** (*string*) –

The pending or in-progress change of the master user password for the cluster.

- **NodeType** (*string*) –

The pending or in-progress change of the cluster's node type.

- **NumberOfNodes** (*integer*) –

The pending or in-progress change of the number of nodes in the cluster.

- **ClusterType** (*string*) –

The pending or in-progress change of the cluster type.

- **ClusterVersion** (*string*) –

The pending or in-progress change of the service version.

- **AutomatedSnapshotRetentionPeriod** (*integer*) –

The pending or in-progress change of the automated snapshot retention period.

- **ClusterIdentifier** (*string*) –

The pending or in-progress change of the new identifier for the cluster.

- * **ClusterVersion** (*string*) –

The version ID of the Amazon Redshift engine that is running on the cluster.

- * **AllowVersionUpgrade** (*boolean*) –

If `true`, major version upgrades will be applied automatically to the cluster during the maintenance window.

- * **NumberOfNodes** (*integer*) –

The number of compute nodes in the cluster.

- * **PubliclyAccessible** (*boolean*) –

If `true`, the cluster can be accessed from a public network.

- * **Encrypted** (*boolean*) –

If `true`, data in the cluster is encrypted at rest.

- * **RestoreStatus** (*dict*) –

Describes the status of a cluster restore action. Returns null if the cluster was not created by restoring a snapshot.

- **Status** (*string*) –

The status of the restore action. Returns starting, restoring, completed, or failed.

- **CurrentRestoreRateInMegaBytesPerSecond** (*float*) –

The number of megabytes per second being transferred from the backup storage. Returns the average rate for a completed backup.

- **SnapshotSizeInMegaBytes** (*integer*) –

The size of the set of snapshot data used to restore the cluster.

- **ProgressInMegaBytes** (*integer*) –

The number of megabytes that have been transferred from snapshot storage.

- **ElapsedTimeInSeconds** (*integer*) –

The amount of time an in-progress restore has been running, or the amount of time it took a completed restore to finish.

- **EstimatedTimeToCompletionInSeconds** (*integer*) –

The estimate of the time remaining before the restore will complete. Returns 0 for a completed restore.

- * **HsmStatus** (*dict*) –

Reports whether the Amazon Redshift cluster has finished applying any HSM settings changes specified in a modify cluster command.

Values: active, applying

- **HsmClientCertificateIdentifier** (*string*) –

Specifies the name of the HSM client certificate the Amazon Redshift cluster uses to retrieve the data encryption keys stored in an HSM.

- **HsmConfigurationIdentifier** (*string*) –

Specifies the name of the HSM configuration that contains the information the Amazon Redshift cluster can use to retrieve and store keys in an HSM.

- **Status** (*string*) –

Reports whether the Amazon Redshift cluster has finished applying any HSM settings changes specified in a modify cluster command.

Values: active, applying

- * **ClusterSnapshotCopyStatus** (*dict*) –

Returns the destination region and retention period that are configured for cross-region snapshot copy.

- **DestinationRegion** (*string*) –

The destination region that snapshots are automatically copied to when cross-region snapshot copy is enabled.

- **RetentionPeriod** (*integer*) –

The number of days that automated snapshots are retained in the destination region after they are copied from a source region.

- **SnapshotCopyGrantName** (*string*) –

The name of the snapshot copy grant.

- * **ClusterPublicKey** (*string*) –

The public key for the cluster.

- * **ClusterNodes** (*list*) –

The nodes in a cluster.

- (*dict*) –

The identifier of a node in a cluster.

- **NodeRole** (*string*) –

Whether the node is a leader node or a compute node.

- **PrivateIPAddress** (*string*) –

The private IP address of a node within a cluster.

- **PublicIPAddress** (*string*) –

The public IP address of a node within a cluster.

- * **ElasticIpStatus** (*dict*) –

Describes the status of the elastic IP (EIP) address.

- **ElasticIp** (*string*) –

The elastic IP (EIP) address for the cluster.

- **Status** (*string*) –

Describes the status of the elastic IP (EIP) address.

- * **ClusterRevisionNumber** (*string*) –

The specific revision number of the database in the cluster.

- * **Tags** (*list*) –

The list of tags for the cluster.

- (*dict*) –

A tag consisting of a name/value pair for a resource.

- **Key** (*string*) –
The key, or name, for the resource tag.
- **Value** (*string*) –
The value for the resource tag.
- * **KmsKeyId** (*string*) –
The AWS Key Management Service (KMS) key ID of the encryption key used to encrypt data in the cluster.

delete_cluster_parameter_group (***kwargs*)
Deletes a specified Amazon Redshift parameter group.

Note: You cannot delete a parameter group if it is associated with a cluster.

Request Syntax

```
response = client.delete_cluster_parameter_group(
    ParameterGroupName='string'
)
```

Parameters **ParameterGroupName** (*string*) – [REQUIRED]

The name of the parameter group to be deleted.

Constraints:

- Must be the name of an existing cluster parameter group.
- Cannot delete a default cluster parameter group.

Returns None

delete_cluster_security_group (***kwargs*)
Deletes an Amazon Redshift security group.

Note: You cannot delete a security group that is associated with any clusters. You cannot delete the default security group.

For information about managing security groups, go to [Amazon Redshift Cluster Security Groups](#) in the *Amazon Redshift Cluster Management Guide*.

Request Syntax

```
response = client.delete_cluster_security_group(
    ClusterSecurityGroupName='string'
)
```

Parameters **ClusterSecurityGroupName** (*string*) – [REQUIRED]

The name of the cluster security group to be deleted.

Returns None

delete_cluster_snapshot (***kwargs*)
Deletes the specified manual snapshot. The snapshot must be in the `available` state, with no other users authorized to access the snapshot.

Unlike automated snapshots, manual snapshots are retained even after you delete your cluster. Amazon Redshift does not delete your manual snapshots. You must delete manual snapshot explicitly to avoid getting charged. If other accounts are authorized to access the snapshot, you must revoke all of the authorizations before you can delete the snapshot.

Request Syntax

```
response = client.delete_cluster_snapshot(  
    SnapshotIdentifier='string',  
    SnapshotClusterIdentifier='string'  
)
```

Parameters

- **SnapshotIdentifier** (*string*) – **[REQUIRED]**

The unique identifier of the manual snapshot to be deleted.

Constraints: Must be the name of an existing snapshot that is in the available state.

- **SnapshotClusterIdentifier** (*string*) – The unique identifier of the cluster the snapshot was created from. This parameter is required if your IAM user has a policy containing a snapshot resource element that specifies anything other than * for the cluster name.

Constraints: Must be the name of valid cluster.

Return type dict

Returns

Response Syntax

```
{  
    'Snapshot': {  
        'SnapshotIdentifier': 'string',  
        'ClusterIdentifier': 'string',  
        'SnapshotCreateTime': datetime(2015, 1, 1),  
        'Status': 'string',  
        'Port': 123,  
        'AvailabilityZone': 'string',  
        'ClusterCreateTime': datetime(2015, 1, 1),  
        'MasterUsername': 'string',  
        'ClusterVersion': 'string',  
        'SnapshotType': 'string',  
        'NodeType': 'string',  
        'NumberOfNodes': 123,  
        'DBName': 'string',  
        'VpcId': 'string',  
        'Encrypted': True|False,  
        'KmsKeyId': 'string',  
        'EncryptedWithHSM': True|False,  
        'AccountsWithRestoreAccess': [  
            {  
                'AccountId': 'string'  
            },  
        ],  
        'OwnerAccount': 'string',  
        'TotalBackupSizeInMegabytes': 123.0,  
        'ActualIncrementalBackupSizeInMegabytes': 123.0,  
        'BackupProgressInMegabytes': 123.0,  
        'CurrentBackupRateInMegabytesPerSecond': 123.0,  
        'EstimatedSecondsToCompletion': 123,  
        'ElapsedTimeInSeconds': 123,  
        'SourceRegion': 'string',  
        'Tags': [  
            {  
                'Key': 'string',  
                'Value': 'string'  
            },  
        ]  
    }  
}
```



```

        'Key': 'string',
        'Value': 'string'
    },
],
'RestorableNodeTypes': [
    'string',
]
}
}

```

Response Structure

- *(dict)* –

Describes a snapshot.

- **Snapshot** (*dict*) –

Describes a snapshot.

- * **SnapshotIdentifier** (*string*) –

The snapshot identifier that is provided in the request.

- * **ClusterIdentifier** (*string*) –

The identifier of the cluster for which the snapshot was taken.

- * **SnapshotCreateTime** (*datetime*) –

The time (UTC) when Amazon Redshift began the snapshot. A snapshot contains a copy of the cluster data as of this exact time.

- * **Status** (*string*) –

The snapshot status. The value of the status depends on the API operation used.

- CreateClusterSnapshot and CopyClusterSnapshot returns status as “creating”.
- DescribeClusterSnapshots returns status as “creating”, “available”, “final snapshot”, or “failed”.
- DeleteClusterSnapshot returns status as “deleted”.

- * **Port** (*integer*) –

The port that the cluster is listening on.

- * **AvailabilityZone** (*string*) –

The Availability Zone in which the cluster was created.

- * **ClusterCreateTime** (*datetime*) –

The time (UTC) when the cluster was originally created.

- * **MasterUsername** (*string*) –

The master user name for the cluster.

- * **ClusterVersion** (*string*) –

The version ID of the Amazon Redshift engine that is running on the cluster.

- * **SnapshotType** (*string*) –

The snapshot type. Snapshots created using CreateClusterSnapshot and CopyClusterSnapshot will be of type “manual”.

- * **NodeType** (*string*) –

The node type of the nodes in the cluster.

- * **NumberOfNodes** (*integer*) –

The number of nodes in the cluster.

* **DBName** (*string*) –

The name of the database that was created when the cluster was created.

* **VpcId** (*string*) –

The VPC identifier of the cluster if the snapshot is from a cluster in a VPC. Otherwise, this field is not in the output.

* **Encrypted** (*boolean*) –

If `true`, the data in the snapshot is encrypted at rest.

* **KmsKeyId** (*string*) –

The AWS Key Management Service (KMS) key ID of the encryption key that was used to encrypt data in the cluster from which the snapshot was taken.

* **EncryptedWithHSM** (*boolean*) –

A boolean that indicates whether the snapshot data is encrypted using the HSM keys of the source cluster. `true` indicates that the data is encrypted using HSM keys.

* **AccountsWithRestoreAccess** (*list*) –

A list of the AWS customer accounts authorized to restore the snapshot. Returns `null` if no accounts are authorized. Visible only to the snapshot owner.

· (*dict*) –

Describes an AWS customer account authorized to restore a snapshot.

· **AccountId** (*string*) –

The identifier of an AWS customer account authorized to restore a snapshot.

* **OwnerAccount** (*string*) –

For manual snapshots, the AWS customer account used to create or copy the snapshot. For automatic snapshots, the owner of the cluster. The owner can perform all snapshot actions, such as sharing a manual snapshot.

* **TotalBackupSizeInMegaBytes** (*float*) –

The size of the complete set of backup data that would be used to restore the cluster.

* **ActualIncrementalBackupSizeInMegaBytes** (*float*) –

The size of the incremental backup.

* **BackupProgressInMegaBytes** (*float*) –

The number of megabytes that have been transferred to the snapshot backup.

* **CurrentBackupRateInMegaBytesPerSecond** (*float*) –

The number of megabytes per second being transferred to the snapshot backup. Returns 0 for a completed backup.

* **EstimatedSecondsToCompletion** (*integer*) –

The estimate of the time remaining before the snapshot backup will complete. Returns 0 for a completed backup.

* **ElapsedTimeInSeconds** (*integer*) –

The amount of time an in-progress snapshot backup has been running, or the amount of time it took a completed backup to finish.

* **SourceRegion** (*string*) –

The source region from which the snapshot was copied.

* **Tags** (*list*) –

The list of tags for the cluster snapshot.

· (*dict*) –

A tag consisting of a name/value pair for a resource.

· **Key** (*string*) –

The key, or name, for the resource tag.

· **Value** (*string*) –

The value for the resource tag.

* **RestorableNodeTypes** (*list*) –

The list of node types that this cluster snapshot is able to restore into.

· (*string*) –

delete_cluster_subnet_group (**kwargs)

Deletes the specified cluster subnet group.

Request Syntax

```
response = client.delete_cluster_subnet_group(
    ClusterSubnetGroupName='string'
)
```

Parameters **ClusterSubnetGroupName** (*string*) – [REQUIRED]

The name of the cluster subnet group name to be deleted.

Returns None

delete_event_subscription (**kwargs)

Deletes an Amazon Redshift event notification subscription.

Request Syntax

```
response = client.delete_event_subscription(
    SubscriptionName='string'
)
```

Parameters **SubscriptionName** (*string*) – [REQUIRED]

The name of the Amazon Redshift event notification subscription to be deleted.

Returns None

delete_hsm_client_certificate (**kwargs)

Deletes the specified HSM client certificate.

Request Syntax

```
response = client.delete_hsm_client_certificate(
    HsmClientCertificateIdentifier='string'
)
```

Parameters `HsmClientCertificateIdentifier` (*string*) – [REQUIRED]

The identifier of the HSM client certificate to be deleted.

Returns None

delete_hsm_configuration (***kwargs*)

Deletes the specified Amazon Redshift HSM configuration.

Request Syntax

```
response = client.delete_hsm_configuration(
    HsmConfigurationIdentifier='string'
)
```

Parameters `HsmConfigurationIdentifier` (*string*) – [REQUIRED]

The identifier of the Amazon Redshift HSM configuration to be deleted.

Returns None

delete_snapshot_copy_grant (***kwargs*)

Deletes the specified snapshot copy grant.

Request Syntax

```
response = client.delete_snapshot_copy_grant(
    SnapshotCopyGrantName='string'
)
```

Parameters `SnapshotCopyGrantName` (*string*) – [REQUIRED]

The name of the snapshot copy grant to delete.

Returns None

delete_tags (***kwargs*)

Deletes a tag or tags from a resource. You must provide the ARN of the resource from which you want to delete the tag or tags.

Request Syntax

```
response = client.delete_tags(
    ResourceName='string',
    TagKeys=[
        'string',
    ]
)
```

Parameters

- **ResourceName** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) from which you want to remove the tag or tags. For example, `arn:aws:redshift:us-east-1:123456789:cluster:t1`.

- **TagKeys** (*list*) – [REQUIRED]

The tag key that you want to delete.

– (*string*) –

Returns None

describe_cluster_parameter_groups (***kwargs*)

Returns a list of Amazon Redshift parameter groups, including parameter groups you created and the

default parameter group. For each parameter group, the response includes the parameter group name, description, and parameter group family name. You can optionally specify a name to retrieve the description of a specific parameter group.

For more information about parameters and parameter groups, go to [Amazon Redshift Parameter Groups](#) in the *Amazon Redshift Cluster Management Guide*.

If you specify both tag keys and tag values in the same request, Amazon Redshift returns all parameter groups that match any combination of the specified keys and values. For example, if you have `owner` and `environment` for tag keys, and `admin` and `test` for tag values, all parameter groups that have any combination of those values are returned.

If both tag keys and values are omitted from the request, parameter groups are returned regardless of whether they have tag keys or values associated with them.

Request Syntax

```
response = client.describe_cluster_parameter_groups(
    ParameterGroupName='string',
    MaxRecords=123,
    Marker='string',
    TagKeys=[
        'string',
    ],
    TagValues=[
        'string',
    ]
)
```

Parameters

- **ParameterGroupName** (*string*) – The name of a specific parameter group for which to return details. By default, details about all parameter groups and the default parameter group are returned.
- **MaxRecords** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified `MaxRecords` value, a value is returned in a `marker` field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: 100

Constraints: minimum 20, maximum 100.

- **Marker** (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeClusterParameterGroups` request exceed the value specified in `MaxRecords`, AWS returns a value in the `Marker` field of the response. You can retrieve the next set of response records by providing the returned marker value in the `Marker` parameter and retrying the request.
- **TagKeys** (*list*) – A tag key or keys for which you want to return all matching cluster parameter groups that are associated with the specified key or keys. For example, suppose that you have parameter groups that are tagged with keys called `owner` and `environment`. If you specify both of these tag keys in the request, Amazon Redshift returns a response with the parameter groups that have either or both of these tag keys associated with them.
 - (*string*) –
- **TagValues** (*list*) – A tag value or values for which you want to return all matching cluster parameter groups that are associated with the specified tag value or values. For example, suppose that you have parameter groups that are tagged

with values called `admin` and `test` . If you specify both of these tag values in the request, Amazon Redshift returns a response with the parameter groups that have either or both of these tag values associated with them.

– (string) –

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'ParameterGroups': [
    {
      'ParameterGroupName': 'string',
      'ParameterGroupFamily': 'string',
      'Description': 'string',
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        },
        ...
      ]
    },
    ...
  ]
}
```

Response Structure

• (dict) –

Contains the output from the `DescribeClusterParameterGroups` action.

– **Marker** (string) –

A value that indicates the starting point for the next set of response records in a subsequent request. If a value is returned in a response, you can retrieve the next set of records by providing this returned marker value in the `Marker` parameter and retrying the command. If the `Marker` field is empty, all response records have been retrieved for the request.

– **ParameterGroups** (list) –

A list of `ClusterParameterGroup` instances. Each instance describes one cluster parameter group.

* (dict) –

Describes a parameter group.

· **ParameterGroupName** (string) –

The name of the cluster parameter group.

· **ParameterGroupFamily** (string) –

The name of the cluster parameter group family that this cluster parameter group is compatible with.

· **Description** (string) –

The description of the parameter group.

· **Tags** (list) –

The list of tags for the cluster parameter group.

· (dict) –

A tag consisting of a name/value pair for a resource.

- **Key** (*string*) –
The key, or name, for the resource tag.
- **Value** (*string*) –
The value for the resource tag.

describe_cluster_parameters (***kwargs*)

Returns a detailed list of parameters contained within the specified Amazon Redshift parameter group. For each parameter the response includes information such as parameter name, description, data type, value, whether the parameter value is modifiable, and so on.

You can specify *source* filter to retrieve parameters of only specific type. For example, to retrieve parameters that were modified by a user action such as from `ModifyClusterParameterGroup`, you can specify *source* equal to *user*.

For more information about parameters and parameter groups, go to [Amazon Redshift Parameter Groups](#) in the *Amazon Redshift Cluster Management Guide*.

Request Syntax

```
response = client.describe_cluster_parameters(
    ParameterGroupName='string',
    Source='string',
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **ParameterGroupName** (*string*) – **[REQUIRED]**
The name of a cluster parameter group for which to return details.
- **Source** (*string*) – The parameter types to return. Specify *user* to show parameters that are different from the default. Similarly, specify *engine-default* to show parameters that are the same as the default parameter group.

Default: All parameter types returned.

Valid Values: *user* | *engine-default*
- **MaxRecords** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified *MaxRecords* value, a value is returned in a *marker* field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: 100

Constraints: minimum 20, maximum 100.
- **Marker** (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeClusterParameters` request exceed the value specified in *MaxRecords*, AWS returns a value in the *Marker* field of the response. You can retrieve the next set of response records by providing the returned marker value in the *Marker* parameter and retrying the request.

Return type dict

Returns

Response Syntax

```
{
  'Parameters': [
    {
      'ParameterName': 'string',
      'ParameterValue': 'string',
      'Description': 'string',
      'Source': 'string',
      'DataType': 'string',
      'AllowedValues': 'string',
      'ApplyType': 'static'|'dynamic',
      'IsModifiable': True|False,
      'MinimumEngineVersion': 'string'
    },
  ],
  'Marker': 'string'
}
```

Response Structure

- *(dict)* –

Contains the output from the DescribeClusterParameters action.

- **Parameters** (*list*) –

A list of Parameter instances. Each instance lists the parameters of one cluster parameter group.

- * *(dict)* –

Describes a parameter in a cluster parameter group.

- **ParameterName** (*string*) –

The name of the parameter.

- **ParameterValue** (*string*) –

The value of the parameter.

- **Description** (*string*) –

A description of the parameter.

- **Source** (*string*) –

The source of the parameter value, such as “engine-default” or “user”.

- **DataType** (*string*) –

The data type of the parameter.

- **AllowedValues** (*string*) –

The valid range of values for the parameter.

- **ApplyType** (*string*) –

Specifies how to apply the parameter. Supported value: `static`.

- **IsModifiable** (*boolean*) –

If `true`, the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.

- **MinimumEngineVersion** (*string*) –

The earliest engine version to which the parameter can apply.

- **Marker** (*string*) –

A value that indicates the starting point for the next set of response records in a subsequent request. If a value is returned in a response, you can retrieve the next set of records by providing this returned marker value in the `Marker` parameter and retrying the command. If the `Marker` field is empty, all response records have been retrieved for the request.

describe_cluster_security_groups (***kwargs*)

Returns information about Amazon Redshift security groups. If the name of a security group is specified, the response will contain only information about only that security group.

For information about managing security groups, go to [Amazon Redshift Cluster Security Groups](#) in the *Amazon Redshift Cluster Management Guide*.

If you specify both tag keys and tag values in the same request, Amazon Redshift returns all security groups that match any combination of the specified keys and values. For example, if you have `owner` and `environment` for tag keys, and `admin` and `test` for tag values, all security groups that have any combination of those values are returned.

If both tag keys and values are omitted from the request, security groups are returned regardless of whether they have tag keys or values associated with them.

Request Syntax

```
response = client.describe_cluster_security_groups(
    ClusterSecurityGroupName='string',
    MaxRecords=123,
    Marker='string',
    TagKeys=[
        'string',
    ],
    TagValues=[
        'string',
    ]
)
```

Parameters

- **ClusterSecurityGroupName** (*string*) – The name of a cluster security group for which you are requesting details. You can specify either the **Marker** parameter or a **ClusterSecurityGroupName** parameter, but not both.

Example: `securitygroup1`

- **MaxRecords** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified `MaxRecords` value, a value is returned in a `marker` field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: 100

Constraints: minimum 20, maximum 100.

- **Marker** (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeClusterSecurityGroups` request exceed the value specified in `MaxRecords`, AWS returns a value in the `Marker` field of the response. You can retrieve the next set of response records by providing the returned marker value in the `Marker` parameter and retrying the request.

Constraints: You can specify either the **ClusterSecurityGroupName** parameter or the **Marker** parameter, but not both.

- **TagKeys** (*list*) – A tag key or keys for which you want to return all matching cluster security groups that are associated with the specified key or keys. For example, suppose that you have security groups that are tagged with keys called `owner` and `environment`. If you specify both of these tag keys in the request, Amazon Redshift returns a response with the security groups that have either or both of these tag keys associated with them.
 - (*string*) –
- **TagValues** (*list*) – A tag value or values for which you want to return all matching cluster security groups that are associated with the specified tag value or values. For example, suppose that you have security groups that are tagged with values called `admin` and `test`. If you specify both of these tag values in the request, Amazon Redshift returns a response with the security groups that have either or both of these tag values associated with them.
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'ClusterSecurityGroups': [
    {
      'ClusterSecurityGroupName': 'string',
      'Description': 'string',
      'EC2SecurityGroups': [
        {
          'Status': 'string',
          'EC2SecurityGroupName': 'string',
          'EC2SecurityGroupOwnerId': 'string',
          'Tags': [
            {
              'Key': 'string',
              'Value': 'string'
            }
          ],
        }
      ],
    },
  ],
  'IPRanges': [
    {
      'Status': 'string',
      'CIDRIP': 'string',
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        }
      ],
    },
  ],
  'Tags': [
    {
      'Key': 'string',
      'Value': 'string'
    },
  ],
}
```

```
]
}
```

Response Structure

- *(dict)* –

Contains the output from the DescribeClusterSecurityGroups action.

- **Marker** (*string*) –

A value that indicates the starting point for the next set of response records in a subsequent request. If a value is returned in a response, you can retrieve the next set of records by providing this returned marker value in the `Marker` parameter and retrying the command. If the `Marker` field is empty, all response records have been retrieved for the request.

- **ClusterSecurityGroups** (*list*) –

A list of ClusterSecurityGroup instances.

- * *(dict)* –

Describes a security group.

- **ClusterSecurityGroupName** (*string*) –

The name of the cluster security group to which the operation was applied.

- **Description** (*string*) –

A description of the security group.

- **EC2SecurityGroups** (*list*) –

A list of EC2 security groups that are permitted to access clusters associated with this cluster security group.

- *(dict)* –

Describes an Amazon EC2 security group.

- **Status** (*string*) –

The status of the EC2 security group.

- **EC2SecurityGroupName** (*string*) –

The name of the EC2 Security Group.

- **EC2SecurityGroupOwnerId** (*string*) –

The AWS ID of the owner of the EC2 security group specified in the `EC2SecurityGroupName` field.

- **Tags** (*list*) –

The list of tags for the EC2 security group.

- *(dict)* –

A tag consisting of a name/value pair for a resource.

- **Key** (*string*) –

The key, or name, for the resource tag.

- **Value** (*string*) –

The value for the resource tag.

- **IPRanges** (*list*) –

A list of IP ranges (CIDR blocks) that are permitted to access clusters associated with this cluster security group.

- *(dict)* –
Describes an IP range used in a security group.
- **Status** (*string*) –
The status of the IP range, for example, “authorized”.
- **CIDRIP** (*string*) –
The IP range in Classless Inter-Domain Routing (CIDR) notation.
- **Tags** (*list*) –
The list of tags for the IP range.
- *(dict)* –
A tag consisting of a name/value pair for a resource.
- **Key** (*string*) –
The key, or name, for the resource tag.
- **Value** (*string*) –
The value for the resource tag.
- **Tags** (*list*) –
The list of tags for the cluster security group.
- *(dict)* –
A tag consisting of a name/value pair for a resource.
- **Key** (*string*) –
The key, or name, for the resource tag.
- **Value** (*string*) –
The value for the resource tag.

describe_cluster_snapshots (***kwargs*)

Returns one or more snapshot objects, which contain metadata about your cluster snapshots. By default, this operation returns information about all snapshots of all clusters that are owned by you AWS customer account. No information is returned for snapshots owned by inactive AWS customer accounts.

If you specify both tag keys and tag values in the same request, Amazon Redshift returns all snapshots that match any combination of the specified keys and values. For example, if you have `owner` and `environment` for tag keys, and `admin` and `test` for tag values, all snapshots that have any combination of those values are returned. Only snapshots that you own are returned in the response; shared snapshots are not returned with the tag key and tag value request parameters.

If both tag keys and values are omitted from the request, snapshots are returned regardless of whether they have tag keys or values associated with them.

Request Syntax

```
response = client.describe_cluster_snapshots(
    ClusterIdentifier='string',
    SnapshotIdentifier='string',
    SnapshotType='string',
    StartTime=datetime(2015, 1, 1),
    EndTime=datetime(2015, 1, 1),
    MaxRecords=123,
    Marker='string',
    OwnerAccount='string',
    TagKeys=[
        'string',
```

```

],
TagValues=[
    'string',
]
)

```

Parameters

- **ClusterIdentifier** (*string*) – The identifier of the cluster for which information about snapshots is requested.
- **SnapshotIdentifier** (*string*) – The snapshot identifier of the snapshot about which to return information.
- **SnapshotType** (*string*) – The type of snapshots for which you are requesting information. By default, snapshots of all types are returned.

Valid Values: `automated|manual`

- **StartTime** (*datetime*) – A value that requests only snapshots created at or after the specified time. The time value is specified in ISO 8601 format. For more information about ISO 8601, go to the [ISO8601 Wikipedia page](#).

Example: `2012-07-16T18:00:00Z`

- **EndTime** (*datetime*) – A time value that requests only snapshots created at or before the specified time. The time value is specified in ISO 8601 format. For more information about ISO 8601, go to the [ISO8601 Wikipedia page](#).

Example: `2012-07-16T18:00:00Z`

- **MaxRecords** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified `MaxRecords` value, a value is returned in a `marker` field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: `100`

Constraints: minimum `20`, maximum `100`.

- **Marker** (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeClusterSnapshots` request exceed the value specified in `MaxRecords`, AWS returns a value in the `Marker` field of the response. You can retrieve the next set of response records by providing the returned marker value in the `Marker` parameter and retrying the request.
- **OwnerAccount** (*string*) – The AWS customer account used to create or copy the snapshot. Use this field to filter the results to snapshots owned by a particular account. To describe snapshots you own, either specify your AWS customer account, or do not specify the parameter.
- **TagKeys** (*list*) – A tag key or keys for which you want to return all matching cluster snapshots that are associated with the specified key or keys. For example, suppose that you have snapshots that are tagged with keys called `owner` and `environment`. If you specify both of these tag keys in the request, Amazon Redshift returns a response with the snapshots that have either or both of these tag keys associated with them.
 - (*string*) –
- **TagValues** (*list*) – A tag value or values for which you want to return all matching cluster snapshots that are associated with the specified tag value or values. For example, suppose that you have snapshots that are tagged with values called `admin` and `test`. If you specify both of these tag values in the request, Amazon Redshift returns a response with the snapshots that have either or both of these tag values associated with them.

– (string) –

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'Snapshots': [
    {
      'SnapshotIdentifier': 'string',
      'ClusterIdentifier': 'string',
      'SnapshotCreateTime': datetime(2015, 1, 1),
      'Status': 'string',
      'Port': 123,
      'AvailabilityZone': 'string',
      'ClusterCreateTime': datetime(2015, 1, 1),
      'MasterUsername': 'string',
      'ClusterVersion': 'string',
      'SnapshotType': 'string',
      'NodeType': 'string',
      'NumberOfNodes': 123,
      'DBName': 'string',
      'VpcId': 'string',
      'Encrypted': True|False,
      'KmsKeyId': 'string',
      'EncryptedWithHSM': True|False,
      'AccountsWithRestoreAccess': [
        {
          'AccountId': 'string'
        },
      ],
      'OwnerAccount': 'string',
      'TotalBackupSizeInMegabytes': 123.0,
      'ActualIncrementalBackupSizeInMegabytes': 123.0,
      'BackupProgressInMegabytes': 123.0,
      'CurrentBackupRateInMegabytesPerSecond': 123.0,
      'EstimatedSecondsToCompletion': 123,
      'ElapsedTimeInSeconds': 123,
      'SourceRegion': 'string',
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        },
      ],
      'RestorableNodeTypes': [
        'string',
      ]
    },
  ]
}
```

Response Structure

- (dict) –

Contains the output from the DescribeClusterSnapshots action.

– **Marker** (string) –

A value that indicates the starting point for the next set of response records in a subsequent request. If a value is returned in a response, you can retrieve the next set of records by providing this returned marker value in the `Marker` parameter and retrying the command. If the `Marker` field is empty, all response records have been retrieved for the request.

– **Snapshots** (*list*) –

A list of Snapshot instances.

* (*dict*) –

Describes a snapshot.

- **SnapshotIdentifier** (*string*) –

The snapshot identifier that is provided in the request.

- **ClusterIdentifier** (*string*) –

The identifier of the cluster for which the snapshot was taken.

- **SnapshotCreateTime** (*datetime*) –

The time (UTC) when Amazon Redshift began the snapshot. A snapshot contains a copy of the cluster data as of this exact time.

- **Status** (*string*) –

The snapshot status. The value of the status depends on the API operation used.

- `CreateClusterSnapshot` and `CopyClusterSnapshot` returns status as “creating”.

- `DescribeClusterSnapshots` returns status as “creating”, “available”, “final snapshot”, or “failed”.

- `DeleteClusterSnapshot` returns status as “deleted”.

- **Port** (*integer*) –

The port that the cluster is listening on.

- **AvailabilityZone** (*string*) –

The Availability Zone in which the cluster was created.

- **ClusterCreateTime** (*datetime*) –

The time (UTC) when the cluster was originally created.

- **MasterUsername** (*string*) –

The master user name for the cluster.

- **ClusterVersion** (*string*) –

The version ID of the Amazon Redshift engine that is running on the cluster.

- **SnapshotType** (*string*) –

The snapshot type. Snapshots created using `CreateClusterSnapshot` and `CopyClusterSnapshot` will be of type “manual”.

- **NodeType** (*string*) –

The node type of the nodes in the cluster.

- **NumberOfNodes** (*integer*) –

The number of nodes in the cluster.

- **DBName** (*string*) –

The name of the database that was created when the cluster was created.

- **VpcId** (*string*) –
The VPC identifier of the cluster if the snapshot is from a cluster in a VPC. Otherwise, this field is not in the output.
- **Encrypted** (*boolean*) –
If `true`, the data in the snapshot is encrypted at rest.
- **KmsKeyId** (*string*) –
The AWS Key Management Service (KMS) key ID of the encryption key that was used to encrypt data in the cluster from which the snapshot was taken.
- **EncryptedWithHSM** (*boolean*) –
A boolean that indicates whether the snapshot data is encrypted using the HSM keys of the source cluster. `true` indicates that the data is encrypted using HSM keys.
- **AccountsWithRestoreAccess** (*list*) –
A list of the AWS customer accounts authorized to restore the snapshot. Returns `null` if no accounts are authorized. Visible only to the snapshot owner.
- (*dict*) –
Describes an AWS customer account authorized to restore a snapshot.
- **AccountId** (*string*) –
The identifier of an AWS customer account authorized to restore a snapshot.
- **OwnerAccount** (*string*) –
For manual snapshots, the AWS customer account used to create or copy the snapshot. For automatic snapshots, the owner of the cluster. The owner can perform all snapshot actions, such as sharing a manual snapshot.
- **TotalBackupSizeInMegaBytes** (*float*) –
The size of the complete set of backup data that would be used to restore the cluster.
- **ActualIncrementalBackupSizeInMegaBytes** (*float*) –
The size of the incremental backup.
- **BackupProgressInMegaBytes** (*float*) –
The number of megabytes that have been transferred to the snapshot backup.
- **CurrentBackupRateInMegaBytesPerSecond** (*float*) –
The number of megabytes per second being transferred to the snapshot backup. Returns 0 for a completed backup.
- **EstimatedSecondsToCompletion** (*integer*) –
The estimate of the time remaining before the snapshot backup will complete. Returns 0 for a completed backup.
- **ElapsedTimeInSeconds** (*integer*) –
The amount of time an in-progress snapshot backup has been running, or the amount of time it took a completed backup to finish.

- **SourceRegion** (*string*) –
The source region from which the snapshot was copied.
- **Tags** (*list*) –
The list of tags for the cluster snapshot.
- (*dict*) –
A tag consisting of a name/value pair for a resource.
- **Key** (*string*) –
The key, or name, for the resource tag.
- **Value** (*string*) –
The value for the resource tag.
- **RestorableNodeTypes** (*list*) –
The list of node types that this cluster snapshot is able to restore into.
- (*string*) –

describe_cluster_subnet_groups (***kwargs*)

Returns one or more cluster subnet group objects, which contain metadata about your cluster subnet groups. By default, this operation returns information about all cluster subnet groups that are defined in your AWS account.

If you specify both tag keys and tag values in the same request, Amazon Redshift returns all subnet groups that match any combination of the specified keys and values. For example, if you have `owner` and `environment` for tag keys, and `admin` and `test` for tag values, all subnet groups that have any combination of those values are returned.

If both tag keys and values are omitted from the request, subnet groups are returned regardless of whether they have tag keys or values associated with them.

Request Syntax

```
response = client.describe_cluster_subnet_groups(
    ClusterSubnetGroupName='string',
    MaxRecords=123,
    Marker='string',
    TagKeys=[
        'string',
    ],
    TagValues=[
        'string',
    ]
)
```

Parameters

- **ClusterSubnetGroupName** (*string*) – The name of the cluster subnet group for which information is requested.
- **MaxRecords** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified `MaxRecords` value, a value is returned in a `marker` field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: 100

Constraints: minimum 20, maximum 100.

- **Marker** (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeClusterSubnetGroups` request exceed the value specified in `MaxRecords`, AWS returns a value in the `Marker` field of the response. You can retrieve the next set of response records by providing the returned marker value in the `Marker` parameter and retrying the request.
- **TagKeys** (*list*) – A tag key or keys for which you want to return all matching cluster subnet groups that are associated with the specified key or keys. For example, suppose that you have subnet groups that are tagged with keys called `owner` and `environment`. If you specify both of these tag keys in the request, Amazon Redshift returns a response with the subnet groups that have either or both of these tag keys associated with them.
 - (*string*) –
- **TagValues** (*list*) – A tag value or values for which you want to return all matching cluster subnet groups that are associated with the specified tag value or values. For example, suppose that you have subnet groups that are tagged with values called `admin` and `test`. If you specify both of these tag values in the request, Amazon Redshift returns a response with the subnet groups that have either or both of these tag values associated with them.
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'ClusterSubnetGroups': [
    {
      'ClusterSubnetGroupName': 'string',
      'Description': 'string',
      'VpcId': 'string',
      'SubnetGroupStatus': 'string',
      'Subnets': [
        {
          'SubnetIdentifier': 'string',
          'SubnetAvailabilityZone': {
            'Name': 'string'
          },
          'SubnetStatus': 'string'
        },
        ...
      ],
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        },
        ...
      ]
    },
    ...
  ]
}
```

Response Structure

- (*dict*) –
 - Contains the output from the `DescribeClusterSubnetGroups` action.
 - **Marker** (*string*) –

A value that indicates the starting point for the next set of response records in a subsequent request. If a value is returned in a response, you can retrieve the next set of records by providing this returned marker value in the `Marker` parameter and retrying the command. If the `Marker` field is empty, all response records have been retrieved for the request.

– **ClusterSubnetGroups** (*list*) –

A list of ClusterSubnetGroup instances.

* (*dict*) –

Describes a subnet group.

· **ClusterSubnetGroupName** (*string*) –

The name of the cluster subnet group.

· **Description** (*string*) –

The description of the cluster subnet group.

· **VpcId** (*string*) –

The VPC ID of the cluster subnet group.

· **SubnetGroupStatus** (*string*) –

The status of the cluster subnet group. Possible values are `Complete`, `Incomplete` and `Invalid`.

· **Subnets** (*list*) –

A list of the VPC Subnet elements.

· (*dict*) –

Describes a subnet.

· **SubnetIdentifier** (*string*) –

The identifier of the subnet.

· **SubnetAvailabilityZone** (*dict*) –

Describes an availability zone.

· **Name** (*string*) –

The name of the availability zone.

· **SubnetStatus** (*string*) –

The status of the subnet.

· **Tags** (*list*) –

The list of tags for the cluster subnet group.

· (*dict*) –

A tag consisting of a name/value pair for a resource.

· **Key** (*string*) –

The key, or name, for the resource tag.

· **Value** (*string*) –

The value for the resource tag.

describe_cluster_versions (***kwargs*)

Returns descriptions of the available Amazon Redshift cluster versions. You can call this operation even before creating any clusters to learn more about the Amazon Redshift versions. For more information about managing clusters, go to [Amazon Redshift Clusters](#) in the *Amazon Redshift Cluster Management Guide*

Request Syntax

```
response = client.describe_cluster_versions(
    ClusterVersion='string',
    ClusterParameterGroupFamily='string',
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **ClusterVersion** (*string*) – The specific cluster version to return.

Example: 1.0

- **ClusterParameterGroupFamily** (*string*) – The name of a specific cluster parameter group family to return details for.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **MaxRecords** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified `MaxRecords` value, a value is returned in a `marker` field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: 100

Constraints: minimum 20, maximum 100.

- **Marker** (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeClusterVersions` request exceed the value specified in `MaxRecords`, AWS returns a value in the `Marker` field of the response. You can retrieve the next set of response records by providing the returned marker value in the `Marker` parameter and retrying the request.

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'ClusterVersions': [
    {
      'ClusterVersion': 'string',
      'ClusterParameterGroupFamily': 'string',
      'Description': 'string'
    },
  ],
}
```

Response Structure

- (*dict*) –

Contains the output from the `DescribeClusterVersions` action.

- **Marker** (*string*) –

A value that indicates the starting point for the next set of response records in a subsequent request. If a value is returned in a response, you can retrieve the next set of records by providing this returned marker value in

the `Marker` parameter and retrying the command. If the `Marker` field is empty, all response records have been retrieved for the request.

– **ClusterVersions** (*list*) –

A list of `Version` elements.

* (*dict*) –

Describes a cluster version, including the parameter group family and description of the version.

· **ClusterVersion** (*string*) –

The version number used by the cluster.

· **ClusterParameterGroupFamily** (*string*) –

The name of the cluster parameter group family for the cluster.

· **Description** (*string*) –

The description of the cluster version.

describe_clusters (***kwargs*)

Returns properties of provisioned clusters including general cluster properties, cluster database properties, maintenance and backup properties, and security and access properties. This operation supports pagination. For more information about managing clusters, go to [Amazon Redshift Clusters](#) in the *Amazon Redshift Cluster Management Guide*.

If you specify both tag keys and tag values in the same request, Amazon Redshift returns all clusters that match any combination of the specified keys and values. For example, if you have `owner` and `environment` for tag keys, and `admin` and `test` for tag values, all clusters that have any combination of those values are returned.

If both tag keys and values are omitted from the request, clusters are returned regardless of whether they have tag keys or values associated with them.

Request Syntax

```
response = client.describe_clusters(
    ClusterIdentifier='string',
    MaxRecords=123,
    Marker='string',
    TagKeys=[
        'string',
    ],
    TagValues=[
        'string',
    ]
)
```

Parameters

- **ClusterIdentifier** (*string*) – The unique identifier of a cluster whose properties you are requesting. This parameter is case sensitive.

The default is that all clusters defined for an account are returned.

- **MaxRecords** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified `MaxRecords` value, a value is returned in a `marker` field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: 100

Constraints: minimum 20, maximum 100.

- **Marker** (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeClusters` request exceed the value specified in `MaxRecords`, AWS returns a value in the `Marker` field of the response. You can retrieve the next set of response records by providing the returned marker value in the `Marker` parameter and retrying the request.

Constraints: You can specify either the **ClusterIdentifier** parameter or the **Marker** parameter, but not both.

- **TagKeys** (*list*) – A tag key or keys for which you want to return all matching clusters that are associated with the specified key or keys. For example, suppose that you have clusters that are tagged with keys called `owner` and `environment`. If you specify both of these tag keys in the request, Amazon Redshift returns a response with the clusters that have either or both of these tag keys associated with them.
 - (*string*) –
- **TagValues** (*list*) – A tag value or values for which you want to return all matching clusters that are associated with the specified tag value or values. For example, suppose that you have clusters that are tagged with values called `admin` and `test`. If you specify both of these tag values in the request, Amazon Redshift returns a response with the clusters that have either or both of these tag values associated with them.
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'Clusters': [
    {
      'ClusterIdentifier': 'string',
      'NodeType': 'string',
      'ClusterStatus': 'string',
      'ModifyStatus': 'string',
      'MasterUsername': 'string',
      'DBName': 'string',
      'Endpoint': {
        'Address': 'string',
        'Port': 123
      },
      'ClusterCreateTime': datetime(2015, 1, 1),
      'AutomatedSnapshotRetentionPeriod': 123,
      'ClusterSecurityGroups': [
        {
          'ClusterSecurityGroupName': 'string',
          'Status': 'string'
        },
      ],
      'VpcSecurityGroups': [
        {
          'VpcSecurityGroupId': 'string',
          'Status': 'string'
        },
      ],
      'ClusterParameterGroups': [
```

```

        {
            'ParameterGroupName': 'string',
            'ParameterApplyStatus': 'string',
            'ClusterParameterStatusList': [
                {
                    'ParameterName': 'string',
                    'ParameterApplyStatus': 'string',
                    'ParameterApplyErrorDescription': 'string'
                },
            ],
        },
    ],
    'ClusterSubnetGroupName': 'string',
    'VpcId': 'string',
    'AvailabilityZone': 'string',
    'PreferredMaintenanceWindow': 'string',
    'PendingModifiedValues': {
        'MasterUserPassword': 'string',
        'NodeType': 'string',
        'NumberOfNodes': 123,
        'ClusterType': 'string',
        'ClusterVersion': 'string',
        'AutomatedSnapshotRetentionPeriod': 123,
        'ClusterIdentifier': 'string'
    },
    'ClusterVersion': 'string',
    'AllowVersionUpgrade': True|False,
    'NumberOfNodes': 123,
    'PubliclyAccessible': True|False,
    'Encrypted': True|False,
    'RestoreStatus': {
        'Status': 'string',
        'CurrentRestoreRateInMegaBytesPerSecond': 123.0,
        'SnapshotSizeInMegaBytes': 123,
        'ProgressInMegaBytes': 123,
        'ElapsedTimeInSeconds': 123,
        'EstimatedTimeToCompletionInSeconds': 123
    },
    'HsmStatus': {
        'HsmClientCertificateIdentifier': 'string',
        'HsmConfigurationIdentifier': 'string',
        'Status': 'string'
    },
    'ClusterSnapshotCopyStatus': {
        'DestinationRegion': 'string',
        'RetentionPeriod': 123,
        'SnapshotCopyGrantName': 'string'
    },
    'ClusterPublicKey': 'string',
    'ClusterNodes': [
        {
            'NodeRole': 'string',
            'PrivateIpAddress': 'string',
            'PublicIpAddress': 'string'
        },
    ],
    'ElasticIpStatus': {
        'ElasticIp': 'string',

```

```
        'Status': 'string'
    },
    'ClusterRevisionNumber': 'string',
    'Tags': [
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
    'KmsKeyId': 'string'
}
]
```

Response Structure

- (*dict*) –

Contains the output from the DescribeClusters action.

- **Marker** (*string*) –

A value that indicates the starting point for the next set of response records in a subsequent request. If a value is returned in a response, you can retrieve the next set of records by providing this returned marker value in the `Marker` parameter and retrying the command. If the `Marker` field is empty, all response records have been retrieved for the request.

- **Clusters** (*list*) –

A list of Cluster objects, where each object describes one cluster.

- * (*dict*) –

Describes a cluster.

- **ClusterIdentifier** (*string*) –

The unique identifier of the cluster.

- **NodeType** (*string*) –

The node type for the nodes in the cluster.

- **ClusterStatus** (*string*) –

The current state of this cluster. Possible values include `available`, `creating`, `deleting`, `rebooting`, `renaming`, and `resizing`.

- **ModifyStatus** (*string*) –

The status of a modify operation, if any, initiated for the cluster.

- **MasterUsername** (*string*) –

The master user name for the cluster. This name is used to connect to the database that is specified in **DBName**.

- **DBName** (*string*) –

The name of the initial database that was created when the cluster was created. This same name is returned for the life of the cluster. If an initial database was not specified, a database named “dev” was created by default.

- **Endpoint** (*dict*) –

The connection endpoint.

- **Address** (*string*) –

The DNS address of the Cluster.

- **Port** (*integer*) –

The port that the database engine is listening on.

- **ClusterCreateTime** (*datetime*) –

The date and time that the cluster was created.

- **AutomatedSnapshotRetentionPeriod** (*integer*) –

The number of days that automatic cluster snapshots are retained.

- **ClusterSecurityGroups** (*list*) –

A list of cluster security group that are associated with the cluster. Each security group is represented by an element that contains `ClusterSecurityGroup.Name` and `ClusterSecurityGroup.Status` subelements.

Cluster security groups are used when the cluster is not created in a VPC. Clusters that are created in a VPC use VPC security groups, which are listed by the **VpcSecurityGroups** parameter.

- (*dict*) –

Describes a security group.

- **ClusterSecurityGroupName** (*string*) –

The name of the cluster security group.

- **Status** (*string*) –

The status of the cluster security group.

- **VpcSecurityGroups** (*list*) –

A list of Virtual Private Cloud (VPC) security groups that are associated with the cluster. This parameter is returned only if the cluster is in a VPC.

- (*dict*) –

Describes the members of a VPC security group.

- **VpcSecurityGroupId** (*string*) –

- **Status** (*string*) –

- **ClusterParameterGroups** (*list*) –

The list of cluster parameter groups that are associated with this cluster. Each parameter group in the list is returned with its status.

- (*dict*) –

Describes the status of a parameter group.

- **ParameterGroupName** (*string*) –

The name of the cluster parameter group.

- **ParameterApplyStatus** (*string*) –

The status of parameter updates.

- **ClusterParameterStatusList** (*list*) –

The list of parameter statuses.

For more information about parameters and parameter groups, go to [Amazon Redshift Parameter Groups](#) in the *Amazon Redshift Cluster Management Guide*.

- *(dict)* –
Describes the status of a parameter group.
- **ParameterName** (*string*) –
The name of the parameter.
- **ParameterApplyStatus** (*string*) –
The status of the parameter that indicates whether the parameter is in sync with the database, waiting for a cluster reboot, or encountered an error when being applied.

The following are possible statuses and descriptions.
 - *in-sync* : The parameter value is in sync with the database.
 - *pending-reboot* : The parameter value will be applied after the cluster reboots.
 - *applying* : The parameter value is being applied to the database.
 - *invalid-parameter* : Cannot apply the parameter value because it has an invalid value or syntax.
 - *apply-deferred* : The parameter contains static property changes. The changes are deferred until the cluster reboots.
 - *apply-error* : Cannot connect to the cluster. The parameter change will be applied after the cluster reboots.
 - *unknown-error* : Cannot apply the parameter change right now. The change will be applied after the cluster reboots.
- **ParameterApplyErrorDescription** (*string*) –
The error that prevented the parameter from being applied to the database.
- **ClusterSubnetGroupName** (*string*) –
The name of the subnet group that is associated with the cluster. This parameter is valid only when the cluster is in a VPC.
- **VpcId** (*string*) –
The identifier of the VPC the cluster is in, if the cluster is in a VPC.
- **AvailabilityZone** (*string*) –
The name of the Availability Zone in which the cluster is located.
- **PreferredMaintenanceWindow** (*string*) –
The weekly time range (in UTC) during which system maintenance can occur.
- **PendingModifiedValues** (*dict*) –
If present, changes to the cluster are pending. Specific pending changes are identified by subelements.
- **MasterUserPassword** (*string*) –
The pending or in-progress change of the master user password for the cluster.
- **NodeType** (*string*) –
The pending or in-progress change of the cluster's node type.
- **NumberOfNodes** (*integer*) –

The pending or in-progress change of the number of nodes in the cluster.

- **ClusterType** (*string*) –

The pending or in-progress change of the cluster type.

- **ClusterVersion** (*string*) –

The pending or in-progress change of the service version.

- **AutomatedSnapshotRetentionPeriod** (*integer*) –

The pending or in-progress change of the automated snapshot retention period.

- **ClusterIdentifier** (*string*) –

The pending or in-progress change of the new identifier for the cluster.

- **ClusterVersion** (*string*) –

The version ID of the Amazon Redshift engine that is running on the cluster.

- **AllowVersionUpgrade** (*boolean*) –

If `true`, major version upgrades will be applied automatically to the cluster during the maintenance window.

- **NumberOfNodes** (*integer*) –

The number of compute nodes in the cluster.

- **PubliclyAccessible** (*boolean*) –

If `true`, the cluster can be accessed from a public network.

- **Encrypted** (*boolean*) –

If `true`, data in the cluster is encrypted at rest.

- **RestoreStatus** (*dict*) –

Describes the status of a cluster restore action. Returns null if the cluster was not created by restoring a snapshot.

- **Status** (*string*) –

The status of the restore action. Returns starting, restoring, completed, or failed.

- **CurrentRestoreRateInMegaBytesPerSecond** (*float*) –

The number of megabytes per second being transferred from the backup storage. Returns the average rate for a completed backup.

- **SnapshotSizeInMegaBytes** (*integer*) –

The size of the set of snapshot data used to restore the cluster.

- **ProgressInMegaBytes** (*integer*) –

The number of megabytes that have been transferred from snapshot storage.

- **ElapsedTimeInSeconds** (*integer*) –

The amount of time an in-progress restore has been running, or the amount of time it took a completed restore to finish.

- **EstimatedTimeToCompletionInSeconds** (*integer*) –

The estimate of the time remaining before the restore will complete. Returns 0 for a completed restore.

- **HsmStatus** (*dict*) –
Reports whether the Amazon Redshift cluster has finished applying any HSM settings changes specified in a modify cluster command.
Values: active, applying
- **HsmClientCertificateIdentifier** (*string*) –
Specifies the name of the HSM client certificate the Amazon Redshift cluster uses to retrieve the data encryption keys stored in an HSM.
- **HsmConfigurationIdentifier** (*string*) –
Specifies the name of the HSM configuration that contains the information the Amazon Redshift cluster can use to retrieve and store keys in an HSM.
- **Status** (*string*) –
Reports whether the Amazon Redshift cluster has finished applying any HSM settings changes specified in a modify cluster command.
Values: active, applying
- **ClusterSnapshotCopyStatus** (*dict*) –
Returns the destination region and retention period that are configured for cross-region snapshot copy.
- **DestinationRegion** (*string*) –
The destination region that snapshots are automatically copied to when cross-region snapshot copy is enabled.
- **RetentionPeriod** (*integer*) –
The number of days that automated snapshots are retained in the destination region after they are copied from a source region.
- **SnapshotCopyGrantName** (*string*) –
The name of the snapshot copy grant.
- **ClusterPublicKey** (*string*) –
The public key for the cluster.
- **ClusterNodes** (*list*) –
The nodes in a cluster.
- (*dict*) –
The identifier of a node in a cluster.
- **NodeRole** (*string*) –
Whether the node is a leader node or a compute node.
- **PrivateIpAddress** (*string*) –
The private IP address of a node within a cluster.
- **PublicIpAddress** (*string*) –
The public IP address of a node within a cluster.
- **ElasticIpStatus** (*dict*) –
Describes the status of the elastic IP (EIP) address.

- **ElasticIp** (*string*) –
The elastic IP (EIP) address for the cluster.
- **Status** (*string*) –
Describes the status of the elastic IP (EIP) address.
- **ClusterRevisionNumber** (*string*) –
The specific revision number of the database in the cluster.
- **Tags** (*list*) –
The list of tags for the cluster.
- (*dict*) –
A tag consisting of a name/value pair for a resource.
- **Key** (*string*) –
The key, or name, for the resource tag.
- **Value** (*string*) –
The value for the resource tag.
- **KmsKeyId** (*string*) –
The AWS Key Management Service (KMS) key ID of the encryption key used to encrypt data in the cluster.

describe_default_cluster_parameters (**kwargs)

Returns a list of parameter settings for the specified parameter group family.

For more information about parameters and parameter groups, go to [Amazon Redshift Parameter Groups](#) in the *Amazon Redshift Cluster Management Guide*.

Request Syntax

```
response = client.describe_default_cluster_parameters(
    ParameterGroupFamily='string',
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **ParameterGroupFamily** (*string*) – [REQUIRED]
The name of the cluster parameter group family.
- **MaxRecords** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified `MaxRecords` value, a value is returned in a `marker` field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: 100

Constraints: minimum 20, maximum 100.
- **Marker** (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeDefaultClusterParameters` request exceed the value specified in `MaxRecords`, AWS returns a value in the `Marker` field of the response. You can retrieve the next set of response records by providing the returned marker value in the `Marker` parameter and retrying the request.

Return type dict

Returns

Response Syntax

```
{
  'DefaultClusterParameters': {
    'ParameterGroupFamily': 'string',
    'Marker': 'string',
    'Parameters': [
      {
        'ParameterName': 'string',
        'ParameterValue': 'string',
        'Description': 'string',
        'Source': 'string',
        'DataType': 'string',
        'AllowedValues': 'string',
        'ApplyType': 'static'|'dynamic',
        'IsModifiable': True|False,
        'MinimumEngineVersion': 'string'
      },
    ],
  }
}
```

Response Structure

- (*dict*) –

Describes the default cluster parameters for a parameter group family.

- **DefaultClusterParameters** (*dict*) –

Describes the default cluster parameters for a parameter group family.

- * **ParameterGroupFamily** (*string*) –

The name of the cluster parameter group family to which the engine default parameters apply.

- * **Marker** (*string*) –

A value that indicates the starting point for the next set of response records in a subsequent request. If a value is returned in a response, you can retrieve the next set of records by providing this returned marker value in the `Marker` parameter and retrying the command. If the `Marker` field is empty, all response records have been retrieved for the request.

- * **Parameters** (*list*) –

The list of cluster default parameters.

- (*dict*) –

Describes a parameter in a cluster parameter group.

- **ParameterName** (*string*) –

The name of the parameter.

- **ParameterValue** (*string*) –

The value of the parameter.

- **Description** (*string*) –

A description of the parameter.

- **Source** (*string*) –

The source of the parameter value, such as “engine-default” or “user”.

- **DataType** (*string*) –

The data type of the parameter.

- **AllowedValues** (*string*) –

The valid range of values for the parameter.

- **ApplyType** (*string*) –

Specifies how to apply the parameter. Supported value: `static`.

- **IsModifiable** (*boolean*) –

If `true`, the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.

- **MinimumEngineVersion** (*string*) –

The earliest engine version to which the parameter can apply.

describe_event_categories (***kwargs*)

Displays a list of event categories for all event source types, or for a specified source type. For a list of the event categories and source types, go to [Amazon Redshift Event Notifications](#).

Request Syntax

```
response = client.describe_event_categories(
    SourceType='string'
)
```

Parameters **SourceType** (*string*) – The source type, such as cluster or parameter group, to which the described event categories apply.

Valid values: cluster, snapshot, parameter group, and security group.

Return type dict

Returns

Response Syntax

```
{
  'EventCategoriesMapList': [
    {
      'SourceType': 'string',
      'Events': [
        {
          'EventId': 'string',
          'EventCategories': [
            'string',
          ],
          'EventDescription': 'string',
          'Severity': 'string'
        },
      ],
    },
  ],
}
```

Response Structure

- (*dict*) –

– **EventCategoriesMapList** (*list*) –

A list of event categories descriptions.

* (*dict*) –

· **SourceType** (*string*) –

The Amazon Redshift source type, such as cluster or cluster-snapshot, that the returned categories belong to.

· **Events** (*list*) –

The events in the event category.

· (*dict*) –

· **EventId** (*string*) –

The identifier of an Amazon Redshift event.

· **EventCategories** (*list*) –

The category of an Amazon Redshift event.

· (*string*) –

· **EventDescription** (*string*) –

The description of an Amazon Redshift event.

· **Severity** (*string*) –

The severity of the event.

Values: ERROR, INFO

describe_event_subscriptions (***kwargs*)

Lists descriptions of all the Amazon Redshift event notifications subscription for a customer account. If you specify a subscription name, lists the description for that subscription.

Request Syntax

```
response = client.describe_event_subscriptions(
    SubscriptionName='string',
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **SubscriptionName** (*string*) – The name of the Amazon Redshift event notification subscription to be described.
- **MaxRecords** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified `MaxRecords` value, a value is returned in a `marker` field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: 100

Constraints: minimum 20, maximum 100.

- **Marker** (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeEventSubscriptions` request exceed the value specified in `MaxRecords`, AWS returns a value in the `Marker` field of the response. You can retrieve the next set of response records by providing the returned marker value in the `Marker` parameter and retrying the request.

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'EventSubscriptionsList': [
    {
      'CustomerAwsId': 'string',
      'CustSubscriptionId': 'string',
      'SnsTopicArn': 'string',
      'Status': 'string',
      'SubscriptionCreationTime': datetime(2015, 1, 1),
      'SourceType': 'string',
      'SourceIdsList': [
        'string',
      ],
      'EventCategoriesList': [
        'string',
      ],
      'Severity': 'string',
      'Enabled': True|False,
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        },
      ],
    },
  ],
}
```

Response Structure

- (*dict*) –

- **Marker** (*string*) –

A value that indicates the starting point for the next set of response records in a subsequent request. If a value is returned in a response, you can retrieve the next set of records by providing this returned marker value in the `Marker` parameter and retrying the command. If the `Marker` field is empty, all response records have been retrieved for the request.

- **EventSubscriptionsList** (*list*) –

A list of event subscriptions.

- * (*dict*) –

- **CustomerAwsId** (*string*) –

The AWS customer account associated with the Amazon Redshift event notification subscription.

- **CustSubscriptionId** (*string*) –

The name of the Amazon Redshift event notification subscription.

- **SnsTopicArn** (*string*) –

The Amazon Resource Name (ARN) of the Amazon SNS topic used by the event notification subscription.

- **Status** (*string*) –

The status of the Amazon Redshift event notification subscription.

Constraints:

- Can be one of the following: active | no-permission | topic-not-exist
- The status “no-permission” indicates that Amazon Redshift no longer has permission to post to the Amazon SNS topic. The status “topic-not-exist” indicates that the topic was deleted after the subscription was created.
- **SubscriptionCreationTime** (*datetime*) –

The date and time the Amazon Redshift event notification subscription was created.

- **SourceType** (*string*) –

The source type of the events returned the Amazon Redshift event notification, such as cluster, or cluster-snapshot.

- **SourceIdsList** (*list*) –

A list of the sources that publish events to the Amazon Redshift event notification subscription.

- (*string*) –
- **EventCategoriesList** (*list*) –

The list of Amazon Redshift event categories specified in the event notification subscription.

Values: Configuration, Management, Monitoring, Security

- (*string*) –
- **Severity** (*string*) –

The event severity specified in the Amazon Redshift event notification subscription.

Values: ERROR, INFO

- **Enabled** (*boolean*) –

A Boolean value indicating whether the subscription is enabled. `true` indicates the subscription is enabled.

- **Tags** (*list*) –

The list of tags for the event subscription.

- (*dict*) –

A tag consisting of a name/value pair for a resource.

- **Key** (*string*) –

The key, or name, for the resource tag.

- **Value** (*string*) –

The value for the resource tag.

describe_events (***kwargs*)

Returns events related to clusters, security groups, snapshots, and parameter groups for the past 14 days. Events specific to a particular cluster, security group, snapshot or parameter group can be obtained by providing the name as a parameter. By default, the past hour of events are returned.

Request Syntax

```

response = client.describe_events(
    SourceIdentifier='string',
    SourceType='cluster'|'cluster-parameter-group'|'cluster-security-group'|'cluster-snapsh
    StartTime=datetime(2015, 1, 1),
    EndTime=datetime(2015, 1, 1),
    Duration=123,
    MaxRecords=123,
    Marker='string'
)

```

Parameters

- **SourceIdentifier** (*string*) – The identifier of the event source for which events will be returned. If this parameter is not specified, then all sources are included in the response.

Constraints:

If *SourceIdentifier* is supplied, *SourceType* must also be provided.

- Specify a cluster identifier when *SourceType* is `cluster`.
- Specify a cluster security group name when *SourceType* is `cluster-security-group`.
- Specify a cluster parameter group name when *SourceType* is `cluster-parameter-group`.
- Specify a cluster snapshot identifier when *SourceType* is `cluster-snapshot`.

- **SourceType** (*string*) – The event source to retrieve events for. If no value is specified, all events are returned.

Constraints:

If *SourceType* is supplied, *SourceIdentifier* must also be provided.

- Specify `cluster` when *SourceIdentifier* is a cluster identifier.
- Specify `cluster-security-group` when *SourceIdentifier* is a cluster security group name.
- Specify `cluster-parameter-group` when *SourceIdentifier* is a cluster parameter group name.
- Specify `cluster-snapshot` when *SourceIdentifier* is a cluster snapshot identifier.

- **StartTime** (*datetime*) – The beginning of the time interval to retrieve events for, specified in ISO 8601 format. For more information about ISO 8601, go to the [ISO8601 Wikipedia page](#).

Example: 2009-07-08T18:00Z

- **EndTime** (*datetime*) – The end of the time interval for which to retrieve events, specified in ISO 8601 format. For more information about ISO 8601, go to the [ISO8601 Wikipedia page](#).

Example: 2009-07-08T18:00Z

- **Duration** (*integer*) – The number of minutes prior to the time of the request for which to retrieve events. For example, if the request is sent at 18:00 and you specify a duration of 60, then only events which have occurred after 17:00 will be returned.

Default: 60

- **MaxRecords** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified `MaxRecords` value, a value is returned in a `marker` field of the response. You

can retrieve the next set of records by retrying the command with the returned marker value.

Default: 100

Constraints: minimum 20, maximum 100.

- **Marker** (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeEvents` request exceed the value specified in `MaxRecords`, AWS returns a value in the `Marker` field of the response. You can retrieve the next set of response records by providing the returned marker value in the `Marker` parameter and retrying the request.

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'Events': [
    {
      'SourceIdentifier': 'string',
      'SourceType': 'cluster'|'cluster-parameter-group'|'cluster-security-gr...',
      'Message': 'string',
      'EventCategories': [
        'string',
      ],
      'Severity': 'string',
      'Date': datetime(2015, 1, 1),
      'EventId': 'string'
    },
  ]
}
```

Response Structure

- (*dict*) –

Contains the output from the `DescribeEvents` action.

- **Marker** (*string*) –

A value that indicates the starting point for the next set of response records in a subsequent request. If a value is returned in a response, you can retrieve the next set of records by providing this returned marker value in the `Marker` parameter and retrying the command. If the `Marker` field is empty, all response records have been retrieved for the request.

- **Events** (*list*) –

A list of Event instances.

- * (*dict*) –

Describes an event.

- **SourceIdentifier** (*string*) –

The identifier for the source of the event.

- **SourceType** (*string*) –

The source type for this event.

- **Message** (*string*) –

The text of this event.

- **EventCategories** (*list*) –
A list of the event categories.
Values: Configuration, Management, Monitoring, Security
- (*string*) –
- **Severity** (*string*) –
The severity of the event.
Values: ERROR, INFO
- **Date** (*datetime*) –
The date and time of the event.
- **EventId** (*string*) –
The identifier of the event.

describe_hsm_client_certificates (***kwargs*)

Returns information about the specified HSM client certificate. If no certificate ID is specified, returns information about all the HSM certificates owned by your AWS customer account.

If you specify both tag keys and tag values in the same request, Amazon Redshift returns all HSM client certificates that match any combination of the specified keys and values. For example, if you have `owner` and `environment` for tag keys, and `admin` and `test` for tag values, all HSM client certificates that have any combination of those values are returned.

If both tag keys and values are omitted from the request, HSM client certificates are returned regardless of whether they have tag keys or values associated with them.

Request Syntax

```
response = client.describe_hsm_client_certificates(
    HsmClientCertificateIdentifier='string',
    MaxRecords=123,
    Marker='string',
    TagKeys=[
        'string',
    ],
    TagValues=[
        'string',
    ]
)
```

Parameters

- **HsmClientCertificateIdentifier** (*string*) – The identifier of a specific HSM client certificate for which you want information. If no identifier is specified, information is returned for all HSM client certificates owned by your AWS customer account.
- **MaxRecords** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified `MaxRecords` value, a value is returned in a `marker` field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: 100

Constraints: minimum 20, maximum 100.

- **Marker** (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeHsmClientCertificates` request exceed the value specified in `MaxRecords`, AWS returns a value

in the `Marker` field of the response. You can retrieve the next set of response records by providing the returned marker value in the `Marker` parameter and retrying the request.

- **TagKeys** (*list*) – A tag key or keys for which you want to return all matching HSM client certificates that are associated with the specified key or keys. For example, suppose that you have HSM client certificates that are tagged with keys called `owner` and `environment`. If you specify both of these tag keys in the request, Amazon Redshift returns a response with the HSM client certificates that have either or both of these tag keys associated with them.
 - (*string*) –
- **TagValues** (*list*) – A tag value or values for which you want to return all matching HSM client certificates that are associated with the specified tag value or values. For example, suppose that you have HSM client certificates that are tagged with values called `admin` and `test`. If you specify both of these tag values in the request, Amazon Redshift returns a response with the HSM client certificates that have either or both of these tag values associated with them.
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'Marker': 'string',
    'HsmClientCertificates': [
        {
            'HsmClientCertificateIdentifier': 'string',
            'HsmClientCertificatePublicKey': 'string',
            'Tags': [
                {
                    'Key': 'string',
                    'Value': 'string'
                },
            ],
        },
    ],
}
```

Response Structure

- (*dict*) –
 - **Marker** (*string*) –

A value that indicates the starting point for the next set of response records in a subsequent request. If a value is returned in a response, you can retrieve the next set of records by providing this returned marker value in the `Marker` parameter and retrying the command. If the `Marker` field is empty, all response records have been retrieved for the request.
 - **HsmClientCertificates** (*list*) –

A list of the identifiers for one or more HSM client certificates used by Amazon Redshift clusters to store and retrieve database encryption keys in an HSM.

 - * (*dict*) –

Returns information about an HSM client certificate. The certificate is stored in a secure Hardware Storage Module (HSM), and used by the Amazon Redshift cluster to encrypt data files.

- **HsmClientCertificateIdentifier** (*string*) –
The identifier of the HSM client certificate.
- **HsmClientCertificatePublicKey** (*string*) –
The public key that the Amazon Redshift cluster will use to connect to the HSM. You must register the public key in the HSM.
- **Tags** (*list*) –
The list of tags for the HSM client certificate.
- (*dict*) –
A tag consisting of a name/value pair for a resource.
- **Key** (*string*) –
The key, or name, for the resource tag.
- **Value** (*string*) –
The value for the resource tag.

describe_hsm_configurations (***kwargs*)

Returns information about the specified Amazon Redshift HSM configuration. If no configuration ID is specified, returns information about all the HSM configurations owned by your AWS customer account.

If you specify both tag keys and tag values in the same request, Amazon Redshift returns all HSM connections that match any combination of the specified keys and values. For example, if you have `owner` and `environment` for tag keys, and `admin` and `test` for tag values, all HSM connections that have any combination of those values are returned.

If both tag keys and values are omitted from the request, HSM connections are returned regardless of whether they have tag keys or values associated with them.

Request Syntax

```
response = client.describe_hsm_configurations(
    HsmConfigurationIdentifier='string',
    MaxRecords=123,
    Marker='string',
    TagKeys=[
        'string',
    ],
    TagValues=[
        'string',
    ]
)
```

Parameters

- **HsmConfigurationIdentifier** (*string*) – The identifier of a specific Amazon Redshift HSM configuration to be described. If no identifier is specified, information is returned for all HSM configurations owned by your AWS customer account.
- **MaxRecords** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified `MaxRecords` value, a value is returned in a `marker` field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: 100

Constraints: minimum 20, maximum 100.

- **Marker** (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeHsmConfigurations` request exceed the value specified in `MaxRecords`, AWS returns a value in the `Marker` field of the response. You can retrieve the next set of response records by providing the returned marker value in the `Marker` parameter and retrying the request.
- **TagKeys** (*list*) – A tag key or keys for which you want to return all matching HSM configurations that are associated with the specified key or keys. For example, suppose that you have HSM configurations that are tagged with keys called `owner` and `environment`. If you specify both of these tag keys in the request, Amazon Redshift returns a response with the HSM configurations that have either or both of these tag keys associated with them.
 - (*string*) –
- **TagValues** (*list*) – A tag value or values for which you want to return all matching HSM configurations that are associated with the specified tag value or values. For example, suppose that you have HSM configurations that are tagged with values called `admin` and `test`. If you specify both of these tag values in the request, Amazon Redshift returns a response with the HSM configurations that have either or both of these tag values associated with them.
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'HsmConfigurations': [
    {
      'HsmConfigurationIdentifier': 'string',
      'Description': 'string',
      'HsmIpAddress': 'string',
      'HsmPartitionName': 'string',
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        }
      ]
    }
  ]
}
```

Response Structure

- (*dict*) –
 - **Marker** (*string*) –

A value that indicates the starting point for the next set of response records in a subsequent request. If a value is returned in a response, you can retrieve the next set of records by providing this returned marker value in the `Marker` parameter and retrying the command. If the `Marker` field is empty, all response records have been retrieved for the request.
 - **HsmConfigurations** (*list*) –

A list of Amazon Redshift HSM configurations.

 - * (*dict*) –

Returns information about an HSM configuration, which is an object that describes to Amazon Redshift clusters the information they require to connect to an HSM where they can store database encryption keys.

- **HsmConfigurationIdentifier** (*string*) –
The name of the Amazon Redshift HSM configuration.
- **Description** (*string*) –
A text description of the HSM configuration.
- **HsmIpAddress** (*string*) –
The IP address that the Amazon Redshift cluster must use to access the HSM.
- **HsmPartitionName** (*string*) –
The name of the partition in the HSM where the Amazon Redshift clusters will store their database encryption keys.
- **Tags** (*list*) –
The list of tags for the HSM configuration.
- (*dict*) –
A tag consisting of a name/value pair for a resource.
- **Key** (*string*) –
The key, or name, for the resource tag.
- **Value** (*string*) –
The value for the resource tag.

describe_logging_status (***kwargs*)

Describes whether information, such as queries and connection attempts, is being logged for the specified Amazon Redshift cluster.

Request Syntax

```
response = client.describe_logging_status(
    ClusterIdentifier='string'
)
```

Parameters **ClusterIdentifier** (*string*) – [REQUIRED]

The identifier of the cluster to get the logging status from.

Example: examplecluster

Return type dict

Returns

Response Syntax

```
{
    'LoggingEnabled': True|False,
    'BucketName': 'string',
    'S3KeyPrefix': 'string',
    'LastSuccessfulDeliveryTime': datetime(2015, 1, 1),
    'LastFailureTime': datetime(2015, 1, 1),
    'LastFailureMessage': 'string'
}
```

Response Structure

- (*dict*) –

Describes the status of logging for a cluster.

- **LoggingEnabled** (*boolean*) –
true if logging is on, false if logging is off.
- **BucketName** (*string*) –
The name of the S3 bucket where the log files are stored.
- **S3KeyPrefix** (*string*) –
The prefix applied to the log file names.
- **LastSuccessfulDeliveryTime** (*datetime*) –
The last time when logs were delivered.
- **LastFailureTime** (*datetime*) –
The last time when logs failed to be delivered.
- **LastFailureMessage** (*string*) –
The message indicating that logs failed to be delivered.

describe_orderable_cluster_options (***kwargs*)

Returns a list of orderable cluster options. Before you create a new cluster you can use this operation to find what options are available, such as the EC2 Availability Zones (AZ) in the specific AWS region that you can specify, and the node types you can request. The node types differ by available storage, memory, CPU and price. With the cost involved you might want to obtain a list of cluster options in the specific region and specify values when creating a cluster. For more information about managing clusters, go to [Amazon Redshift Clusters](#) in the *Amazon Redshift Cluster Management Guide*

Request Syntax

```
response = client.describe_orderable_cluster_options(
    ClusterVersion='string',
    NodeType='string',
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **ClusterVersion** (*string*) – The version filter value. Specify this parameter to show only the available offerings matching the specified version.

Default: All versions.

Constraints: Must be one of the version returned from DescribeClusterVersions

- **NodeType** (*string*) – The node type filter value. Specify this parameter to show only the available offerings matching the specified node type.
- **MaxRecords** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified MaxRecords value, a value is returned in a marker field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: 100

Constraints: minimum 20, maximum 100.

- **Marker** (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a DescribeOrderableClusterOptions request exceed the value specified in MaxRecords, AWS returns a

value in the `Marker` field of the response. You can retrieve the next set of response records by providing the returned marker value in the `Marker` parameter and retrying the request.

Return type dict

Returns

Response Syntax

```
{
  'OrderableClusterOptions': [
    {
      'ClusterVersion': 'string',
      'ClusterType': 'string',
      'NodeType': 'string',
      'AvailabilityZones': [
        {
          'Name': 'string'
        }
      ],
    },
  ],
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output from the `DescribeOrderableClusterOptions` action.

- **OrderableClusterOptions** (*list*) –

An `OrderableClusterOption` structure containing information about orderable options for the Cluster.

- * (*dict*) –

Describes an orderable cluster option.

- **ClusterVersion** (*string*) –

The version of the orderable cluster.

- **ClusterType** (*string*) –

The cluster type, for example `multi-node`.

- **NodeType** (*string*) –

The node type for the orderable cluster.

- **AvailabilityZones** (*list*) –

A list of availability zones for the orderable cluster.

- (*dict*) –

Describes an availability zone.

- **Name** (*string*) –

The name of the availability zone.

- **Marker** (*string*) –

A value that indicates the starting point for the next set of response records in a subsequent request. If a value is returned in a response, you can retrieve the next set of records by providing this returned marker value in the `Marker` parameter and retrying the command. If the `Marker` field is empty, all response records have been retrieved for the request.

describe_reserved_node_offerings (***kwargs*)

Returns a list of the available reserved node offerings by Amazon Redshift with their descriptions including the node type, the fixed and recurring costs of reserving the node and duration the node will be reserved for you. These descriptions help you determine which reserve node offering you want to purchase. You then use the unique offering ID in you call to `PurchaseReservedNodeOffering` to reserve one or more nodes for your Amazon Redshift cluster.

For more information about reserved node offerings, go to [Purchasing Reserved Nodes](#) in the *Amazon Redshift Cluster Management Guide*.

Request Syntax

```
response = client.describe_reserved_node_offerings(
    ReservedNodeOfferingId='string',
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **ReservedNodeOfferingId** (*string*) – The unique identifier for the offering.
- **MaxRecords** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified `MaxRecords` value, a value is returned in a `marker` field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: 100

Constraints: minimum 20, maximum 100.

- **Marker** (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeReservedNodeOfferings` request exceed the value specified in `MaxRecords`, AWS returns a value in the `Marker` field of the response. You can retrieve the next set of response records by providing the returned marker value in the `Marker` parameter and retrying the request.

Return type dict

Returns**Response Syntax**

```
{
  'Marker': 'string',
  'ReservedNodeOfferings': [
    {
      'ReservedNodeOfferingId': 'string',
      'NodeType': 'string',
      'Duration': 123,
      'FixedPrice': 123.0,
      'UsagePrice': 123.0,
      'CurrencyCode': 'string',
      'OfferingType': 'string',
      'RecurringCharges': [
        {
          'RecurringChargeAmount': 123.0,
          'RecurringChargeFrequency': 'string'
        }
      ],
    }
  ],
}
```

```
]
}
```

Response Structure

- (*dict*) –

Contains the output from the DescribeReservedNodeOfferings action.

- **Marker** (*string*) –

A value that indicates the starting point for the next set of response records in a subsequent request. If a value is returned in a response, you can retrieve the next set of records by providing this returned marker value in the `Marker` parameter and retrying the command. If the `Marker` field is empty, all response records have been retrieved for the request.

- **ReservedNodeOfferings** (*list*) –

A list of reserved node offerings.

- * (*dict*) –

Describes a reserved node offering.

- **ReservedNodeOfferingId** (*string*) –

The offering identifier.

- **NodeType** (*string*) –

The node type offered by the reserved node offering.

- **Duration** (*integer*) –

The duration, in seconds, for which the offering will reserve the node.

- **FixedPrice** (*float*) –

The upfront fixed charge you will pay to purchase the specific reserved node offering.

- **UsagePrice** (*float*) –

The rate you are charged for each hour the cluster that is using the offering is running.

- **CurrencyCode** (*string*) –

The currency code for the compute nodes offering.

- **OfferingType** (*string*) –

The anticipated utilization of the reserved node, as defined in the reserved node offering.

- **RecurringCharges** (*list*) –

The charge to your account regardless of whether you are creating any clusters using the node offering. Recurring charges are only in effect for heavy-utilization reserved nodes.

- (*dict*) –

Describes a recurring charge.

- **RecurringChargeAmount** (*float*) –

The amount charged per the period of time specified by the recurring charge frequency.

- **RecurringChargeFrequency** (*string*) –

The frequency at which the recurring charge amount is applied.

describe_reserved_nodes (**kwargs)

Returns the descriptions of the reserved nodes.

Request Syntax

```
response = client.describe_reserved_nodes(
    ReservedNodeId='string',
    MaxRecords=123,
    Marker='string'
)
```

Parameters

- **ReservedNodeId** (*string*) – Identifier for the node reservation.
- **MaxRecords** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified `MaxRecords` value, a value is returned in a `marker` field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: 100

Constraints: minimum 20, maximum 100.

- **Marker** (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeReservedNodes` request exceed the value specified in `MaxRecords`, AWS returns a value in the `Marker` field of the response. You can retrieve the next set of response records by providing the returned marker value in the `Marker` parameter and retrying the request.

Return type dict

Returns**Response Syntax**

```
{
  'Marker': 'string',
  'ReservedNodes': [
    {
      'ReservedNodeId': 'string',
      'ReservedNodeOfferingId': 'string',
      'NodeType': 'string',
      'StartTime': datetime(2015, 1, 1),
      'Duration': 123,
      'FixedPrice': 123.0,
      'UsagePrice': 123.0,
      'CurrencyCode': 'string',
      'NodeCount': 123,
      'State': 'string',
      'OfferingType': 'string',
      'RecurringCharges': [
        {
          'RecurringChargeAmount': 123.0,
          'RecurringChargeFrequency': 'string'
        }
      ],
    },
  ],
}
```

Response Structure

- *(dict)* –

Contains the output from the DescribeReservedNodes action.

- **Marker** (*string*) –

A value that indicates the starting point for the next set of response records in a subsequent request. If a value is returned in a response, you can retrieve the next set of records by providing this returned marker value in the `Marker` parameter and retrying the command. If the `Marker` field is empty, all response records have been retrieved for the request.

- **ReservedNodes** (*list*) –

The list of reserved nodes.

- * *(dict)* –

Describes a reserved node. You can call the DescribeReservedNodeOfferings API to obtain the available reserved node offerings.

- **ReservedNodeId** (*string*) –

The unique identifier for the reservation.

- **ReservedNodeOfferingId** (*string*) –

The identifier for the reserved node offering.

- **NodeType** (*string*) –

The node type of the reserved node.

- **StartTime** (*datetime*) –

The time the reservation started. You purchase a reserved node offering for a duration. This is the start time of that duration.

- **Duration** (*integer*) –

The duration of the node reservation in seconds.

- **FixedPrice** (*float*) –

The fixed cost Amazon Redshift charges you for this reserved node.

- **UsagePrice** (*float*) –

The hourly rate Amazon Redshift charges you for this reserved node.

- **CurrencyCode** (*string*) –

The currency code for the reserved cluster.

- **NodeCount** (*integer*) –

The number of reserved compute nodes.

- **State** (*string*) –

The state of the reserved compute node.

Possible Values:

- `pending-payment`-This reserved node has recently been purchased, and the sale has been approved, but payment has not yet been confirmed.
- `active`-This reserved node is owned by the caller and is available for use.
- `payment-failed`-Payment failed for the purchase attempt.

- **OfferingType** (*string*) –
The anticipated utilization of the reserved node, as defined in the reserved node offering.
- **RecurringCharges** (*list*) –
The recurring charges for the reserved node.
- (*dict*) –
Describes a recurring charge.
- **RecurringChargeAmount** (*float*) –
The amount charged per the period of time specified by the recurring charge frequency.
- **RecurringChargeFrequency** (*string*) –
The frequency at which the recurring charge amount is applied.

describe_resize (***kwargs*)

Returns information about the last resize operation for the specified cluster. If no resize operation has ever been initiated for the specified cluster, a HTTP 404 error is returned. If a resize operation was initiated and completed, the status of the resize remains as SUCCEEDED until the next resize.

A resize operation can be requested using `ModifyCluster` and specifying a different number or type of nodes for the cluster.

Request Syntax

```
response = client.describe_resize(  
    ClusterIdentifier='string'  
)
```

Parameters **ClusterIdentifier** (*string*) – [REQUIRED]

The unique identifier of a cluster whose resize progress you are requesting. This parameter is case-sensitive.

By default, resize operations for all clusters defined for an AWS account are returned.

Return type dict

Returns**Response Syntax**

```
{  
    'TargetNodeType': 'string',  
    'TargetNumberOfNodes': 123,  
    'TargetClusterType': 'string',  
    'Status': 'string',  
    'ImportTablesCompleted': [  
        'string',  
    ],  
    'ImportTablesInProgress': [  
        'string',  
    ],  
    'ImportTablesNotStarted': [  
        'string',  
    ],  
    'AvgResizeRateInMegaBytesPerSecond': 123.0,  
    'TotalResizeDataInMegaBytes': 123,  
    'ProgressInMegaBytes': 123,
```



```

    'ElapsedTimeInSeconds': 123,
    'EstimatedTimeToCompletionInSeconds': 123
}

```

Response Structure

- (*dict*) –

Describes the result of a cluster resize operation.

- **TargetNodeType** (*string*) –

The node type that the cluster will have after the resize operation is complete.

- **TargetNumberOfNodes** (*integer*) –

The number of nodes that the cluster will have after the resize operation is complete.

- **TargetClusterType** (*string*) –

The cluster type after the resize operation is complete.

Valid Values: multi-node | single-node

- **Status** (*string*) –

The status of the resize operation.

Valid Values: NONE | IN_PROGRESS | FAILED | SUCCEEDED

- **ImportTablesCompleted** (*list*) –

The names of tables that have been completely imported .

Valid Values: List of table names.

* (*string*) –

- **ImportTablesInProgress** (*list*) –

The names of tables that are being currently imported.

Valid Values: List of table names.

* (*string*) –

- **ImportTablesNotStarted** (*list*) –

The names of tables that have not been yet imported.

Valid Values: List of table names

* (*string*) –

- **AvgResizeRateInMegaBytesPerSecond** (*float*) –

The average rate of the resize operation over the last few minutes, measured in megabytes per second. After the resize operation completes, this value shows the average rate of the entire resize operation.

- **TotalResizeDataInMegaBytes** (*integer*) –

The estimated total amount of data, in megabytes, on the cluster before the resize operation began.

- **ProgressInMegaBytes** (*integer*) –

While the resize operation is in progress, this value shows the current amount of data, in megabytes, that has been processed so far. When the resize operation is complete, this value shows the total amount of data, in megabytes, on the cluster, which may be more or less than TotalResizeDataInMegaBytes (the estimated total amount of data before resize).

- **ElapsedTimeInSeconds** (*integer*) –

The amount of seconds that have elapsed since the resize operation began. After the resize operation completes, this value shows the total actual time, in seconds, for the resize operation.

- **EstimatedTimeToCompletionInSeconds** (*integer*) –

The estimated time remaining, in seconds, until the resize operation is complete. This value is calculated based on the average resize rate and the estimated amount of data remaining to be processed. Once the resize operation is complete, this value will be 0.

describe_snapshot_copy_grants (***kwargs*)

Returns a list of snapshot copy grants owned by the AWS account in the destination region.

For more information about managing snapshot copy grants, go to [Amazon Redshift Database Encryption](#) in the *Amazon Redshift Cluster Management Guide* .

Request Syntax

```
response = client.describe_snapshot_copy_grants(
    SnapshotCopyGrantName='string',
    MaxRecords=123,
    Marker='string',
    TagKeys=[
        'string',
    ],
    TagValues=[
        'string',
    ]
)
```

Parameters

- **SnapshotCopyGrantName** (*string*) – The name of the snapshot copy grant.
- **MaxRecords** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified `MaxRecords` value, a value is returned in a `marker` field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: 100

Constraints: minimum 20, maximum 100.

- **Marker** (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeSnapshotCopyGrant` request exceed the value specified in `MaxRecords` , AWS returns a value in the `Marker` field of the response. You can retrieve the next set of response records by providing the returned marker value in the `Marker` parameter and retrying the request.

Constraints: You can specify either the **SnapshotCopyGrantName** parameter or the **Marker** parameter, but not both.

- **TagKeys** (*list*) – A tag key or keys for which you want to return all matching resources that are associated with the specified key or keys. For example, suppose that you have resources tagged with keys called `owner` and `environment` . If you specify both of these tag keys in the request, Amazon Redshift returns a response with all resources that have either or both of these tag keys associated with them.

- (*string*) –

- **TagValues** (*list*) – A tag value or values for which you want to return all matching resources that are associated with the specified value or values. For example, suppose that you have resources tagged with values called `admin` and `test`. If you specify both of these tag values in the request, Amazon Redshift returns a response with all resources that have either or both of these tag values associated with them.

– (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'Marker': 'string',
  'SnapshotCopyGrants': [
    {
      'SnapshotCopyGrantName': 'string',
      'KmsKeyId': 'string',
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        },
      ],
    },
  ]
}
```

Response Structure

- (*dict*) –

The result of the snapshot copy grant.

- **Marker** (*string*) –

An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeSnapshotCopyGrant` request exceed the value specified in `MaxRecords`, AWS returns a value in the `Marker` field of the response. You can retrieve the next set of response records by providing the returned marker value in the `Marker` parameter and retrying the request.

Constraints: You can specify either the **SnapshotCopyGrantName** parameter or the **Marker** parameter, but not both.

- **SnapshotCopyGrants** (*list*) –

The list of snapshot copy grants.

- * (*dict*) –

The snapshot copy grant that grants Amazon Redshift permission to encrypt copied snapshots with the specified customer master key (CMK) from AWS KMS in the destination region.

For more information about managing snapshot copy grants, go to [Amazon Redshift Database Encryption](#) in the *Amazon Redshift Cluster Management Guide*.

- **SnapshotCopyGrantName** (*string*) –

The name of the snapshot copy grant.

- **KmsKeyId** (*string*) –
The unique identifier of the customer master key (CMK) in AWS KMS to which Amazon Redshift is granted permission.
- **Tags** (*list*) –
A list of tag instances.
- (*dict*) –
A tag consisting of a name/value pair for a resource.
- **Key** (*string*) –
The key, or name, for the resource tag.
- **Value** (*string*) –
The value for the resource tag.

describe_tags (***kwargs*)

Returns a list of tags. You can return tags from a specific resource by specifying an ARN, or you can return all tags for a given type of resource, such as clusters, snapshots, and so on.

The following are limitations for `DescribeTags` :

- You cannot specify an ARN and a resource-type value together in the same request.
- You cannot use the `MaxRecords` and `Marker` parameters together with the ARN parameter.
- The `MaxRecords` parameter can be a range from 10 to 50 results to return in a request.

If you specify both tag keys and tag values in the same request, Amazon Redshift returns all resources that match any combination of the specified keys and values. For example, if you have `owner` and `environment` for tag keys, and `admin` and `test` for tag values, all resources that have any combination of those values are returned.

If both tag keys and values are omitted from the request, resources are returned regardless of whether they have tag keys or values associated with them.

Request Syntax

```
response = client.describe_tags(  
    ResourceName='string',  
    ResourceType='string',  
    MaxRecords=123,  
    Marker='string',  
    TagKeys=[  
        'string',  
    ],  
    TagValues=[  
        'string',  
    ]  
)
```

Parameters

- **ResourceName** (*string*) – The Amazon Resource Name (ARN) for which you want to describe the tag or tags. For example, `arn:aws:redshift:us-east-1:123456789:cluster:t1`.
- **ResourceType** (*string*) – The type of resource with which you want to view tags. Valid resource types are:
 - Cluster
 - CIDR/IP
 - EC2 security group
 - Snapshot
 - Cluster security group

- Subnet group
- HSM connection
- HSM certificate
- Parameter group
- Snapshot copy grant

For more information about Amazon Redshift resource types and constructing ARNs, go to [Constructing an Amazon Redshift Amazon Resource Name \(ARN\)](#) in the Amazon Redshift Cluster Management Guide.

- **MaxRecords** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified MaxRecords value, a value is returned in a `marker` field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.
- **Marker** (*string*) – A value that indicates the starting point for the next set of response records in a subsequent request. If a value is returned in a response, you can retrieve the next set of records by providing this returned marker value in the `marker` parameter and retrying the command. If the `marker` field is empty, all response records have been retrieved for the request.
- **TagKeys** (*list*) – A tag key or keys for which you want to return all matching resources that are associated with the specified key or keys. For example, suppose that you have resources tagged with keys called `owner` and `environment`. If you specify both of these tag keys in the request, Amazon Redshift returns a response with all resources that have either or both of these tag keys associated with them.
 - (*string*) –
- **TagValues** (*list*) – A tag value or values for which you want to return all matching resources that are associated with the specified value or values. For example, suppose that you have resources tagged with values called `admin` and `test`. If you specify both of these tag values in the request, Amazon Redshift returns a response with all resources that have either or both of these tag values associated with them.
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'TaggedResources': [
    {
      'Tag': {
        'Key': 'string',
        'Value': 'string'
      },
      'ResourceName': 'string',
      'ResourceType': 'string'
    },
  ],
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –
 - Contains the output from the `DescribeTags` action.
 - **TaggedResources** (*list*) –

A list of tags with their associated resources.

* (*dict*) –

A tag and its associated resource.

· **Tag** (*dict*) –

The tag for the resource.

· **Key** (*string*) –

The key, or name, for the resource tag.

· **Value** (*string*) –

The value for the resource tag.

· **ResourceName** (*string*) –

The Amazon Resource Name (ARN) with which the tag is associated. For example, `arn:aws:redshift:us-east-1:123456789:cluster:t1`.

· **ResourceType** (*string*) –

The type of resource with which the tag is associated. Valid resource types are:

- Cluster
- CIDR/IP
- EC2 security group
- Snapshot
- Cluster security group
- Subnet group
- HSM connection
- HSM certificate
- Parameter group

For more information about Amazon Redshift resource types and constructing ARNs, go to [Constructing an Amazon Redshift Amazon Resource Name \(ARN\)](#) in the Amazon Redshift Cluster Management Guide.

– **Marker** (*string*) –

A value that indicates the starting point for the next set of response records in a subsequent request. If a value is returned in a response, you can retrieve the next set of records by providing this returned marker value in the `Marker` parameter and retrying the command. If the `Marker` field is empty, all response records have been retrieved for the request.

disable_logging (***kwargs*)

Stops logging information, such as queries and connection attempts, for the specified Amazon Redshift cluster.

Request Syntax

```
response = client.disable_logging(  
    ClusterIdentifier='string'  
)
```

Parameters `ClusterIdentifier` (*string*) – [REQUIRED]

The identifier of the cluster on which logging is to be stopped.

Example: `examplecluster`

Return type `dict`

Returns

Response Syntax

```
{
    'LoggingEnabled': True|False,
    'BucketName': 'string',
    'S3KeyPrefix': 'string',
    'LastSuccessfulDeliveryTime': datetime(2015, 1, 1),
    'LastFailureTime': datetime(2015, 1, 1),
    'LastFailureMessage': 'string'
}
```

Response Structure

- (*dict*) –

Describes the status of logging for a cluster.

- **LoggingEnabled** (*boolean*) –
true if logging is on, false if logging is off.
- **BucketName** (*string*) –
The name of the S3 bucket where the log files are stored.
- **S3KeyPrefix** (*string*) –
The prefix applied to the log file names.
- **LastSuccessfulDeliveryTime** (*datetime*) –
The last time when logs were delivered.
- **LastFailureTime** (*datetime*) –
The last time when logs failed to be delivered.
- **LastFailureMessage** (*string*) –
The message indicating that logs failed to be delivered.

disable_snapshot_copy (***kwargs*)

Disables the automatic copying of snapshots from one region to another region for a specified cluster.

If your cluster and its snapshots are encrypted using a customer master key (CMK) from AWS KMS, use `DeleteSnapshotCopyGrant` to delete the grant that grants Amazon Redshift permission to the CMK in the destination region.

Request Syntax

```
response = client.disable_snapshot_copy(
    ClusterIdentifier='string'
)
```

Parameters **ClusterIdentifier** (*string*) – [REQUIRED]

The unique identifier of the source cluster that you want to disable copying of snapshots to a destination region.

Constraints: Must be the valid name of an existing cluster that has cross-region snapshot copy enabled.

Return type dict

Returns

Response Syntax

```
{
  'Cluster': {
    'ClusterIdentifier': 'string',
    'NodeType': 'string',
    'ClusterStatus': 'string',
    'ModifyStatus': 'string',
    'MasterUsername': 'string',
    'DBName': 'string',
    'Endpoint': {
      'Address': 'string',
      'Port': 123
    },
  },
  'ClusterCreateTime': datetime(2015, 1, 1),
  'AutomatedSnapshotRetentionPeriod': 123,
  'ClusterSecurityGroups': [
    {
      'ClusterSecurityGroupName': 'string',
      'Status': 'string'
    },
  ],
  'VpcSecurityGroups': [
    {
      'VpcSecurityGroupId': 'string',
      'Status': 'string'
    },
  ],
  'ClusterParameterGroups': [
    {
      'ParameterGroupName': 'string',
      'ParameterApplyStatus': 'string',
      'ClusterParameterStatusList': [
        {
          'ParameterName': 'string',
          'ParameterApplyStatus': 'string',
          'ParameterApplyErrorDescription': 'string'
        },
      ],
    },
  ],
  'ClusterSubnetGroupName': 'string',
  'VpcId': 'string',
  'AvailabilityZone': 'string',
  'PreferredMaintenanceWindow': 'string',
  'PendingModifiedValues': {
    'MasterUserPassword': 'string',
    'NodeType': 'string',
    'NumberOfNodes': 123,
    'ClusterType': 'string',
    'ClusterVersion': 'string',
    'AutomatedSnapshotRetentionPeriod': 123,
    'ClusterIdentifier': 'string'
  },
  'ClusterVersion': 'string',
  'AllowVersionUpgrade': True|False,
  'NumberOfNodes': 123,
  'PubliclyAccessible': True|False,
  'Encrypted': True|False,
  'RestoreStatus': {
```



```

        'Status': 'string',
        'CurrentRestoreRateInMegaBytesPerSecond': 123.0,
        'SnapshotSizeInMegaBytes': 123,
        'ProgressInMegaBytes': 123,
        'ElapsedTimeInSeconds': 123,
        'EstimatedTimeToCompletionInSeconds': 123
    },
    'HsmStatus': {
        'HsmClientCertificateIdentifier': 'string',
        'HsmConfigurationIdentifier': 'string',
        'Status': 'string'
    },
    'ClusterSnapshotCopyStatus': {
        'DestinationRegion': 'string',
        'RetentionPeriod': 123,
        'SnapshotCopyGrantName': 'string'
    },
    'ClusterPublicKey': 'string',
    'ClusterNodes': [
        {
            'NodeRole': 'string',
            'PrivateIPAddress': 'string',
            'PublicIPAddress': 'string'
        },
    ],
    'ElasticIpStatus': {
        'ElasticIp': 'string',
        'Status': 'string'
    },
    'ClusterRevisionNumber': 'string',
    'Tags': [
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
    'KmsKeyId': 'string'
}

```

Response Structure

- (*dict*) –

Describes a cluster.

- **Cluster** (*dict*) –

Describes a cluster.

- * **ClusterIdentifier** (*string*) –

The unique identifier of the cluster.

- * **NodeType** (*string*) –

The node type for the nodes in the cluster.

- * **ClusterStatus** (*string*) –

The current state of this cluster. Possible values include available , creating , deleting , rebooting , renaming , and resizing .

- * **ModifyStatus** (*string*) –

The status of a modify operation, if any, initiated for the cluster.

* **MasterUsername** (*string*) –

The master user name for the cluster. This name is used to connect to the database that is specified in **DBName**.

* **DBName** (*string*) –

The name of the initial database that was created when the cluster was created. This same name is returned for the life of the cluster. If an initial database was not specified, a database named “dev” was created by default.

* **Endpoint** (*dict*) –

The connection endpoint.

· **Address** (*string*) –

The DNS address of the Cluster.

· **Port** (*integer*) –

The port that the database engine is listening on.

* **ClusterCreateTime** (*datetime*) –

The date and time that the cluster was created.

* **AutomatedSnapshotRetentionPeriod** (*integer*) –

The number of days that automatic cluster snapshots are retained.

* **ClusterSecurityGroups** (*list*) –

A list of cluster security group that are associated with the cluster. Each security group is represented by an element that contains `ClusterSecurityGroup.Name` and `ClusterSecurityGroup.Status` subelements.

Cluster security groups are used when the cluster is not created in a VPC. Clusters that are created in a VPC use VPC security groups, which are listed by the **VpcSecurityGroups** parameter.

· (*dict*) –

Describes a security group.

· **ClusterSecurityGroupName** (*string*) –

The name of the cluster security group.

· **Status** (*string*) –

The status of the cluster security group.

* **VpcSecurityGroups** (*list*) –

A list of Virtual Private Cloud (VPC) security groups that are associated with the cluster. This parameter is returned only if the cluster is in a VPC.

· (*dict*) –

Describes the members of a VPC security group.

· **VpcSecurityGroupId** (*string*) –

· **Status** (*string*) –

* **ClusterParameterGroups** (*list*) –

The list of cluster parameter groups that are associated with this cluster. Each parameter group in the list is returned with its status.

· (*dict*) –

Describes the status of a parameter group.

- **ParameterGroupName** (*string*) –

The name of the cluster parameter group.

- **ParameterApplyStatus** (*string*) –

The status of parameter updates.

- **ClusterParameterStatusList** (*list*) –

The list of parameter statuses.

For more information about parameters and parameter groups, go to [Amazon Redshift Parameter Groups](#) in the *Amazon Redshift Cluster Management Guide*.

- (*dict*) –

Describes the status of a parameter group.

- **ParameterName** (*string*) –

The name of the parameter.

- **ParameterApplyStatus** (*string*) –

The status of the parameter that indicates whether the parameter is in sync with the database, waiting for a cluster reboot, or encountered an error when being applied.

The following are possible statuses and descriptions.

- `in-sync` : The parameter value is in sync with the database.
- `pending-reboot` : The parameter value will be applied after the cluster reboots.
- `applying` : The parameter value is being applied to the database.
- `invalid-parameter` : Cannot apply the parameter value because it has an invalid value or syntax.
- `apply-deferred` : The parameter contains static property changes. The changes are deferred until the cluster reboots.
- `apply-error` : Cannot connect to the cluster. The parameter change will be applied after the cluster reboots.
- `unknown-error` : Cannot apply the parameter change right now. The change will be applied after the cluster reboots.

- **ParameterApplyErrorDescription** (*string*) –

The error that prevented the parameter from being applied to the database.

- * **ClusterSubnetGroupName** (*string*) –

The name of the subnet group that is associated with the cluster. This parameter is valid only when the cluster is in a VPC.

- * **VpcId** (*string*) –

The identifier of the VPC the cluster is in, if the cluster is in a VPC.

- * **AvailabilityZone** (*string*) –

The name of the Availability Zone in which the cluster is located.

- * **PreferredMaintenanceWindow** (*string*) –

The weekly time range (in UTC) during which system maintenance can occur.

- * **PendingModifiedValues** (*dict*) –

If present, changes to the cluster are pending. Specific pending changes are identified by subelements.

- **MasterUserPassword** (*string*) –

The pending or in-progress change of the master user password for the cluster.

- **NodeType** (*string*) –

The pending or in-progress change of the cluster's node type.

- **NumberOfNodes** (*integer*) –

The pending or in-progress change of the number of nodes in the cluster.

- **ClusterType** (*string*) –

The pending or in-progress change of the cluster type.

- **ClusterVersion** (*string*) –

The pending or in-progress change of the service version.

- **AutomatedSnapshotRetentionPeriod** (*integer*) –

The pending or in-progress change of the automated snapshot retention period.

- **ClusterIdentifier** (*string*) –

The pending or in-progress change of the new identifier for the cluster.

- * **ClusterVersion** (*string*) –

The version ID of the Amazon Redshift engine that is running on the cluster.

- * **AllowVersionUpgrade** (*boolean*) –

If `true`, major version upgrades will be applied automatically to the cluster during the maintenance window.

- * **NumberOfNodes** (*integer*) –

The number of compute nodes in the cluster.

- * **PubliclyAccessible** (*boolean*) –

If `true`, the cluster can be accessed from a public network.

- * **Encrypted** (*boolean*) –

If `true`, data in the cluster is encrypted at rest.

- * **RestoreStatus** (*dict*) –

Describes the status of a cluster restore action. Returns null if the cluster was not created by restoring a snapshot.

- **Status** (*string*) –

The status of the restore action. Returns starting, restoring, completed, or failed.

- **CurrentRestoreRateInMegaBytesPerSecond** (*float*) –

The number of megabytes per second being transferred from the backup storage. Returns the average rate for a completed backup.

- **SnapshotSizeInMegaBytes** (*integer*) –

The size of the set of snapshot data used to restore the cluster.

- **ProgressInMegaBytes** (*integer*) –

The number of megabytes that have been transferred from snapshot storage.

- **ElapsedTimeInSeconds** (*integer*) –

The amount of time an in-progress restore has been running, or the amount of time it took a completed restore to finish.

- **EstimatedTimeToCompletionInSeconds** (*integer*) –

The estimate of the time remaining before the restore will complete. Returns 0 for a completed restore.

- * **HsmStatus** (*dict*) –

Reports whether the Amazon Redshift cluster has finished applying any HSM settings changes specified in a modify cluster command.

Values: active, applying

- **HsmClientCertificateIdentifier** (*string*) –

Specifies the name of the HSM client certificate the Amazon Redshift cluster uses to retrieve the data encryption keys stored in an HSM.

- **HsmConfigurationIdentifier** (*string*) –

Specifies the name of the HSM configuration that contains the information the Amazon Redshift cluster can use to retrieve and store keys in an HSM.

- **Status** (*string*) –

Reports whether the Amazon Redshift cluster has finished applying any HSM settings changes specified in a modify cluster command.

Values: active, applying

- * **ClusterSnapshotCopyStatus** (*dict*) –

Returns the destination region and retention period that are configured for cross-region snapshot copy.

- **DestinationRegion** (*string*) –

The destination region that snapshots are automatically copied to when cross-region snapshot copy is enabled.

- **RetentionPeriod** (*integer*) –

The number of days that automated snapshots are retained in the destination region after they are copied from a source region.

- **SnapshotCopyGrantName** (*string*) –

The name of the snapshot copy grant.

- * **ClusterPublicKey** (*string*) –

The public key for the cluster.

- * **ClusterNodes** (*list*) –

The nodes in a cluster.

- (*dict*) –

The identifier of a node in a cluster.

- **NodeRole** (*string*) –

Whether the node is a leader node or a compute node.

- **PrivateIPAddress** (*string*) –
The private IP address of a node within a cluster.
- **PublicIPAddress** (*string*) –
The public IP address of a node within a cluster.
- * **ElasticIpStatus** (*dict*) –
Describes the status of the elastic IP (EIP) address.
 - **ElasticIp** (*string*) –
The elastic IP (EIP) address for the cluster.
 - **Status** (*string*) –
Describes the status of the elastic IP (EIP) address.
- * **ClusterRevisionNumber** (*string*) –
The specific revision number of the database in the cluster.
- * **Tags** (*list*) –
The list of tags for the cluster.
 - (*dict*) –
A tag consisting of a name/value pair for a resource.
 - **Key** (*string*) –
The key, or name, for the resource tag.
 - **Value** (*string*) –
The value for the resource tag.
- * **KmsKeyId** (*string*) –
The AWS Key Management Service (KMS) key ID of the encryption key used to encrypt data in the cluster.

enable_logging (***kwargs*)

Starts logging information, such as queries and connection attempts, for the specified Amazon Redshift cluster.

Request Syntax

```
response = client.enable_logging(  
    ClusterIdentifier='string',  
    BucketName='string',  
    S3KeyPrefix='string'  
)
```

Parameters

- **ClusterIdentifier** (*string*) – **[REQUIRED]**
The identifier of the cluster on which logging is to be started.
Example: examplecluster
- **BucketName** (*string*) – **[REQUIRED]**
The name of an existing S3 bucket where the log files are to be stored.
Constraints:
 - Must be in the same region as the cluster
 - The cluster must have read bucket and put object permissions
- **S3KeyPrefix** (*string*) – The prefix applied to the log file names.
Constraints:

- Cannot exceed 512 characters
- Cannot contain spaces(), double quotes ("), single quotes ('), a backslash (\\), or control characters. The hexadecimal codes for invalid characters are:
 - * x00 to x20
- x22
- x27
- x5c
- x7f or larger

Return type dict

Returns

Response Syntax

```
{
    'LoggingEnabled': True|False,
    'BucketName': 'string',
    'S3KeyPrefix': 'string',
    'LastSuccessfulDeliveryTime': datetime(2015, 1, 1),
    'LastFailureTime': datetime(2015, 1, 1),
    'LastFailureMessage': 'string'
}
```

Response Structure

- (*dict*) –

Describes the status of logging for a cluster.

- **LoggingEnabled** (*boolean*) –

true if logging is on, false if logging is off.
- **BucketName** (*string*) –

The name of the S3 bucket where the log files are stored.
- **S3KeyPrefix** (*string*) –

The prefix applied to the log file names.
- **LastSuccessfulDeliveryTime** (*datetime*) –

The last time when logs were delivered.
- **LastFailureTime** (*datetime*) –

The last time when logs failed to be delivered.
- **LastFailureMessage** (*string*) –

The message indicating that logs failed to be delivered.

enable_snapshot_copy (***kwargs*)

Enables the automatic copy of snapshots from one region to another region for a specified cluster.

Request Syntax

```
response = client.enable_snapshot_copy(
    ClusterIdentifier='string',
    DestinationRegion='string',
    RetentionPeriod=123,
    SnapshotCopyGrantName='string'
)
```

Parameters

- **ClusterIdentifier** (*string*) – **[REQUIRED]**

The unique identifier of the source cluster to copy snapshots from.

Constraints: Must be the valid name of an existing cluster that does not already have cross-region snapshot copy enabled.

- **DestinationRegion** (*string*) – [REQUIRED]

The destination region that you want to copy snapshots to.

Constraints: Must be the name of a valid region. For more information, see [Regions and Endpoints](#) in the Amazon Web Services General Reference.

- **RetentionPeriod** (*integer*) – The number of days to retain automated snapshots in the destination region after they are copied from the source region.

Default: 7.

Constraints: Must be at least 1 and no more than 35.

- **SnapshotCopyGrantName** (*string*) – The name of the snapshot copy grant to use when snapshots of an AWS KMS-encrypted cluster are copied to the destination region.

Return type dict

Returns

Response Syntax

```
{
  'Cluster': {
    'ClusterIdentifier': 'string',
    'NodeType': 'string',
    'ClusterStatus': 'string',
    'ModifyStatus': 'string',
    'MasterUsername': 'string',
    'DBName': 'string',
    'Endpoint': {
      'Address': 'string',
      'Port': 123
    },
  },
  'ClusterCreateTime': datetime(2015, 1, 1),
  'AutomatedSnapshotRetentionPeriod': 123,
  'ClusterSecurityGroups': [
    {
      'ClusterSecurityGroupName': 'string',
      'Status': 'string'
    },
  ],
  'VpcSecurityGroups': [
    {
      'VpcSecurityGroupId': 'string',
      'Status': 'string'
    },
  ],
  'ClusterParameterGroups': [
    {
      'ParameterGroupName': 'string',
      'ParameterApplyStatus': 'string',
      'ClusterParameterStatusList': [
        {
          'ParameterName': 'string',
          'ParameterApplyStatus': 'string',
          'ParameterApplyErrorDescription': 'string'
        },
      ],
    },
  ]
}
```



```

    },
],
'ClusterSubnetGroupName': 'string',
'VpcId': 'string',
'AvailabilityZone': 'string',
'PreferredMaintenanceWindow': 'string',
'PendingModifiedValues': {
    'MasterUserPassword': 'string',
    'NodeType': 'string',
    'NumberOfNodes': 123,
    'ClusterType': 'string',
    'ClusterVersion': 'string',
    'AutomatedSnapshotRetentionPeriod': 123,
    'ClusterIdentifier': 'string'
},
'ClusterVersion': 'string',
'AllowVersionUpgrade': True|False,
'NumberOfNodes': 123,
'PubliclyAccessible': True|False,
'Encrypted': True|False,
'RestoreStatus': {
    'Status': 'string',
    'CurrentRestoreRateInMegaBytesPerSecond': 123.0,
    'SnapshotSizeInMegaBytes': 123,
    'ProgressInMegaBytes': 123,
    'ElapsedTimeInSeconds': 123,
    'EstimatedTimeToCompletionInSeconds': 123
},
'HsmStatus': {
    'HsmClientCertificateIdentifier': 'string',
    'HsmConfigurationIdentifier': 'string',
    'Status': 'string'
},
'ClusterSnapshotCopyStatus': {
    'DestinationRegion': 'string',
    'RetentionPeriod': 123,
    'SnapshotCopyGrantName': 'string'
},
'ClusterPublicKey': 'string',
'ClusterNodes': [
    {
        'NodeRole': 'string',
        'PrivateIpAddress': 'string',
        'PublicIpAddress': 'string'
    },
],
'ElasticIpStatus': {
    'ElasticIp': 'string',
    'Status': 'string'
},
'ClusterRevisionNumber': 'string',
'Tags': [
    {
        'Key': 'string',
        'Value': 'string'
    },
],
'KmsKeyId': 'string'

```

```
    }  
}
```

Response Structure

- (*dict*) –

Describes a cluster.

- **Cluster** (*dict*) –

Describes a cluster.

- * **ClusterIdentifier** (*string*) –

The unique identifier of the cluster.

- * **NodeType** (*string*) –

The node type for the nodes in the cluster.

- * **ClusterStatus** (*string*) –

The current state of this cluster. Possible values include `available`, `creating`, `deleting`, `rebooting`, `renaming`, and `resizing`.

- * **ModifyStatus** (*string*) –

The status of a modify operation, if any, initiated for the cluster.

- * **MasterUsername** (*string*) –

The master user name for the cluster. This name is used to connect to the database that is specified in **DBName**.

- * **DBName** (*string*) –

The name of the initial database that was created when the cluster was created. This same name is returned for the life of the cluster. If an initial database was not specified, a database named “dev” was created by default.

- * **Endpoint** (*dict*) –

The connection endpoint.

- **Address** (*string*) –

The DNS address of the Cluster.

- **Port** (*integer*) –

The port that the database engine is listening on.

- * **ClusterCreateTime** (*datetime*) –

The date and time that the cluster was created.

- * **AutomatedSnapshotRetentionPeriod** (*integer*) –

The number of days that automatic cluster snapshots are retained.

- * **ClusterSecurityGroups** (*list*) –

A list of cluster security group that are associated with the cluster. Each security group is represented by an element that contains `ClusterSecurityGroup.Name` and `ClusterSecurityGroup.Status` subelements.

Cluster security groups are used when the cluster is not created in a VPC. Clusters that are created in a VPC use VPC security groups, which are listed by the **VpcSecurityGroups** parameter.

- (*dict*) –

Describes a security group.

- **ClusterSecurityGroupName** (*string*) –
The name of the cluster security group.
- **Status** (*string*) –
The status of the cluster security group.
- * **VpcSecurityGroups** (*list*) –
A list of Virtual Private Cloud (VPC) security groups that are associated with the cluster. This parameter is returned only if the cluster is in a VPC.
 - (*dict*) –
Describes the members of a VPC security group.
 - **VpcSecurityGroupId** (*string*) –
 - **Status** (*string*) –
- * **ClusterParameterGroups** (*list*) –
The list of cluster parameter groups that are associated with this cluster. Each parameter group in the list is returned with its status.
 - (*dict*) –
Describes the status of a parameter group.
 - **ParameterGroupName** (*string*) –
The name of the cluster parameter group.
 - **ParameterApplyStatus** (*string*) –
The status of parameter updates.
 - **ClusterParameterStatusList** (*list*) –
The list of parameter statuses.

For more information about parameters and parameter groups, go to [Amazon Redshift Parameter Groups](#) in the *Amazon Redshift Cluster Management Guide*.
 - (*dict*) –
Describes the status of a parameter group.
 - **ParameterName** (*string*) –
The name of the parameter.
 - **ParameterApplyStatus** (*string*) –
The status of the parameter that indicates whether the parameter is in sync with the database, waiting for a cluster reboot, or encountered an error when being applied.

The following are possible statuses and descriptions.
 - **in-sync** : The parameter value is in sync with the database.
 - **pending-reboot** : The parameter value will be applied after the cluster reboots.
 - **applying** : The parameter value is being applied to the database.
 - **invalid-parameter** : Cannot apply the parameter value because it has an invalid value or syntax.
 - **apply-deferred** : The parameter contains static property changes. The changes are deferred until the cluster reboots.
 - **apply-error** : Cannot connect to the cluster. The parameter change will be applied after the cluster reboots.

- `unknown-error` : Cannot apply the parameter change right now. The change will be applied after the cluster reboots.
- **ParameterApplyErrorDescription** (*string*) –
The error that prevented the parameter from being applied to the database.
- * **ClusterSubnetGroupName** (*string*) –
The name of the subnet group that is associated with the cluster. This parameter is valid only when the cluster is in a VPC.
- * **VpcId** (*string*) –
The identifier of the VPC the cluster is in, if the cluster is in a VPC.
- * **AvailabilityZone** (*string*) –
The name of the Availability Zone in which the cluster is located.
- * **PreferredMaintenanceWindow** (*string*) –
The weekly time range (in UTC) during which system maintenance can occur.
- * **PendingModifiedValues** (*dict*) –
If present, changes to the cluster are pending. Specific pending changes are identified by subelements.
 - **MasterUserPassword** (*string*) –
The pending or in-progress change of the master user password for the cluster.
 - **NodeType** (*string*) –
The pending or in-progress change of the cluster's node type.
 - **NumberOfNodes** (*integer*) –
The pending or in-progress change of the number of nodes in the cluster.
 - **ClusterType** (*string*) –
The pending or in-progress change of the cluster type.
 - **ClusterVersion** (*string*) –
The pending or in-progress change of the service version.
 - **AutomatedSnapshotRetentionPeriod** (*integer*) –
The pending or in-progress change of the automated snapshot retention period.
 - **ClusterIdentifier** (*string*) –
The pending or in-progress change of the new identifier for the cluster.
- * **ClusterVersion** (*string*) –
The version ID of the Amazon Redshift engine that is running on the cluster.
- * **AllowVersionUpgrade** (*boolean*) –
If `true`, major version upgrades will be applied automatically to the cluster during the maintenance window.
- * **NumberOfNodes** (*integer*) –
The number of compute nodes in the cluster.

* **PubliclyAccessible** (*boolean*) –

If `true`, the cluster can be accessed from a public network.

* **Encrypted** (*boolean*) –

If `true`, data in the cluster is encrypted at rest.

* **RestoreStatus** (*dict*) –

Describes the status of a cluster restore action. Returns null if the cluster was not created by restoring a snapshot.

· **Status** (*string*) –

The status of the restore action. Returns starting, restoring, completed, or failed.

· **CurrentRestoreRateInMegaBytesPerSecond** (*float*) –

The number of megabytes per second being transferred from the backup storage. Returns the average rate for a completed backup.

· **SnapshotSizeInMegaBytes** (*integer*) –

The size of the set of snapshot data used to restore the cluster.

· **ProgressInMegaBytes** (*integer*) –

The number of megabytes that have been transferred from snapshot storage.

· **ElapsedTimeInSeconds** (*integer*) –

The amount of time an in-progress restore has been running, or the amount of time it took a completed restore to finish.

· **EstimatedTimeToCompletionInSeconds** (*integer*) –

The estimate of the time remaining before the restore will complete. Returns 0 for a completed restore.

* **HsmStatus** (*dict*) –

Reports whether the Amazon Redshift cluster has finished applying any HSM settings changes specified in a modify cluster command.

Values: active, applying

· **HsmClientCertificateIdentifier** (*string*) –

Specifies the name of the HSM client certificate the Amazon Redshift cluster uses to retrieve the data encryption keys stored in an HSM.

· **HsmConfigurationIdentifier** (*string*) –

Specifies the name of the HSM configuration that contains the information the Amazon Redshift cluster can use to retrieve and store keys in an HSM.

· **Status** (*string*) –

Reports whether the Amazon Redshift cluster has finished applying any HSM settings changes specified in a modify cluster command.

Values: active, applying

* **ClusterSnapshotCopyStatus** (*dict*) –

Returns the destination region and retention period that are configured for cross-region snapshot copy.

- **DestinationRegion** (*string*) –
The destination region that snapshots are automatically copied to when cross-region snapshot copy is enabled.
- **RetentionPeriod** (*integer*) –
The number of days that automated snapshots are retained in the destination region after they are copied from a source region.
- **SnapshotCopyGrantName** (*string*) –
The name of the snapshot copy grant.
- * **ClusterPublicKey** (*string*) –
The public key for the cluster.
- * **ClusterNodes** (*list*) –
The nodes in a cluster.
 - (*dict*) –
The identifier of a node in a cluster.
 - **NodeRole** (*string*) –
Whether the node is a leader node or a compute node.
 - **PrivateIPAddress** (*string*) –
The private IP address of a node within a cluster.
 - **PublicIPAddress** (*string*) –
The public IP address of a node within a cluster.
- * **ElasticIpStatus** (*dict*) –
Describes the status of the elastic IP (EIP) address.
 - **ElasticIp** (*string*) –
The elastic IP (EIP) address for the cluster.
 - **Status** (*string*) –
Describes the status of the elastic IP (EIP) address.
- * **ClusterRevisionNumber** (*string*) –
The specific revision number of the database in the cluster.
- * **Tags** (*list*) –
The list of tags for the cluster.
 - (*dict*) –
A tag consisting of a name/value pair for a resource.
 - **Key** (*string*) –
The key, or name, for the resource tag.
 - **Value** (*string*) –
The value for the resource tag.
- * **KmsKeyId** (*string*) –
The AWS Key Management Service (KMS) key ID of the encryption key used to encrypt data in the cluster.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for

- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_waiter (*waiter_name*)

modify_cluster (***kwargs*)

Modifies the settings for a cluster. For example, you can add another security or parameter group, update the preferred maintenance window, or change the master user password. Resetting a cluster password or modifying the security groups associated with a cluster do not need a reboot. However, modifying a parameter group requires a reboot for parameters to take effect. For more information about managing clusters, go to [Amazon Redshift Clusters](#) in the *Amazon Redshift Cluster Management Guide*.

You can also change node type and the number of nodes to scale up or down the cluster. When resizing a cluster, you must specify both the number of nodes and the node type even if one of the parameters does not change.

Request Syntax

```
response = client.modify_cluster(
    ClusterIdentifier='string',
    ClusterType='string',
    NodeType='string',
    NumberOfNodes=123,
    ClusterSecurityGroups=[
        'string',
    ],
    VpcSecurityGroupIds=[
        'string',
    ],
    MasterUserPassword='string',
    ClusterParameterGroupName='string',
    AutomatedSnapshotRetentionPeriod=123,
    PreferredMaintenanceWindow='string',
    ClusterVersion='string',
    AllowVersionUpgrade=True|False,
    HsmClientCertificateIdentifier='string',
    HsmConfigurationIdentifier='string',
    NewClusterIdentifier='string'
)
```

Parameters

- **ClusterIdentifier** (*string*) – [REQUIRED]

The unique identifier of the cluster to be modified.

Example: `examplecluster`

- **ClusterType** (*string*) – The new cluster type.

When you submit your cluster resize request, your existing cluster goes into a read-only mode. After Amazon Redshift provisions a new cluster based on your resize requirements, there will be outage for a period while the old cluster is deleted and your connection is switched to the new cluster. You can use `DescribeResize` to track the progress of the resize request.

Valid Values: `multi-node` | `single-node`

- **NodeType** (*string*) – The new node type of the cluster. If you specify a new node type, you must also specify the number of nodes parameter.

When you submit your request to resize a cluster, Amazon Redshift sets access permissions for the cluster to read-only. After Amazon Redshift provisions a new cluster according to your resize requirements, there will be a temporary outage while the old cluster is deleted and your connection is switched to the new cluster. When the new connection is complete, the original access permissions for the cluster are restored. You can use `DescribeResize` to track the progress of the resize request.

Valid Values: `ds1.xlarge` | `ds1.8xlarge` | `ds2.xlarge` | `ds2.8xlarge` | `dc1.large` | `dc1.8xlarge`.

- **NumberOfNodes** (*integer*) – The new number of nodes of the cluster. If you specify a new number of nodes, you must also specify the node type parameter.

When you submit your request to resize a cluster, Amazon Redshift sets access permissions for the cluster to read-only. After Amazon Redshift provisions a new cluster according to your resize requirements, there will be a temporary outage while the old cluster is deleted and your connection is switched to the new cluster. When the new connection is complete, the original access permissions for the cluster are restored. You can use `DescribeResize` to track the progress of the resize request.

Valid Values: Integer greater than 0 .

- **ClusterSecurityGroups** (*list*) – A list of cluster security groups to be authorized on this cluster. This change is asynchronously applied as soon as possible.

Security groups currently associated with the cluster, and not in the list of groups to apply, will be revoked from the cluster.

Constraints:

- Must be 1 to 255 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

– (*string*) –

- **VpcSecurityGroupIds** (*list*) – A list of virtual private cloud (VPC) security groups to be associated with the cluster.

– (*string*) –

- **MasterUserPassword** (*string*) – The new password for the cluster master user. This change is asynchronously applied as soon as possible. Between the time of the request and the completion of the request, the `MasterUserPassword` element exists in the `PendingModifiedValues` element of the operation response.

Note: Operations never return the password, so this operation provides a way

to regain access to the master user account for a cluster if the password is lost.

Default: Uses existing setting.

Constraints:

- Must be between 8 and 64 characters in length.
 - Must contain at least one uppercase letter.
 - Must contain at least one lowercase letter.
 - Must contain one number.
 - Can be any printable ASCII character (ASCII code 33 to 126) except ‘ (single quote), ” (double quote), , /, @, or space.
- **ClusterParameterGroupName** (*string*) – The name of the cluster parameter group to apply to this cluster. This change is applied only after the cluster is rebooted. To reboot a cluster use `RebootCluster` .

Default: Uses existing setting.

Constraints: The cluster parameter group must be in the same parameter group family that matches the cluster version.

- **AutomatedSnapshotRetentionPeriod** (*integer*) – The number of days that automated snapshots are retained. If the value is 0, automated snapshots are disabled. Even if automated snapshots are disabled, you can still create manual snapshots when you want with `CreateClusterSnapshot` .

If you decrease the automated snapshot retention period from its current value, existing automated snapshots that fall outside of the new retention period will be immediately deleted.

Default: Uses existing setting.

Constraints: Must be a value from 0 to 35.

- **PreferredMaintenanceWindow** (*string*) – The weekly time range (in UTC) during which system maintenance can occur, if necessary. If system maintenance is necessary during the window, it may result in an outage.

This maintenance window change is made immediately. If the new maintenance window indicates the current time, there must be at least 120 minutes between the current time and end of the window in order to ensure that pending changes are applied.

Default: Uses existing setting.

Format: ddd:hh24:mi-ddd:hh24:mi, for example `wed:07:30-wed:08:00` .

Valid Days: Mon | Tue | Wed | Thu | Fri | Sat | Sun

Constraints: Must be at least 30 minutes.

- **ClusterVersion** (*string*) – The new version number of the Amazon Redshift engine to upgrade to.

For major version upgrades, if a non-default cluster parameter group is currently in use, a new cluster parameter group in the cluster parameter group family for the new version must be specified. The new cluster parameter group can be the default for that cluster parameter group family. For more information about parameters and parameter groups, go to [Amazon Redshift Parameter Groups](#) in the *Amazon Redshift Cluster Management Guide* .

Example: 1.0

- **AllowVersionUpgrade** (*boolean*) – If `true` , major version upgrades will be applied automatically to the cluster during the maintenance window.

Default: `false`

- **HsmClientCertificateIdentifier** (*string*) – Specifies the name of the HSM client certificate the Amazon Redshift cluster uses to retrieve the data encryption keys stored in an HSM.
- **HsmConfigurationIdentifier** (*string*) – Specifies the name of the HSM configuration that contains the information the Amazon Redshift cluster can use to retrieve and store keys in an HSM.
- **NewClusterIdentifier** (*string*) – The new identifier for the cluster.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens.
- Alphabetic characters must be lowercase.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.
- Must be unique for all clusters within an AWS account.

Example: `examplecluster`

Return type dict

Returns

Response Syntax

```
{
  'Cluster': {
    'ClusterIdentifier': 'string',
    'NodeType': 'string',
    'ClusterStatus': 'string',
    'ModifyStatus': 'string',
    'MasterUsername': 'string',
    'DBName': 'string',
    'Endpoint': {
      'Address': 'string',
      'Port': 123
    },
    'ClusterCreateTime': datetime(2015, 1, 1),
    'AutomatedSnapshotRetentionPeriod': 123,
    'ClusterSecurityGroups': [
      {
        'ClusterSecurityGroupName': 'string',
        'Status': 'string'
      },
    ],
    'VpcSecurityGroups': [
      {
        'VpcSecurityGroupId': 'string',
        'Status': 'string'
      },
    ],
    'ClusterParameterGroups': [
      {
        'ParameterGroupName': 'string',
        'ParameterApplyStatus': 'string',
        'ClusterParameterStatusList': [
          {
            'ParameterName': 'string',
            'ParameterApplyStatus': 'string',
            'ParameterApplyErrorDescription': 'string'
          },
        ],
      },
    ],
  },
}
```

```

    },
],
'ClusterSubnetGroupName': 'string',
'VpcId': 'string',
'AvailabilityZone': 'string',
'PreferredMaintenanceWindow': 'string',
'PendingModifiedValues': {
    'MasterUserPassword': 'string',
    'NodeType': 'string',
    'NumberOfNodes': 123,
    'ClusterType': 'string',
    'ClusterVersion': 'string',
    'AutomatedSnapshotRetentionPeriod': 123,
    'ClusterIdentifier': 'string'
},
'ClusterVersion': 'string',
'AllowVersionUpgrade': True|False,
'NumberOfNodes': 123,
'PubliclyAccessible': True|False,
'Encrypted': True|False,
'RestoreStatus': {
    'Status': 'string',
    'CurrentRestoreRateInMegaBytesPerSecond': 123.0,
    'SnapshotSizeInMegaBytes': 123,
    'ProgressInMegaBytes': 123,
    'ElapsedTimeInSeconds': 123,
    'EstimatedTimeToCompletionInSeconds': 123
},
'HsmStatus': {
    'HsmClientCertificateIdentifier': 'string',
    'HsmConfigurationIdentifier': 'string',
    'Status': 'string'
},
'ClusterSnapshotCopyStatus': {
    'DestinationRegion': 'string',
    'RetentionPeriod': 123,
    'SnapshotCopyGrantName': 'string'
},
'ClusterPublicKey': 'string',
'ClusterNodes': [
    {
        'NodeRole': 'string',
        'PrivateIpAddress': 'string',
        'PublicIpAddress': 'string'
    },
],
'ElasticIpStatus': {
    'ElasticIp': 'string',
    'Status': 'string'
},
'ClusterRevisionNumber': 'string',
'Tags': [
    {
        'Key': 'string',
        'Value': 'string'
    },
],
'KmsKeyId': 'string'

```

```
}  
}
```

Response Structure

- (*dict*) –

Describes a cluster.

- **Cluster** (*dict*) –

Describes a cluster.

- * **ClusterIdentifier** (*string*) –

The unique identifier of the cluster.

- * **NodeType** (*string*) –

The node type for the nodes in the cluster.

- * **ClusterStatus** (*string*) –

The current state of this cluster. Possible values include `available`, `creating`, `deleting`, `rebooting`, `renaming`, and `resizing`.

- * **ModifyStatus** (*string*) –

The status of a modify operation, if any, initiated for the cluster.

- * **MasterUsername** (*string*) –

The master user name for the cluster. This name is used to connect to the database that is specified in **DBName**.

- * **DBName** (*string*) –

The name of the initial database that was created when the cluster was created. This same name is returned for the life of the cluster. If an initial database was not specified, a database named “dev” was created by default.

- * **Endpoint** (*dict*) –

The connection endpoint.

- **Address** (*string*) –

The DNS address of the Cluster.

- **Port** (*integer*) –

The port that the database engine is listening on.

- * **ClusterCreateTime** (*datetime*) –

The date and time that the cluster was created.

- * **AutomatedSnapshotRetentionPeriod** (*integer*) –

The number of days that automatic cluster snapshots are retained.

- * **ClusterSecurityGroups** (*list*) –

A list of cluster security group that are associated with the cluster. Each security group is represented by an element that contains `ClusterSecurityGroup.Name` and `ClusterSecurityGroup.Status` subelements.

Cluster security groups are used when the cluster is not created in a VPC. Clusters that are created in a VPC use VPC security groups, which are listed by the **VpcSecurityGroups** parameter.

- (*dict*) –

Describes a security group.

- **ClusterSecurityGroupName** (*string*) –
The name of the cluster security group.
- **Status** (*string*) –
The status of the cluster security group.
- * **VpcSecurityGroups** (*list*) –
A list of Virtual Private Cloud (VPC) security groups that are associated with the cluster. This parameter is returned only if the cluster is in a VPC.
 - (*dict*) –
Describes the members of a VPC security group.
 - **VpcSecurityGroupId** (*string*) –
 - **Status** (*string*) –
- * **ClusterParameterGroups** (*list*) –
The list of cluster parameter groups that are associated with this cluster. Each parameter group in the list is returned with its status.
 - (*dict*) –
Describes the status of a parameter group.
 - **ParameterGroupName** (*string*) –
The name of the cluster parameter group.
 - **ParameterApplyStatus** (*string*) –
The status of parameter updates.
 - **ClusterParameterStatusList** (*list*) –
The list of parameter statuses.

For more information about parameters and parameter groups, go to [Amazon Redshift Parameter Groups](#) in the *Amazon Redshift Cluster Management Guide*.
 - (*dict*) –
Describes the status of a parameter group.
 - **ParameterName** (*string*) –
The name of the parameter.
 - **ParameterApplyStatus** (*string*) –
The status of the parameter that indicates whether the parameter is in sync with the database, waiting for a cluster reboot, or encountered an error when being applied.

The following are possible statuses and descriptions.
 - **in-sync** : The parameter value is in sync with the database.
 - **pending-reboot** : The parameter value will be applied after the cluster reboots.
 - **applying** : The parameter value is being applied to the database.
 - **invalid-parameter** : Cannot apply the parameter value because it has an invalid value or syntax.
 - **apply-deferred** : The parameter contains static property changes. The changes are deferred until the cluster reboots.
 - **apply-error** : Cannot connect to the cluster. The parameter change will be applied after the cluster reboots.

- **unknown-error** : Cannot apply the parameter change right now. The change will be applied after the cluster reboots.
- **ParameterApplyErrorDescription** (*string*) –
The error that prevented the parameter from being applied to the database.
- * **ClusterSubnetGroupName** (*string*) –
The name of the subnet group that is associated with the cluster. This parameter is valid only when the cluster is in a VPC.
- * **VpcId** (*string*) –
The identifier of the VPC the cluster is in, if the cluster is in a VPC.
- * **AvailabilityZone** (*string*) –
The name of the Availability Zone in which the cluster is located.
- * **PreferredMaintenanceWindow** (*string*) –
The weekly time range (in UTC) during which system maintenance can occur.
- * **PendingModifiedValues** (*dict*) –
If present, changes to the cluster are pending. Specific pending changes are identified by subelements.
 - **MasterUserPassword** (*string*) –
The pending or in-progress change of the master user password for the cluster.
 - **NodeType** (*string*) –
The pending or in-progress change of the cluster's node type.
 - **NumberOfNodes** (*integer*) –
The pending or in-progress change of the number of nodes in the cluster.
 - **ClusterType** (*string*) –
The pending or in-progress change of the cluster type.
 - **ClusterVersion** (*string*) –
The pending or in-progress change of the service version.
 - **AutomatedSnapshotRetentionPeriod** (*integer*) –
The pending or in-progress change of the automated snapshot retention period.
 - **ClusterIdentifier** (*string*) –
The pending or in-progress change of the new identifier for the cluster.
- * **ClusterVersion** (*string*) –
The version ID of the Amazon Redshift engine that is running on the cluster.
- * **AllowVersionUpgrade** (*boolean*) –
If `true`, major version upgrades will be applied automatically to the cluster during the maintenance window.
- * **NumberOfNodes** (*integer*) –
The number of compute nodes in the cluster.

* **PubliclyAccessible** (*boolean*) –

If `true`, the cluster can be accessed from a public network.

* **Encrypted** (*boolean*) –

If `true`, data in the cluster is encrypted at rest.

* **RestoreStatus** (*dict*) –

Describes the status of a cluster restore action. Returns null if the cluster was not created by restoring a snapshot.

· **Status** (*string*) –

The status of the restore action. Returns starting, restoring, completed, or failed.

· **CurrentRestoreRateInMegaBytesPerSecond** (*float*) –

The number of megabytes per second being transferred from the backup storage. Returns the average rate for a completed backup.

· **SnapshotSizeInMegaBytes** (*integer*) –

The size of the set of snapshot data used to restore the cluster.

· **ProgressInMegaBytes** (*integer*) –

The number of megabytes that have been transferred from snapshot storage.

· **ElapsedTimeInSeconds** (*integer*) –

The amount of time an in-progress restore has been running, or the amount of time it took a completed restore to finish.

· **EstimatedTimeToCompletionInSeconds** (*integer*) –

The estimate of the time remaining before the restore will complete. Returns 0 for a completed restore.

* **HsmStatus** (*dict*) –

Reports whether the Amazon Redshift cluster has finished applying any HSM settings changes specified in a modify cluster command.

Values: active, applying

· **HsmClientCertificateIdentifier** (*string*) –

Specifies the name of the HSM client certificate the Amazon Redshift cluster uses to retrieve the data encryption keys stored in an HSM.

· **HsmConfigurationIdentifier** (*string*) –

Specifies the name of the HSM configuration that contains the information the Amazon Redshift cluster can use to retrieve and store keys in an HSM.

· **Status** (*string*) –

Reports whether the Amazon Redshift cluster has finished applying any HSM settings changes specified in a modify cluster command.

Values: active, applying

* **ClusterSnapshotCopyStatus** (*dict*) –

Returns the destination region and retention period that are configured for cross-region snapshot copy.

- **DestinationRegion** (*string*) –
The destination region that snapshots are automatically copied to when cross-region snapshot copy is enabled.
- **RetentionPeriod** (*integer*) –
The number of days that automated snapshots are retained in the destination region after they are copied from a source region.
- **SnapshotCopyGrantName** (*string*) –
The name of the snapshot copy grant.
- * **ClusterPublicKey** (*string*) –
The public key for the cluster.
- * **ClusterNodes** (*list*) –
The nodes in a cluster.
 - (*dict*) –
The identifier of a node in a cluster.
 - **NodeRole** (*string*) –
Whether the node is a leader node or a compute node.
 - **PrivateIPAddress** (*string*) –
The private IP address of a node within a cluster.
 - **PublicIPAddress** (*string*) –
The public IP address of a node within a cluster.
- * **ElasticIpStatus** (*dict*) –
Describes the status of the elastic IP (EIP) address.
 - **ElasticIp** (*string*) –
The elastic IP (EIP) address for the cluster.
 - **Status** (*string*) –
Describes the status of the elastic IP (EIP) address.
- * **ClusterRevisionNumber** (*string*) –
The specific revision number of the database in the cluster.
- * **Tags** (*list*) –
The list of tags for the cluster.
 - (*dict*) –
A tag consisting of a name/value pair for a resource.
 - **Key** (*string*) –
The key, or name, for the resource tag.
 - **Value** (*string*) –
The value for the resource tag.
- * **KmsKeyId** (*string*) –
The AWS Key Management Service (KMS) key ID of the encryption key used to encrypt data in the cluster.

modify_cluster_parameter_group (***kwargs*)

Modifies the parameters of a parameter group.

For more information about parameters and parameter groups, go to [Amazon Redshift Parameter Groups](#) in the *Amazon Redshift Cluster Management Guide* .

Request Syntax

```
response = client.modify_cluster_parameter_group(
    ParameterGroupName='string',
    Parameters=[
        {
            'ParameterName': 'string',
            'ParameterValue': 'string',
            'Description': 'string',
            'Source': 'string',
            'DataType': 'string',
            'AllowedValues': 'string',
            'ApplyType': 'static'|'dynamic',
            'IsModifiable': True|False,
            'MinimumEngineVersion': 'string'
        },
    ]
)
```

Parameters

- **ParameterGroupName** (*string*) – [REQUIRED]

The name of the parameter group to be modified.

- **Parameters** (*list*) – [REQUIRED]

An array of parameters to be modified. A maximum of 20 parameters can be modified in a single request.

For each parameter to be modified, you must supply at least the parameter name and parameter value; other name-value pairs of the parameter are optional.

For the workload management (WLM) configuration, you must supply all the name-value pairs in the `wlm_json_configuration` parameter.

– (*dict*) –

Describes a parameter in a cluster parameter group.

- * **ParameterName** (*string*) –

The name of the parameter.

- * **ParameterValue** (*string*) –

The value of the parameter.

- * **Description** (*string*) –

A description of the parameter.

- * **Source** (*string*) –

The source of the parameter value, such as “engine-default” or “user”.

- * **DataType** (*string*) –

The data type of the parameter.

- * **AllowedValues** (*string*) –

The valid range of values for the parameter.

- * **ApplyType** (*string*) –

Specifies how to apply the parameter. Supported value: `static`.

- * **IsModifiable** (*boolean*) –

If `true`, the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.

* **MinimumEngineVersion** (*string*) –

The earliest engine version to which the parameter can apply.

Return type dict

Returns

Response Syntax

```
{
    'ParameterGroupName': 'string',
    'ParameterGroupStatus': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output from the `ModifyClusterParameterGroup` and `ResetClusterParameterGroup` actions and indicate the parameter group involved and the status of the operation on the parameter group.

– **ParameterGroupName** (*string*) –

The name of the cluster parameter group.

– **ParameterGroupStatus** (*string*) –

The status of the parameter group. For example, if you made a change to a parameter group name-value pair, then the change could be pending a reboot of an associated cluster.

modify_cluster_subnet_group (***kwargs*)

Modifies a cluster subnet group to include the specified list of VPC subnets. The operation replaces the existing list of subnets with the new list of subnets.

Request Syntax

```
response = client.modify_cluster_subnet_group(
    ClusterSubnetGroupName='string',
    Description='string',
    SubnetIds=[
        'string',
    ]
)
```

Parameters

- **ClusterSubnetGroupName** (*string*) – [REQUIRED]

The name of the subnet group to be modified.

- **Description** (*string*) – A text description of the subnet group to be modified.
- **SubnetIds** (*list*) – [REQUIRED]

An array of VPC subnet IDs. A maximum of 20 subnets can be modified in a single request.

– (*string*) –

Return type dict

Returns

Response Syntax

```

{
  'ClusterSubnetGroup': {
    'ClusterSubnetGroupName': 'string',
    'Description': 'string',
    'VpcId': 'string',
    'SubnetGroupStatus': 'string',
    'Subnets': [
      {
        'SubnetIdentifier': 'string',
        'SubnetAvailabilityZone': {
          'Name': 'string'
        },
        'SubnetStatus': 'string'
      },
    ],
    'Tags': [
      {
        'Key': 'string',
        'Value': 'string'
      },
    ],
  ]
}

```

Response Structure

- (*dict*) –

Describes a subnet group.

- **ClusterSubnetGroup** (*dict*) –

Describes a subnet group.

- * **ClusterSubnetGroupName** (*string*) –

The name of the cluster subnet group.

- * **Description** (*string*) –

The description of the cluster subnet group.

- * **VpcId** (*string*) –

The VPC ID of the cluster subnet group.

- * **SubnetGroupStatus** (*string*) –

The status of the cluster subnet group. Possible values are Complete, Incomplete and Invalid.

- * **Subnets** (*list*) –

A list of the VPC Subnet elements.

- (*dict*) –

Describes a subnet.

- **SubnetIdentifier** (*string*) –

The identifier of the subnet.

- **SubnetAvailabilityZone** (*dict*) –

Describes an availability zone.

- **Name** (*string*) –

The name of the availability zone.

- **SubnetStatus** (*string*) –

The status of the subnet.

* **Tags** (*list*) –

The list of tags for the cluster subnet group.

· (*dict*) –

A tag consisting of a name/value pair for a resource.

· **Key** (*string*) –

The key, or name, for the resource tag.

· **Value** (*string*) –

The value for the resource tag.

modify_event_subscription (**kwargs)

Modifies an existing Amazon Redshift event notification subscription.

Request Syntax

```
response = client.modify_event_subscription(
    SubscriptionName='string',
    SnsTopicArn='string',
    SourceType='string',
    SourceIds=[
        'string',
    ],
    EventCategories=[
        'string',
    ],
    Severity='string',
    Enabled=True|False
)
```

Parameters

- **SubscriptionName** (*string*) – [REQUIRED]

The name of the modified Amazon Redshift event notification subscription.

- **SnsTopicArn** (*string*) – The Amazon Resource Name (ARN) of the SNS topic to be used by the event notification subscription.
- **SourceType** (*string*) – The type of source that will be generating the events. For example, if you want to be notified of events generated by a cluster, you would set this parameter to cluster. If this value is not specified, events are returned for all Amazon Redshift objects in your AWS account. You must specify a source type in order to specify source IDs.

Valid values: cluster, cluster-parameter-group, cluster-security-group, and cluster-snapshot.

- **SourceIds** (*list*) – A list of one or more identifiers of Amazon Redshift source objects. All of the objects must be of the same type as was specified in the source type parameter. The event subscription will return only events generated by the specified objects. If not specified, then events are returned for all objects within the source type specified.

Example: my-cluster-1, my-cluster-2

Example: my-snapshot-20131010

– (*string*) –

- **EventCategories** (*list*) – Specifies the Amazon Redshift event categories to be published by the event notification subscription.

Values: Configuration, Management, Monitoring, Security

- (*string*) –
- **Severity** (*string*) – Specifies the Amazon Redshift event severity to be published by the event notification subscription.

Values: ERROR, INFO

- **Enabled** (*boolean*) – A Boolean value indicating if the subscription is enabled.
true indicates the subscription is enabled

Return type dict

Returns

Response Syntax

```
{
  'EventSubscription': {
    'CustomerAwsId': 'string',
    'CustSubscriptionId': 'string',
    'SnsTopicArn': 'string',
    'Status': 'string',
    'SubscriptionCreationTime': datetime(2015, 1, 1),
    'SourceType': 'string',
    'SourceIdsList': [
      'string',
    ],
    'EventCategoriesList': [
      'string',
    ],
    'Severity': 'string',
    'Enabled': True|False,
    'Tags': [
      {
        'Key': 'string',
        'Value': 'string'
      },
    ]
  }
}
```

Response Structure

- (*dict*) –
 - **EventSubscription** (*dict*) –
 - * **CustomerAwsId** (*string*) –
The AWS customer account associated with the Amazon Redshift event notification subscription.
 - * **CustSubscriptionId** (*string*) –
The name of the Amazon Redshift event notification subscription.
 - * **SnsTopicArn** (*string*) –
The Amazon Resource Name (ARN) of the Amazon SNS topic used by the event notification subscription.
 - * **Status** (*string*) –
The status of the Amazon Redshift event notification subscription.
- Constraints:
- Can be one of the following: active | no-permission | topic-not-exist

- The status “no-permission” indicates that Amazon Redshift no longer has permission to post to the Amazon SNS topic. The status “topic-not-exist” indicates that the topic was deleted after the subscription was created.

* **SubscriptionCreationTime** (*datetime*) –

The date and time the Amazon Redshift event notification subscription was created.

* **SourceType** (*string*) –

The source type of the events returned the Amazon Redshift event notification, such as cluster, or cluster-snapshot.

* **SourceIdsList** (*list*) –

A list of the sources that publish events to the Amazon Redshift event notification subscription.

- (*string*) –

* **EventCategoriesList** (*list*) –

The list of Amazon Redshift event categories specified in the event notification subscription.

Values: Configuration, Management, Monitoring, Security

- (*string*) –

* **Severity** (*string*) –

The event severity specified in the Amazon Redshift event notification subscription.

Values: ERROR, INFO

* **Enabled** (*boolean*) –

A Boolean value indicating whether the subscription is enabled. `true` indicates the subscription is enabled.

* **Tags** (*list*) –

The list of tags for the event subscription.

- (*dict*) –

A tag consisting of a name/value pair for a resource.

- **Key** (*string*) –

The key, or name, for the resource tag.

- **Value** (*string*) –

The value for the resource tag.

modify_snapshot_copy_retention_period (**kwargs)

Modifies the number of days to retain automated snapshots in the destination region after they are copied from the source region.

Request Syntax

```
response = client.modify_snapshot_copy_retention_period(
    ClusterIdentifier='string',
    RetentionPeriod=123
)
```

Parameters

- **ClusterIdentifier** (*string*) – [REQUIRED]

The unique identifier of the cluster for which you want to change the retention period for automated snapshots that are copied to a destination region.

Constraints: Must be the valid name of an existing cluster that has cross-region snapshot copy enabled.

- **RetentionPeriod** (*integer*) – [REQUIRED]

The number of days to retain automated snapshots in the destination region after they are copied from the source region.

If you decrease the retention period for automated snapshots that are copied to a destination region, Amazon Redshift will delete any existing automated snapshots that were copied to the destination region and that fall outside of the new retention period.

Constraints: Must be at least 1 and no more than 35.

Return type dict

Returns

Response Syntax

```
{
  'Cluster': {
    'ClusterIdentifier': 'string',
    'NodeType': 'string',
    'ClusterStatus': 'string',
    'ModifyStatus': 'string',
    'MasterUsername': 'string',
    'DBName': 'string',
    'Endpoint': {
      'Address': 'string',
      'Port': 123
    },
  },
  'ClusterCreateTime': datetime(2015, 1, 1),
  'AutomatedSnapshotRetentionPeriod': 123,
  'ClusterSecurityGroups': [
    {
      'ClusterSecurityGroupName': 'string',
      'Status': 'string'
    },
  ],
  'VpcSecurityGroups': [
    {
      'VpcSecurityGroupId': 'string',
      'Status': 'string'
    },
  ],
  'ClusterParameterGroups': [
    {
      'ParameterGroupName': 'string',
      'ParameterApplyStatus': 'string',
      'ClusterParameterStatusList': [
        {
          'ParameterName': 'string',
          'ParameterApplyStatus': 'string',
          'ParameterApplyErrorDescription': 'string'
        },
      ],
    },
  ],
}
```

```
],
'ClusterSubnetGroupName': 'string',
'VpcId': 'string',
'AvailabilityZone': 'string',
'PreferredMaintenanceWindow': 'string',
'PendingModifiedValues': {
    'MasterUserPassword': 'string',
    'NodeType': 'string',
    'NumberOfNodes': 123,
    'ClusterType': 'string',
    'ClusterVersion': 'string',
    'AutomatedSnapshotRetentionPeriod': 123,
    'ClusterIdentifier': 'string'
},
'ClusterVersion': 'string',
'AllowVersionUpgrade': True|False,
'NumberOfNodes': 123,
'PubliclyAccessible': True|False,
'Encrypted': True|False,
'RestoreStatus': {
    'Status': 'string',
    'CurrentRestoreRateInMegaBytesPerSecond': 123.0,
    'SnapshotSizeInMegaBytes': 123,
    'ProgressInMegaBytes': 123,
    'ElapsedTimeInSeconds': 123,
    'EstimatedTimeToCompletionInSeconds': 123
},
'HsmStatus': {
    'HsmClientCertificateIdentifier': 'string',
    'HsmConfigurationIdentifier': 'string',
    'Status': 'string'
},
'ClusterSnapshotCopyStatus': {
    'DestinationRegion': 'string',
    'RetentionPeriod': 123,
    'SnapshotCopyGrantName': 'string'
},
'ClusterPublicKey': 'string',
'ClusterNodes': [
    {
        'NodeRole': 'string',
        'PrivateIpAddress': 'string',
        'PublicIpAddress': 'string'
    },
],
'ElasticIpStatus': {
    'ElasticIp': 'string',
    'Status': 'string'
},
'ClusterRevisionNumber': 'string',
'Tags': [
    {
        'Key': 'string',
        'Value': 'string'
    },
],
'KmsKeyId': 'string'
}
```



```
}

```

Response Structure

- (*dict*) –

Describes a cluster.

- **Cluster** (*dict*) –

Describes a cluster.

- * **ClusterIdentifier** (*string*) –

The unique identifier of the cluster.

- * **NodeType** (*string*) –

The node type for the nodes in the cluster.

- * **ClusterStatus** (*string*) –

The current state of this cluster. Possible values include `available`, `creating`, `deleting`, `rebooting`, `renaming`, and `resizing`.

- * **ModifyStatus** (*string*) –

The status of a modify operation, if any, initiated for the cluster.

- * **MasterUsername** (*string*) –

The master user name for the cluster. This name is used to connect to the database that is specified in **DBName**.

- * **DBName** (*string*) –

The name of the initial database that was created when the cluster was created. This same name is returned for the life of the cluster. If an initial database was not specified, a database named “dev” was created by default.

- * **Endpoint** (*dict*) –

The connection endpoint.

- **Address** (*string*) –

The DNS address of the Cluster.

- **Port** (*integer*) –

The port that the database engine is listening on.

- * **ClusterCreateTime** (*datetime*) –

The date and time that the cluster was created.

- * **AutomatedSnapshotRetentionPeriod** (*integer*) –

The number of days that automatic cluster snapshots are retained.

- * **ClusterSecurityGroups** (*list*) –

A list of cluster security group that are associated with the cluster. Each security group is represented by an element that contains `ClusterSecurityGroup.Name` and `ClusterSecurityGroup.Status` subelements.

Cluster security groups are used when the cluster is not created in a VPC. Clusters that are created in a VPC use VPC security groups, which are listed by the **VpcSecurityGroups** parameter.

- (*dict*) –

Describes a security group.

- **ClusterSecurityGroupName** (*string*) –
The name of the cluster security group.
- **Status** (*string*) –
The status of the cluster security group.
- * **VpcSecurityGroups** (*list*) –
A list of Virtual Private Cloud (VPC) security groups that are associated with the cluster. This parameter is returned only if the cluster is in a VPC.
 - (*dict*) –
Describes the members of a VPC security group.
 - **VpcSecurityGroupId** (*string*) –
 - **Status** (*string*) –
- * **ClusterParameterGroups** (*list*) –
The list of cluster parameter groups that are associated with this cluster. Each parameter group in the list is returned with its status.
 - (*dict*) –
Describes the status of a parameter group.
 - **ParameterGroupName** (*string*) –
The name of the cluster parameter group.
 - **ParameterApplyStatus** (*string*) –
The status of parameter updates.
 - **ClusterParameterStatusList** (*list*) –
The list of parameter statuses.

For more information about parameters and parameter groups, go to [Amazon Redshift Parameter Groups](#) in the *Amazon Redshift Cluster Management Guide* .
 - (*dict*) –
Describes the status of a parameter group.
 - **ParameterName** (*string*) –
The name of the parameter.
 - **ParameterApplyStatus** (*string*) –
The status of the parameter that indicates whether the parameter is in sync with the database, waiting for a cluster reboot, or encountered an error when being applied.

The following are possible statuses and descriptions.
 - **in-sync** : The parameter value is in sync with the database.
 - **pending-reboot** : The parameter value will be applied after the cluster reboots.
 - **applying** : The parameter value is being applied to the database.
 - **invalid-parameter** : Cannot apply the parameter value because it has an invalid value or syntax.
 - **apply-deferred** : The parameter contains static property changes. The changes are deferred until the cluster reboots.
 - **apply-error** : Cannot connect to the cluster. The parameter change will be applied after the cluster reboots.

- `unknown-error` : Cannot apply the parameter change right now. The change will be applied after the cluster reboots.
- **ParameterApplyErrorDescription** (*string*) –
The error that prevented the parameter from being applied to the database.
- * **ClusterSubnetGroupName** (*string*) –
The name of the subnet group that is associated with the cluster. This parameter is valid only when the cluster is in a VPC.
- * **VpcId** (*string*) –
The identifier of the VPC the cluster is in, if the cluster is in a VPC.
- * **AvailabilityZone** (*string*) –
The name of the Availability Zone in which the cluster is located.
- * **PreferredMaintenanceWindow** (*string*) –
The weekly time range (in UTC) during which system maintenance can occur.
- * **PendingModifiedValues** (*dict*) –
If present, changes to the cluster are pending. Specific pending changes are identified by subelements.
 - **MasterUserPassword** (*string*) –
The pending or in-progress change of the master user password for the cluster.
 - **NodeType** (*string*) –
The pending or in-progress change of the cluster's node type.
 - **NumberOfNodes** (*integer*) –
The pending or in-progress change of the number of nodes in the cluster.
 - **ClusterType** (*string*) –
The pending or in-progress change of the cluster type.
 - **ClusterVersion** (*string*) –
The pending or in-progress change of the service version.
 - **AutomatedSnapshotRetentionPeriod** (*integer*) –
The pending or in-progress change of the automated snapshot retention period.
 - **ClusterIdentifier** (*string*) –
The pending or in-progress change of the new identifier for the cluster.
- * **ClusterVersion** (*string*) –
The version ID of the Amazon Redshift engine that is running on the cluster.
- * **AllowVersionUpgrade** (*boolean*) –
If `true`, major version upgrades will be applied automatically to the cluster during the maintenance window.
- * **NumberOfNodes** (*integer*) –
The number of compute nodes in the cluster.

* **PubliclyAccessible** (*boolean*) –

If `true`, the cluster can be accessed from a public network.

* **Encrypted** (*boolean*) –

If `true`, data in the cluster is encrypted at rest.

* **RestoreStatus** (*dict*) –

Describes the status of a cluster restore action. Returns null if the cluster was not created by restoring a snapshot.

· **Status** (*string*) –

The status of the restore action. Returns starting, restoring, completed, or failed.

· **CurrentRestoreRateInMegaBytesPerSecond** (*float*) –

The number of megabytes per second being transferred from the backup storage. Returns the average rate for a completed backup.

· **SnapshotSizeInMegaBytes** (*integer*) –

The size of the set of snapshot data used to restore the cluster.

· **ProgressInMegaBytes** (*integer*) –

The number of megabytes that have been transferred from snapshot storage.

· **ElapsedTimeInSeconds** (*integer*) –

The amount of time an in-progress restore has been running, or the amount of time it took a completed restore to finish.

· **EstimatedTimeToCompletionInSeconds** (*integer*) –

The estimate of the time remaining before the restore will complete. Returns 0 for a completed restore.

* **HsmStatus** (*dict*) –

Reports whether the Amazon Redshift cluster has finished applying any HSM settings changes specified in a modify cluster command.

Values: active, applying

· **HsmClientCertificateIdentifier** (*string*) –

Specifies the name of the HSM client certificate the Amazon Redshift cluster uses to retrieve the data encryption keys stored in an HSM.

· **HsmConfigurationIdentifier** (*string*) –

Specifies the name of the HSM configuration that contains the information the Amazon Redshift cluster can use to retrieve and store keys in an HSM.

· **Status** (*string*) –

Reports whether the Amazon Redshift cluster has finished applying any HSM settings changes specified in a modify cluster command.

Values: active, applying

* **ClusterSnapshotCopyStatus** (*dict*) –

Returns the destination region and retention period that are configured for cross-region snapshot copy.

- **DestinationRegion** (*string*) –
The destination region that snapshots are automatically copied to when cross-region snapshot copy is enabled.
- **RetentionPeriod** (*integer*) –
The number of days that automated snapshots are retained in the destination region after they are copied from a source region.
- **SnapshotCopyGrantName** (*string*) –
The name of the snapshot copy grant.
- * **ClusterPublicKey** (*string*) –
The public key for the cluster.
- * **ClusterNodes** (*list*) –
The nodes in a cluster.
 - (*dict*) –
The identifier of a node in a cluster.
 - **NodeRole** (*string*) –
Whether the node is a leader node or a compute node.
 - **PrivateIPAddress** (*string*) –
The private IP address of a node within a cluster.
 - **PublicIPAddress** (*string*) –
The public IP address of a node within a cluster.
- * **ElasticIpStatus** (*dict*) –
Describes the status of the elastic IP (EIP) address.
 - **ElasticIp** (*string*) –
The elastic IP (EIP) address for the cluster.
 - **Status** (*string*) –
Describes the status of the elastic IP (EIP) address.
- * **ClusterRevisionNumber** (*string*) –
The specific revision number of the database in the cluster.
- * **Tags** (*list*) –
The list of tags for the cluster.
 - (*dict*) –
A tag consisting of a name/value pair for a resource.
 - **Key** (*string*) –
The key, or name, for the resource tag.
 - **Value** (*string*) –
The value for the resource tag.
- * **KmsKeyId** (*string*) –
The AWS Key Management Service (KMS) key ID of the encryption key used to encrypt data in the cluster.

purchase_reserved_node_offering (***kwargs*)

Allows you to purchase reserved nodes. Amazon Redshift offers a predefined set of reserved node offerings. You can purchase one or more of the offerings. You can call the `DescribeReservedNodeOfferings`

API to obtain the available reserved node offerings. You can call this API by providing a specific reserved node offering and the number of nodes you want to reserve.

For more information about reserved node offerings, go to [Purchasing Reserved Nodes](#) in the *Amazon Redshift Cluster Management Guide*.

Request Syntax

```
response = client.purchase_reserved_node_offering(
    ReservedNodeOfferingId='string',
    NodeCount=123
)
```

Parameters

- **ReservedNodeOfferingId** (*string*) – **[REQUIRED]**

The unique identifier of the reserved node offering you want to purchase.

- **NodeCount** (*integer*) – The number of reserved nodes you want to purchase.

Default: 1

Return type dict

Returns

Response Syntax

```
{
  'ReservedNode': {
    'ReservedNodeId': 'string',
    'ReservedNodeOfferingId': 'string',
    'NodeType': 'string',
    'StartTime': datetime(2015, 1, 1),
    'Duration': 123,
    'FixedPrice': 123.0,
    'UsagePrice': 123.0,
    'CurrencyCode': 'string',
    'NodeCount': 123,
    'State': 'string',
    'OfferingType': 'string',
    'RecurringCharges': [
      {
        'RecurringChargeAmount': 123.0,
        'RecurringChargeFrequency': 'string'
      }
    ]
  }
}
```

Response Structure

- (*dict*) –

Describes a reserved node. You can call the DescribeReservedNodeOfferings API to obtain the available reserved node offerings.

- **ReservedNode** (*dict*) –

Describes a reserved node. You can call the DescribeReservedNodeOfferings API to obtain the available reserved node offerings.

- * **ReservedNodeId** (*string*) –

The unique identifier for the reservation.

- * **ReservedNodeOfferingId** (*string*) –
The identifier for the reserved node offering.
- * **NodeType** (*string*) –
The node type of the reserved node.
- * **StartTime** (*datetime*) –
The time the reservation started. You purchase a reserved node offering for a duration. This is the start time of that duration.
- * **Duration** (*integer*) –
The duration of the node reservation in seconds.
- * **FixedPrice** (*float*) –
The fixed cost Amazon Redshift charges you for this reserved node.
- * **UsagePrice** (*float*) –
The hourly rate Amazon Redshift charges you for this reserved node.
- * **CurrencyCode** (*string*) –
The currency code for the reserved cluster.
- * **NodeCount** (*integer*) –
The number of reserved compute nodes.
- * **State** (*string*) –
The state of the reserved compute node.

Possible Values:
 - pending-payment-This reserved node has recently been purchased, and the sale has been approved, but payment has not yet been confirmed.
 - active-This reserved node is owned by the caller and is available for use.
 - payment-failed-Payment failed for the purchase attempt.
- * **OfferingType** (*string*) –
The anticipated utilization of the reserved node, as defined in the reserved node offering.
- * **RecurringCharges** (*list*) –
The recurring charges for the reserved node.
 - (*dict*) –
Describes a recurring charge.
 - **RecurringChargeAmount** (*float*) –
The amount charged per the period of time specified by the recurring charge frequency.
 - **RecurringChargeFrequency** (*string*) –
The frequency at which the recurring charge amount is applied.

reboot_cluster (***kwargs*)

Reboots a cluster. This action is taken as soon as possible. It results in a momentary outage to the cluster, during which the cluster status is set to `rebooting`. A cluster event is created when the reboot is completed. Any pending cluster modifications (see `ModifyCluster`) are applied at this reboot. For more information about managing clusters, go to [Amazon Redshift Clusters](#) in the *Amazon Redshift Cluster Management Guide*

Request Syntax

```
response = client.reboot_cluster(  
    ClusterIdentifier='string'  
)
```

Parameters `ClusterIdentifier` (*string*) – [REQUIRED]

The cluster identifier.

Return type dict

Returns

Response Syntax

```
{  
    'Cluster': {  
        'ClusterIdentifier': 'string',  
        'NodeType': 'string',  
        'ClusterStatus': 'string',  
        'ModifyStatus': 'string',  
        'MasterUsername': 'string',  
        'DBName': 'string',  
        'Endpoint': {  
            'Address': 'string',  
            'Port': 123  
        },  
        'ClusterCreateTime': datetime(2015, 1, 1),  
        'AutomatedSnapshotRetentionPeriod': 123,  
        'ClusterSecurityGroups': [  
            {  
                'ClusterSecurityGroupName': 'string',  
                'Status': 'string'  
            },  
        ],  
        'VpcSecurityGroups': [  
            {  
                'VpcSecurityGroupId': 'string',  
                'Status': 'string'  
            },  
        ],  
        'ClusterParameterGroups': [  
            {  
                'ParameterGroupName': 'string',  
                'ParameterApplyStatus': 'string',  
                'ClusterParameterStatusList': [  
                    {  
                        'ParameterName': 'string',  
                        'ParameterApplyStatus': 'string',  
                        'ParameterApplyErrorDescription': 'string'  
                    },  
                ],  
            },  
        ],  
        'ClusterSubnetGroupName': 'string',  
        'VpcId': 'string',  
        'AvailabilityZone': 'string',  
        'PreferredMaintenanceWindow': 'string',  
        'PendingModifiedValues': {  
            'MasterUserPassword': 'string',
```



```

        'NodeType': 'string',
        'NumberOfNodes': 123,
        'ClusterType': 'string',
        'ClusterVersion': 'string',
        'AutomatedSnapshotRetentionPeriod': 123,
        'ClusterIdentifier': 'string'
    },
    'ClusterVersion': 'string',
    'AllowVersionUpgrade': True|False,
    'NumberOfNodes': 123,
    'PubliclyAccessible': True|False,
    'Encrypted': True|False,
    'RestoreStatus': {
        'Status': 'string',
        'CurrentRestoreRateInMegaBytesPerSecond': 123.0,
        'SnapshotSizeInMegaBytes': 123,
        'ProgressInMegaBytes': 123,
        'ElapsedTimeInSeconds': 123,
        'EstimatedTimeToCompletionInSeconds': 123
    },
    'HsmStatus': {
        'HsmClientCertificateIdentifier': 'string',
        'HsmConfigurationIdentifier': 'string',
        'Status': 'string'
    },
    'ClusterSnapshotCopyStatus': {
        'DestinationRegion': 'string',
        'RetentionPeriod': 123,
        'SnapshotCopyGrantName': 'string'
    },
    'ClusterPublicKey': 'string',
    'ClusterNodes': [
        {
            'NodeRole': 'string',
            'PrivateIpAddress': 'string',
            'PublicIpAddress': 'string'
        },
    ],
    'ElasticIpStatus': {
        'ElasticIp': 'string',
        'Status': 'string'
    },
    'ClusterRevisionNumber': 'string',
    'Tags': [
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
    'KmsKeyId': 'string'
}

```

Response Structure

- (*dict*) –

Describes a cluster.

- **Cluster** (*dict*) –

Describes a cluster.

- * **ClusterIdentifier** (*string*) –

The unique identifier of the cluster.

- * **NodeType** (*string*) –

The node type for the nodes in the cluster.

- * **ClusterStatus** (*string*) –

The current state of this cluster. Possible values include `available`, `creating`, `deleting`, `rebooting`, `renaming`, and `resizing`.

- * **ModifyStatus** (*string*) –

The status of a modify operation, if any, initiated for the cluster.

- * **MasterUsername** (*string*) –

The master user name for the cluster. This name is used to connect to the database that is specified in **DBName**.

- * **DBName** (*string*) –

The name of the initial database that was created when the cluster was created. This same name is returned for the life of the cluster. If an initial database was not specified, a database named “dev” was created by default.

- * **Endpoint** (*dict*) –

The connection endpoint.

- **Address** (*string*) –

The DNS address of the Cluster.

- **Port** (*integer*) –

The port that the database engine is listening on.

- * **ClusterCreateTime** (*datetime*) –

The date and time that the cluster was created.

- * **AutomatedSnapshotRetentionPeriod** (*integer*) –

The number of days that automatic cluster snapshots are retained.

- * **ClusterSecurityGroups** (*list*) –

A list of cluster security group that are associated with the cluster. Each security group is represented by an element that contains `ClusterSecurityGroup.Name` and `ClusterSecurityGroup.Status` subelements.

Cluster security groups are used when the cluster is not created in a VPC. Clusters that are created in a VPC use VPC security groups, which are listed by the **VpcSecurityGroups** parameter.

- (*dict*) –

Describes a security group.

- **ClusterSecurityGroupName** (*string*) –

The name of the cluster security group.

- **Status** (*string*) –

The status of the cluster security group.

- * **VpcSecurityGroups** (*list*) –

A list of Virtual Private Cloud (VPC) security groups that are associated with the cluster. This parameter is returned only if the cluster is in a VPC.

- *(dict)* –

Describes the members of a VPC security group.

- **VpcSecurityGroupId** (*string*) –

- **Status** (*string*) –

* **ClusterParameterGroups** (*list*) –

The list of cluster parameter groups that are associated with this cluster. Each parameter group in the list is returned with its status.

- *(dict)* –

Describes the status of a parameter group.

- **ParameterGroupName** (*string*) –

The name of the cluster parameter group.

- **ParameterApplyStatus** (*string*) –

The status of parameter updates.

- **ClusterParameterStatusList** (*list*) –

The list of parameter statuses.

For more information about parameters and parameter groups, go to [Amazon Redshift Parameter Groups](#) in the *Amazon Redshift Cluster Management Guide*.

- *(dict)* –

Describes the status of a parameter group.

- **ParameterName** (*string*) –

The name of the parameter.

- **ParameterApplyStatus** (*string*) –

The status of the parameter that indicates whether the parameter is in sync with the database, waiting for a cluster reboot, or encountered an error when being applied.

The following are possible statuses and descriptions.

- **in-sync** : The parameter value is in sync with the database.
- **pending-reboot** : The parameter value will be applied after the cluster reboots.

- **applying** : The parameter value is being applied to the database.

- **invalid-parameter** : Cannot apply the parameter value because it has an invalid value or syntax.

- **apply-deferred** : The parameter contains static property changes. The changes are deferred until the cluster reboots.

- **apply-error** : Cannot connect to the cluster. The parameter change will be applied after the cluster reboots.

- **unknown-error** : Cannot apply the parameter change right now. The change will be applied after the cluster reboots.

- **ParameterApplyErrorDescription** (*string*) –

The error that prevented the parameter from being applied to the database.

- * **ClusterSubnetGroupName** (*string*) –

The name of the subnet group that is associated with the cluster. This parameter is valid only when the cluster is in a VPC.

- * **VpcId** (*string*) –

The identifier of the VPC the cluster is in, if the cluster is in a VPC.

- * **AvailabilityZone** (*string*) –

The name of the Availability Zone in which the cluster is located.

- * **PreferredMaintenanceWindow** (*string*) –

The weekly time range (in UTC) during which system maintenance can occur.

- * **PendingModifiedValues** (*dict*) –

If present, changes to the cluster are pending. Specific pending changes are identified by subelements.

- **MasterUserPassword** (*string*) –

The pending or in-progress change of the master user password for the cluster.

- **NodeType** (*string*) –

The pending or in-progress change of the cluster's node type.

- **NumberOfNodes** (*integer*) –

The pending or in-progress change of the number of nodes in the cluster.

- **ClusterType** (*string*) –

The pending or in-progress change of the cluster type.

- **ClusterVersion** (*string*) –

The pending or in-progress change of the service version.

- **AutomatedSnapshotRetentionPeriod** (*integer*) –

The pending or in-progress change of the automated snapshot retention period.

- **ClusterIdentifier** (*string*) –

The pending or in-progress change of the new identifier for the cluster.

- * **ClusterVersion** (*string*) –

The version ID of the Amazon Redshift engine that is running on the cluster.

- * **AllowVersionUpgrade** (*boolean*) –

If `true`, major version upgrades will be applied automatically to the cluster during the maintenance window.

- * **NumberOfNodes** (*integer*) –

The number of compute nodes in the cluster.

- * **PubliclyAccessible** (*boolean*) –

If `true`, the cluster can be accessed from a public network.

- * **Encrypted** (*boolean*) –

If `true`, data in the cluster is encrypted at rest.

- * **RestoreStatus** (*dict*) –

Describes the status of a cluster restore action. Returns null if the cluster was not created by restoring a snapshot.

- **Status** (*string*) –

The status of the restore action. Returns starting, restoring, completed, or failed.

- **CurrentRestoreRateInMegaBytesPerSecond** (*float*) –

The number of megabytes per second being transferred from the backup storage. Returns the average rate for a completed backup.

- **SnapshotSizeInMegaBytes** (*integer*) –

The size of the set of snapshot data used to restore the cluster.

- **ProgressInMegaBytes** (*integer*) –

The number of megabytes that have been transferred from snapshot storage.

- **ElapsedTimeInSeconds** (*integer*) –

The amount of time an in-progress restore has been running, or the amount of time it took a completed restore to finish.

- **EstimatedTimeToCompletionInSeconds** (*integer*) –

The estimate of the time remaining before the restore will complete. Returns 0 for a completed restore.

- * **HsmStatus** (*dict*) –

Reports whether the Amazon Redshift cluster has finished applying any HSM settings changes specified in a modify cluster command.

Values: active, applying

- **HsmClientCertificateIdentifier** (*string*) –

Specifies the name of the HSM client certificate the Amazon Redshift cluster uses to retrieve the data encryption keys stored in an HSM.

- **HsmConfigurationIdentifier** (*string*) –

Specifies the name of the HSM configuration that contains the information the Amazon Redshift cluster can use to retrieve and store keys in an HSM.

- **Status** (*string*) –

Reports whether the Amazon Redshift cluster has finished applying any HSM settings changes specified in a modify cluster command.

Values: active, applying

- * **ClusterSnapshotCopyStatus** (*dict*) –

Returns the destination region and retention period that are configured for cross-region snapshot copy.

- **DestinationRegion** (*string*) –

The destination region that snapshots are automatically copied to when cross-region snapshot copy is enabled.

- **RetentionPeriod** (*integer*) –

The number of days that automated snapshots are retained in the destination region after they are copied from a source

- region.
- **SnapshotCopyGrantName** (*string*) –
The name of the snapshot copy grant.
- * **ClusterPublicKey** (*string*) –
The public key for the cluster.
- * **ClusterNodes** (*list*) –
The nodes in a cluster.
 - (*dict*) –
The identifier of a node in a cluster.
 - **NodeRole** (*string*) –
Whether the node is a leader node or a compute node.
 - **PrivateIPAddress** (*string*) –
The private IP address of a node within a cluster.
 - **PublicIPAddress** (*string*) –
The public IP address of a node within a cluster.
- * **ElasticIpStatus** (*dict*) –
Describes the status of the elastic IP (EIP) address.
 - **ElasticIp** (*string*) –
The elastic IP (EIP) address for the cluster.
 - **Status** (*string*) –
Describes the status of the elastic IP (EIP) address.
- * **ClusterRevisionNumber** (*string*) –
The specific revision number of the database in the cluster.
- * **Tags** (*list*) –
The list of tags for the cluster.
 - (*dict*) –
A tag consisting of a name/value pair for a resource.
 - **Key** (*string*) –
The key, or name, for the resource tag.
 - **Value** (*string*) –
The value for the resource tag.
- * **KmsKeyId** (*string*) –
The AWS Key Management Service (KMS) key ID of the encryption key used to encrypt data in the cluster.

reset_cluster_parameter_group (**kwargs)

Sets one or more parameters of the specified parameter group to their default values and sets the source values of the parameters to “engine-default”. To reset the entire parameter group specify the *ResetAllParameters* parameter. For parameter changes to take effect you must reboot any associated clusters.

Request Syntax

```
response = client.reset_cluster_parameter_group(
    ParameterGroupName='string',
    ResetAllParameters=True|False,
    Parameters=[
        {
```

```

        'ParameterName': 'string',
        'ParameterValue': 'string',
        'Description': 'string',
        'Source': 'string',
        'DataType': 'string',
        'AllowedValues': 'string',
        'ApplyType': 'static'|'dynamic',
        'IsModifiable': True|False,
        'MinimumEngineVersion': 'string'
    },
]
)

```

Parameters

- **ParameterGroupName** (*string*) – [REQUIRED]

The name of the cluster parameter group to be reset.

- **ResetAllParameters** (*boolean*) – If `true`, all parameters in the specified parameter group will be reset to their default values.

Default: `true`

- **Parameters** (*list*) – An array of names of parameters to be reset. If *ResetAllParameters* option is not used, then at least one parameter name must be supplied.

Constraints: A maximum of 20 parameters can be reset in a single request.

– (*dict*) –

Describes a parameter in a cluster parameter group.

- * **ParameterName** (*string*) –

The name of the parameter.

- * **ParameterValue** (*string*) –

The value of the parameter.

- * **Description** (*string*) –

A description of the parameter.

- * **Source** (*string*) –

The source of the parameter value, such as “engine-default” or “user”.

- * **DataType** (*string*) –

The data type of the parameter.

- * **AllowedValues** (*string*) –

The valid range of values for the parameter.

- * **ApplyType** (*string*) –

Specifies how to apply the parameter. Supported value: `static`.

- * **IsModifiable** (*boolean*) –

If `true`, the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.

- * **MinimumEngineVersion** (*string*) –

The earliest engine version to which the parameter can apply.

Return type `dict`

Returns

Response Syntax

```
{
    'ParameterGroupName': 'string',
    'ParameterGroupStatus': 'string'
}
```

Response Structure

- (dict) –

Contains the output from the `ModifyClusterParameterGroup` and `ResetClusterParameterGroup` actions and indicate the parameter group involved and the status of the operation on the parameter group.

- **ParameterGroupName** (string) –

The name of the cluster parameter group.

- **ParameterGroupStatus** (string) –

The status of the parameter group. For example, if you made a change to a parameter group name-value pair, then the change could be pending a reboot of an associated cluster.

restore_from_cluster_snapshot (**kwargs)

Creates a new cluster from a snapshot. By default, Amazon Redshift creates the resulting cluster with the same configuration as the original cluster from which the snapshot was created, except that the new cluster is created with the default cluster security and parameter groups. After Amazon Redshift creates the cluster, you can use the `ModifyCluster` API to associate a different security group and different parameter group with the restored cluster. If you are using a DS node type, you can also choose to change to another DS node type of the same size during restore.

If you restore a cluster into a VPC, you must provide a cluster subnet group where you want the cluster restored.

For more information about working with snapshots, go to [Amazon Redshift Snapshots](#) in the *Amazon Redshift Cluster Management Guide*.

Request Syntax

```
response = client.restore_from_cluster_snapshot(
    ClusterIdentifier='string',
    SnapshotIdentifier='string',
    SnapshotClusterIdentifier='string',
    Port=123,
    AvailabilityZone='string',
    AllowVersionUpgrade=True|False,
    ClusterSubnetGroupName='string',
    PubliclyAccessible=True|False,
    OwnerAccount='string',
    HsmClientCertificateIdentifier='string',
    HsmConfigurationIdentifier='string',
    ElasticIp='string',
    ClusterParameterGroupName='string',
    ClusterSecurityGroups=[
        'string',
    ],
    VpcSecurityGroupIds=[
        'string',
    ],
    PreferredMaintenanceWindow='string',
    AutomatedSnapshotRetentionPeriod=123,
    KmsKeyId='string',
```



```
NodeType='string'
)
```

Parameters

- **ClusterIdentifier** (*string*) – [REQUIRED]

The identifier of the cluster that will be created from restoring the snapshot.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens.
- Alphabetic characters must be lowercase.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.
- Must be unique for all clusters within an AWS account.

- **SnapshotIdentifier** (*string*) – [REQUIRED]

The name of the snapshot from which to create the new cluster. This parameter isn't case sensitive.

Example: `my-snapshot-id`

- **SnapshotClusterIdentifier** (*string*) – The name of the cluster the source snapshot was created from. This parameter is required if your IAM user has a policy containing a snapshot resource element that specifies anything other than `*` for the cluster name.
- **Port** (*integer*) – The port number on which the cluster accepts connections.

Default: The same port as the original cluster.

Constraints: Must be between 1115 and 65535 .

- **AvailabilityZone** (*string*) – The Amazon EC2 Availability Zone in which to restore the cluster.

Default: A random, system-chosen Availability Zone.

Example: `us-east-1a`

- **AllowVersionUpgrade** (*boolean*) – If `true` , major version upgrades can be applied during the maintenance window to the Amazon Redshift engine that is running on the cluster.

Default: `true`

- **ClusterSubnetGroupName** (*string*) – The name of the subnet group where you want to cluster restored.

A snapshot of cluster in VPC can be restored only in VPC. Therefore, you must provide subnet group name where you want the cluster restored.

- **PubliclyAccessible** (*boolean*) – If `true` , the cluster can be accessed from a public network.
- **OwnerAccount** (*string*) – The AWS customer account used to create or copy the snapshot. Required if you are restoring a snapshot you do not own, optional if you own the snapshot.
- **HsmClientCertificateIdentifier** (*string*) – Specifies the name of the HSM client certificate the Amazon Redshift cluster uses to retrieve the data encryption keys stored in an HSM.
- **HsmConfigurationIdentifier** (*string*) – Specifies the name of the HSM configuration that contains the information the Amazon Redshift cluster can use to retrieve and store keys in an HSM.
- **ElasticIp** (*string*) – The elastic IP (EIP) address for the cluster.
- **ClusterParameterGroupName** (*string*) – The name of the parameter group to be associated with this cluster.

Default: The default Amazon Redshift cluster parameter group. For information about the default parameter group, go to [Working with Amazon Redshift Parameter Groups](#).

Constraints:

- Must be 1 to 255 alphanumeric characters or hyphens.
 - First character must be a letter.
 - Cannot end with a hyphen or contain two consecutive hyphens.
- **ClusterSecurityGroups** (*list*) – A list of security groups to be associated with this cluster.

Default: The default cluster security group for Amazon Redshift.

Cluster security groups only apply to clusters outside of VPCs.

– (*string*) –

- **VpcSecurityGroupIds** (*list*) – A list of Virtual Private Cloud (VPC) security groups to be associated with the cluster.

Default: The default VPC security group is associated with the cluster.

VPC security groups only apply to clusters in VPCs.

– (*string*) –

- **PreferredMaintenanceWindow** (*string*) – The weekly time range (in UTC) during which automated cluster maintenance can occur.

Format: ddd:hh24:mi–ddd:hh24:mi

Default: The value selected for the cluster from which the snapshot was taken. For more information about the time blocks for each region, see [Maintenance Windows](#) in Amazon Redshift Cluster Management Guide.

Valid Days: Mon | Tue | Wed | Thu | Fri | Sat | Sun

Constraints: Minimum 30-minute window.

- **AutomatedSnapshotRetentionPeriod** (*integer*) – The number of days that automated snapshots are retained. If the value is 0, automated snapshots are disabled. Even if automated snapshots are disabled, you can still create manual snapshots when you want with `CreateClusterSnapshot`.

Default: The value selected for the cluster from which the snapshot was taken.

Constraints: Must be a value from 0 to 35.

- **KmsKeyId** (*string*) – The AWS Key Management Service (KMS) key ID of the encryption key that you want to use to encrypt data in the cluster that you restore from a shared snapshot.
- **NodeType** (*string*) – The node type that the restored cluster will be provisioned with.

Default: The node type of the cluster from which the snapshot was taken. You can modify this if you are using any DS node type. In that case, you can choose to restore into another DS node type of the same size. For example, you can restore `ds1.8xlarge` into `ds2.8xlarge`, or `ds2.xlarge` into `ds1.xlarge`. If you have a DC instance type, you must restore into that same instance type and size. In other words, you can only restore a `dc1.large` instance type into another `dc1.large` instance type. For more information about node types, see [About Clusters and Nodes](#) in the *Amazon Redshift Cluster Management Guide*.

Return type dict

Returns

Response Syntax

```

{
    'Cluster': {
        'ClusterIdentifier': 'string',
        'NodeType': 'string',
        'ClusterStatus': 'string',
        'ModifyStatus': 'string',
        'MasterUsername': 'string',
        'DBName': 'string',
        'Endpoint': {
            'Address': 'string',
            'Port': 123
        },
    },
    'ClusterCreateTime': datetime(2015, 1, 1),
    'AutomatedSnapshotRetentionPeriod': 123,
    'ClusterSecurityGroups': [
        {
            'ClusterSecurityGroupName': 'string',
            'Status': 'string'
        },
    ],
    'VpcSecurityGroups': [
        {
            'VpcSecurityGroupId': 'string',
            'Status': 'string'
        },
    ],
    'ClusterParameterGroups': [
        {
            'ParameterGroupName': 'string',
            'ParameterApplyStatus': 'string',
            'ClusterParameterStatusList': [
                {
                    'ParameterName': 'string',
                    'ParameterApplyStatus': 'string',
                    'ParameterApplyErrorDescription': 'string'
                },
            ]
        },
    ],
    'ClusterSubnetGroupName': 'string',
    'VpcId': 'string',
    'AvailabilityZone': 'string',
    'PreferredMaintenanceWindow': 'string',
    'PendingModifiedValues': {
        'MasterUserPassword': 'string',
        'NodeType': 'string',
        'NumberOfNodes': 123,
        'ClusterType': 'string',
        'ClusterVersion': 'string',
        'AutomatedSnapshotRetentionPeriod': 123,
        'ClusterIdentifier': 'string'
    },
    'ClusterVersion': 'string',
    'AllowVersionUpgrade': True|False,
    'NumberOfNodes': 123,
    'PubliclyAccessible': True|False,
    'Encrypted': True|False,
    'RestoreStatus': {

```

```
        'Status': 'string',
        'CurrentRestoreRateInMegaBytesPerSecond': 123.0,
        'SnapshotSizeInMegaBytes': 123,
        'ProgressInMegaBytes': 123,
        'ElapsedTimeInSeconds': 123,
        'EstimatedTimeToCompletionInSeconds': 123
    },
    'HsmStatus': {
        'HsmClientCertificateIdentifier': 'string',
        'HsmConfigurationIdentifier': 'string',
        'Status': 'string'
    },
    'ClusterSnapshotCopyStatus': {
        'DestinationRegion': 'string',
        'RetentionPeriod': 123,
        'SnapshotCopyGrantName': 'string'
    },
    'ClusterPublicKey': 'string',
    'ClusterNodes': [
        {
            'NodeRole': 'string',
            'PrivateIpAddress': 'string',
            'PublicIpAddress': 'string'
        },
    ],
    'ElasticIpStatus': {
        'ElasticIp': 'string',
        'Status': 'string'
    },
    'ClusterRevisionNumber': 'string',
    'Tags': [
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
    'KmsKeyId': 'string'
}
```

Response Structure

- (*dict*) –

Describes a cluster.

- **Cluster** (*dict*) –

Describes a cluster.

- * **ClusterIdentifier** (*string*) –

The unique identifier of the cluster.

- * **NodeType** (*string*) –

The node type for the nodes in the cluster.

- * **ClusterStatus** (*string*) –

The current state of this cluster. Possible values include available , creating , deleting , rebooting , renaming , and resizing .

- * **ModifyStatus** (*string*) –

The status of a modify operation, if any, initiated for the cluster.

* **MasterUsername** (*string*) –

The master user name for the cluster. This name is used to connect to the database that is specified in **DBName**.

* **DBName** (*string*) –

The name of the initial database that was created when the cluster was created. This same name is returned for the life of the cluster. If an initial database was not specified, a database named “dev” was created by default.

* **Endpoint** (*dict*) –

The connection endpoint.

· **Address** (*string*) –

The DNS address of the Cluster.

· **Port** (*integer*) –

The port that the database engine is listening on.

* **ClusterCreateTime** (*datetime*) –

The date and time that the cluster was created.

* **AutomatedSnapshotRetentionPeriod** (*integer*) –

The number of days that automatic cluster snapshots are retained.

* **ClusterSecurityGroups** (*list*) –

A list of cluster security group that are associated with the cluster. Each security group is represented by an element that contains `ClusterSecurityGroup.Name` and `ClusterSecurityGroup.Status` subelements.

Cluster security groups are used when the cluster is not created in a VPC. Clusters that are created in a VPC use VPC security groups, which are listed by the **VpcSecurityGroups** parameter.

· (*dict*) –

Describes a security group.

· **ClusterSecurityGroupName** (*string*) –

The name of the cluster security group.

· **Status** (*string*) –

The status of the cluster security group.

* **VpcSecurityGroups** (*list*) –

A list of Virtual Private Cloud (VPC) security groups that are associated with the cluster. This parameter is returned only if the cluster is in a VPC.

· (*dict*) –

Describes the members of a VPC security group.

· **VpcSecurityGroupId** (*string*) –

· **Status** (*string*) –

* **ClusterParameterGroups** (*list*) –

The list of cluster parameter groups that are associated with this cluster. Each parameter group in the list is returned with its status.

· (*dict*) –

Describes the status of a parameter group.

- **ParameterGroupName** (*string*) –

The name of the cluster parameter group.

- **ParameterApplyStatus** (*string*) –

The status of parameter updates.

- **ClusterParameterStatusList** (*list*) –

The list of parameter statuses.

For more information about parameters and parameter groups, go to [Amazon Redshift Parameter Groups](#) in the *Amazon Redshift Cluster Management Guide*.

- (*dict*) –

Describes the status of a parameter group.

- **ParameterName** (*string*) –

The name of the parameter.

- **ParameterApplyStatus** (*string*) –

The status of the parameter that indicates whether the parameter is in sync with the database, waiting for a cluster reboot, or encountered an error when being applied.

The following are possible statuses and descriptions.

- `in-sync` : The parameter value is in sync with the database.
- `pending-reboot` : The parameter value will be applied after the cluster reboots.
- `applying` : The parameter value is being applied to the database.
- `invalid-parameter` : Cannot apply the parameter value because it has an invalid value or syntax.
- `apply-deferred` : The parameter contains static property changes. The changes are deferred until the cluster reboots.
- `apply-error` : Cannot connect to the cluster. The parameter change will be applied after the cluster reboots.
- `unknown-error` : Cannot apply the parameter change right now. The change will be applied after the cluster reboots.

- **ParameterApplyErrorDescription** (*string*) –

The error that prevented the parameter from being applied to the database.

- * **ClusterSubnetGroupName** (*string*) –

The name of the subnet group that is associated with the cluster. This parameter is valid only when the cluster is in a VPC.

- * **VpcId** (*string*) –

The identifier of the VPC the cluster is in, if the cluster is in a VPC.

- * **AvailabilityZone** (*string*) –

The name of the Availability Zone in which the cluster is located.

- * **PreferredMaintenanceWindow** (*string*) –

The weekly time range (in UTC) during which system maintenance can occur.

- * **PendingModifiedValues** (*dict*) –

If present, changes to the cluster are pending. Specific pending changes are identified by subelements.

- **MasterUserPassword** (*string*) –

The pending or in-progress change of the master user password for the cluster.

- **NodeType** (*string*) –

The pending or in-progress change of the cluster's node type.

- **NumberOfNodes** (*integer*) –

The pending or in-progress change of the number of nodes in the cluster.

- **ClusterType** (*string*) –

The pending or in-progress change of the cluster type.

- **ClusterVersion** (*string*) –

The pending or in-progress change of the service version.

- **AutomatedSnapshotRetentionPeriod** (*integer*) –

The pending or in-progress change of the automated snapshot retention period.

- **ClusterIdentifier** (*string*) –

The pending or in-progress change of the new identifier for the cluster.

- * **ClusterVersion** (*string*) –

The version ID of the Amazon Redshift engine that is running on the cluster.

- * **AllowVersionUpgrade** (*boolean*) –

If `true`, major version upgrades will be applied automatically to the cluster during the maintenance window.

- * **NumberOfNodes** (*integer*) –

The number of compute nodes in the cluster.

- * **PubliclyAccessible** (*boolean*) –

If `true`, the cluster can be accessed from a public network.

- * **Encrypted** (*boolean*) –

If `true`, data in the cluster is encrypted at rest.

- * **RestoreStatus** (*dict*) –

Describes the status of a cluster restore action. Returns null if the cluster was not created by restoring a snapshot.

- **Status** (*string*) –

The status of the restore action. Returns starting, restoring, completed, or failed.

- **CurrentRestoreRateInMegaBytesPerSecond** (*float*) –

The number of megabytes per second being transferred from the backup storage. Returns the average rate for a completed backup.

- **SnapshotSizeInMegaBytes** (*integer*) –

The size of the set of snapshot data used to restore the cluster.

- **ProgressInMegaBytes** (*integer*) –

The number of megabytes that have been transferred from snapshot storage.

- **ElapsedTimeInSeconds** (*integer*) –

The amount of time an in-progress restore has been running, or the amount of time it took a completed restore to finish.

- **EstimatedTimeToCompletionInSeconds** (*integer*) –

The estimate of the time remaining before the restore will complete. Returns 0 for a completed restore.

- * **HsmStatus** (*dict*) –

Reports whether the Amazon Redshift cluster has finished applying any HSM settings changes specified in a modify cluster command.

Values: active, applying

- **HsmClientCertificateIdentifier** (*string*) –

Specifies the name of the HSM client certificate the Amazon Redshift cluster uses to retrieve the data encryption keys stored in an HSM.

- **HsmConfigurationIdentifier** (*string*) –

Specifies the name of the HSM configuration that contains the information the Amazon Redshift cluster can use to retrieve and store keys in an HSM.

- **Status** (*string*) –

Reports whether the Amazon Redshift cluster has finished applying any HSM settings changes specified in a modify cluster command.

Values: active, applying

- * **ClusterSnapshotCopyStatus** (*dict*) –

Returns the destination region and retention period that are configured for cross-region snapshot copy.

- **DestinationRegion** (*string*) –

The destination region that snapshots are automatically copied to when cross-region snapshot copy is enabled.

- **RetentionPeriod** (*integer*) –

The number of days that automated snapshots are retained in the destination region after they are copied from a source region.

- **SnapshotCopyGrantName** (*string*) –

The name of the snapshot copy grant.

- * **ClusterPublicKey** (*string*) –

The public key for the cluster.

- * **ClusterNodes** (*list*) –

The nodes in a cluster.

- (*dict*) –

The identifier of a node in a cluster.

- **NodeRole** (*string*) –

Whether the node is a leader node or a compute node.

- **PrivateIPAddress** (*string*) –
The private IP address of a node within a cluster.
- **PublicIPAddress** (*string*) –
The public IP address of a node within a cluster.
- * **ElasticIpStatus** (*dict*) –
Describes the status of the elastic IP (EIP) address.
 - **ElasticIp** (*string*) –
The elastic IP (EIP) address for the cluster.
 - **Status** (*string*) –
Describes the status of the elastic IP (EIP) address.
- * **ClusterRevisionNumber** (*string*) –
The specific revision number of the database in the cluster.
- * **Tags** (*list*) –
The list of tags for the cluster.
 - (*dict*) –
A tag consisting of a name/value pair for a resource.
 - **Key** (*string*) –
The key, or name, for the resource tag.
 - **Value** (*string*) –
The value for the resource tag.
- * **KmsKeyId** (*string*) –
The AWS Key Management Service (KMS) key ID of the encryption key used to encrypt data in the cluster.

revoke_cluster_security_group_ingress (***kwargs*)

Revokes an ingress rule in an Amazon Redshift security group for a previously authorized IP range or Amazon EC2 security group. To add an ingress rule, see `AuthorizeClusterSecurityGroupIngress`. For information about managing security groups, go to [Amazon Redshift Cluster Security Groups](#) in the *Amazon Redshift Cluster Management Guide*.

Request Syntax

```
response = client.revoke_cluster_security_group_ingress(
    ClusterSecurityGroupName='string',
    CIDRIP='string',
    EC2SecurityGroupName='string',
    EC2SecurityGroupOwnerId='string'
)
```

Parameters

- **ClusterSecurityGroupName** (*string*) – [REQUIRED]
The name of the security Group from which to revoke the ingress rule.
- **CIDRIP** (*string*) – The IP range for which to revoke access. This range must be a valid Classless Inter-Domain Routing (CIDR) block of IP addresses. If CIDRIP is specified, EC2SecurityGroupName and EC2SecurityGroupOwnerId cannot be provided.
- **EC2SecurityGroupName** (*string*) – The name of the EC2 Security Group whose access is to be revoked. If EC2SecurityGroupName is specified,

EC2SecurityGroupOwnerId must also be provided and CIDRIP cannot be provided.

- **EC2SecurityGroupOwnerId** (*string*) – The AWS account number of the owner of the security group specified in the EC2SecurityGroupName parameter. The AWS access key ID is not an acceptable value. If EC2SecurityGroupOwnerId is specified, EC2SecurityGroupName must also be provided. and CIDRIP cannot be provided.

Example: 111122223333

Return type dict

Returns

Response Syntax

```
{
  'ClusterSecurityGroup': {
    'ClusterSecurityGroupName': 'string',
    'Description': 'string',
    'EC2SecurityGroups': [
      {
        'Status': 'string',
        'EC2SecurityGroupName': 'string',
        'EC2SecurityGroupOwnerId': 'string',
        'Tags': [
          {
            'Key': 'string',
            'Value': 'string'
          },
        ],
      },
    ],
  },
  'IPRanges': [
    {
      'Status': 'string',
      'CIDRIP': 'string',
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        },
      ],
    },
  ],
  'Tags': [
    {
      'Key': 'string',
      'Value': 'string'
    },
  ],
}
```

Response Structure

- (*dict*) –
Describes a security group.
 - **ClusterSecurityGroup** (*dict*) –
Describes a security group.

* **ClusterSecurityGroupName** (*string*) –

The name of the cluster security group to which the operation was applied.

* **Description** (*string*) –

A description of the security group.

* **EC2SecurityGroups** (*list*) –

A list of EC2 security groups that are permitted to access clusters associated with this cluster security group.

· (*dict*) –

Describes an Amazon EC2 security group.

· **Status** (*string*) –

The status of the EC2 security group.

· **EC2SecurityGroupName** (*string*) –

The name of the EC2 Security Group.

· **EC2SecurityGroupOwnerId** (*string*) –

The AWS ID of the owner of the EC2 security group specified in the `EC2SecurityGroupName` field.

· **Tags** (*list*) –

The list of tags for the EC2 security group.

· (*dict*) –

A tag consisting of a name/value pair for a resource.

· **Key** (*string*) –

The key, or name, for the resource tag.

· **Value** (*string*) –

The value for the resource tag.

* **IPRanges** (*list*) –

A list of IP ranges (CIDR blocks) that are permitted to access clusters associated with this cluster security group.

· (*dict*) –

Describes an IP range used in a security group.

· **Status** (*string*) –

The status of the IP range, for example, “authorized”.

· **CIDRIP** (*string*) –

The IP range in Classless Inter-Domain Routing (CIDR) notation.

· **Tags** (*list*) –

The list of tags for the IP range.

· (*dict*) –

A tag consisting of a name/value pair for a resource.

· **Key** (*string*) –

The key, or name, for the resource tag.

· **Value** (*string*) –

The value for the resource tag.

* **Tags** (*list*) –

The list of tags for the cluster security group.

· (*dict*) –

A tag consisting of a name/value pair for a resource.

· **Key** (*string*) –

The key, or name, for the resource tag.

· **Value** (*string*) –

The value for the resource tag.

revoke_snapshot_access (***kwargs*)

Removes the ability of the specified AWS customer account to restore the specified snapshot. If the account is currently restoring the snapshot, the restore will run to completion.

For more information about working with snapshots, go to [Amazon Redshift Snapshots](#) in the *Amazon Redshift Cluster Management Guide*.

Request Syntax

```
response = client.revoke_snapshot_access(
    SnapshotIdentifier='string',
    SnapshotClusterIdentifier='string',
    AccountWithRestoreAccess='string'
)
```

Parameters

• **SnapshotIdentifier** (*string*) – [REQUIRED]

The identifier of the snapshot that the account can no longer access.

• **SnapshotClusterIdentifier** (*string*) – The identifier of the cluster the snapshot was created from. This parameter is required if your IAM user has a policy containing a snapshot resource element that specifies anything other than * for the cluster name.

• **AccountWithRestoreAccess** (*string*) – [REQUIRED]

The identifier of the AWS customer account that can no longer restore the specified snapshot.

Return type dict

Returns

Response Syntax

```
{
  'Snapshot': {
    'SnapshotIdentifier': 'string',
    'ClusterIdentifier': 'string',
    'SnapshotCreateTime': datetime(2015, 1, 1),
    'Status': 'string',
    'Port': 123,
    'AvailabilityZone': 'string',
    'ClusterCreateTime': datetime(2015, 1, 1),
    'MasterUsername': 'string',
    'ClusterVersion': 'string',
    'SnapshotType': 'string',
    'NodeType': 'string',
    'NumberOfNodes': 123,
    'DBName': 'string',
```

```

'VpcId': 'string',
'Encrypted': True|False,
'KmsKeyId': 'string',
'EncryptedWithHSM': True|False,
'AccountsWithRestoreAccess': [
    {
        'AccountId': 'string'
    },
],
'OwnerAccount': 'string',
'TotalBackupSizeInMegaBytes': 123.0,
'ActualIncrementalBackupSizeInMegaBytes': 123.0,
'BackupProgressInMegaBytes': 123.0,
'CurrentBackupRateInMegaBytesPerSecond': 123.0,
'EstimatedSecondsToCompletion': 123,
'ElapsedTimeInSeconds': 123,
'SourceRegion': 'string',
'Tags': [
    {
        'Key': 'string',
        'Value': 'string'
    },
],
'RestorableNodeTypes': [
    'string',
]
}

```

Response Structure

- (*dict*) –

Describes a snapshot.

- **Snapshot** (*dict*) –

Describes a snapshot.

- * **SnapshotIdentifier** (*string*) –

The snapshot identifier that is provided in the request.

- * **ClusterIdentifier** (*string*) –

The identifier of the cluster for which the snapshot was taken.

- * **SnapshotCreateTime** (*datetime*) –

The time (UTC) when Amazon Redshift began the snapshot. A snapshot contains a copy of the cluster data as of this exact time.

- * **Status** (*string*) –

The snapshot status. The value of the status depends on the API operation used.

- CreateClusterSnapshot and CopyClusterSnapshot returns status as “creating”.
- DescribeClusterSnapshots returns status as “creating”, “available”, “final snapshot”, or “failed”.
- DeleteClusterSnapshot returns status as “deleted”.

- * **Port** (*integer*) –

The port that the cluster is listening on.

- * **AvailabilityZone** (*string*) –

The Availability Zone in which the cluster was created.

* **ClusterCreateTime** (*datetime*) –

The time (UTC) when the cluster was originally created.

* **MasterUsername** (*string*) –

The master user name for the cluster.

* **ClusterVersion** (*string*) –

The version ID of the Amazon Redshift engine that is running on the cluster.

* **SnapshotType** (*string*) –

The snapshot type. Snapshots created using `CreateClusterSnapshot` and `CopyClusterSnapshot` will be of type “manual”.

* **NodeType** (*string*) –

The node type of the nodes in the cluster.

* **NumberOfNodes** (*integer*) –

The number of nodes in the cluster.

* **DBName** (*string*) –

The name of the database that was created when the cluster was created.

* **VpcId** (*string*) –

The VPC identifier of the cluster if the snapshot is from a cluster in a VPC. Otherwise, this field is not in the output.

* **Encrypted** (*boolean*) –

If `true`, the data in the snapshot is encrypted at rest.

* **KmsKeyId** (*string*) –

The AWS Key Management Service (KMS) key ID of the encryption key that was used to encrypt data in the cluster from which the snapshot was taken.

* **EncryptedWithHSM** (*boolean*) –

A boolean that indicates whether the snapshot data is encrypted using the HSM keys of the source cluster. `true` indicates that the data is encrypted using HSM keys.

* **AccountsWithRestoreAccess** (*list*) –

A list of the AWS customer accounts authorized to restore the snapshot. Returns `null` if no accounts are authorized. Visible only to the snapshot owner.

· (*dict*) –

Describes an AWS customer account authorized to restore a snapshot.

· **AccountId** (*string*) –

The identifier of an AWS customer account authorized to restore a snapshot.

* **OwnerAccount** (*string*) –

For manual snapshots, the AWS customer account used to create or copy the snapshot. For automatic snapshots, the owner of the cluster. The owner can perform all snapshot actions, such as sharing a manual snapshot.

- * **TotalBackupSizeInMegaBytes** (*float*) –
The size of the complete set of backup data that would be used to restore the cluster.
- * **ActualIncrementalBackupSizeInMegaBytes** (*float*) –
The size of the incremental backup.
- * **BackupProgressInMegaBytes** (*float*) –
The number of megabytes that have been transferred to the snapshot backup.
- * **CurrentBackupRateInMegaBytesPerSecond** (*float*) –
The number of megabytes per second being transferred to the snapshot backup. Returns 0 for a completed backup.
- * **EstimatedSecondsToCompletion** (*integer*) –
The estimate of the time remaining before the snapshot backup will complete. Returns 0 for a completed backup.
- * **ElapsedTimeInSeconds** (*integer*) –
The amount of time an in-progress snapshot backup has been running, or the amount of time it took a completed backup to finish.
- * **SourceRegion** (*string*) –
The source region from which the snapshot was copied.
- * **Tags** (*list*) –
The list of tags for the cluster snapshot.
 - (*dict*) –
A tag consisting of a name/value pair for a resource.
 - **Key** (*string*) –
The key, or name, for the resource tag.
 - **Value** (*string*) –
The value for the resource tag.
- * **RestorableNodeTypes** (*list*) –
The list of node types that this cluster snapshot is able to restore into.
 - (*string*) –

rotate_encryption_key (**kwargs)
Rotates the encryption keys for a cluster.

Request Syntax

```
response = client.rotate_encryption_key(
    ClusterIdentifier='string'
)
```

Parameters ClusterIdentifier (*string*) – [REQUIRED]

The unique identifier of the cluster that you want to rotate the encryption keys for.

Constraints: Must be the name of valid cluster that has encryption enabled.

Return type dict

Returns

Response Syntax

```
{
  'Cluster': {
    'ClusterIdentifier': 'string',
    'NodeType': 'string',
    'ClusterStatus': 'string',
    'ModifyStatus': 'string',
    'MasterUsername': 'string',
    'DBName': 'string',
    'Endpoint': {
      'Address': 'string',
      'Port': 123
    },
  },
  'ClusterCreateTime': datetime(2015, 1, 1),
  'AutomatedSnapshotRetentionPeriod': 123,
  'ClusterSecurityGroups': [
    {
      'ClusterSecurityGroupName': 'string',
      'Status': 'string'
    },
  ],
  'VpcSecurityGroups': [
    {
      'VpcSecurityGroupId': 'string',
      'Status': 'string'
    },
  ],
  'ClusterParameterGroups': [
    {
      'ParameterGroupName': 'string',
      'ParameterApplyStatus': 'string',
      'ClusterParameterStatusList': [
        {
          'ParameterName': 'string',
          'ParameterApplyStatus': 'string',
          'ParameterApplyErrorDescription': 'string'
        },
      ]
    },
  ],
  'ClusterSubnetGroupName': 'string',
  'VpcId': 'string',
  'AvailabilityZone': 'string',
  'PreferredMaintenanceWindow': 'string',
  'PendingModifiedValues': {
    'MasterUserPassword': 'string',
    'NodeType': 'string',
    'NumberOfNodes': 123,
    'ClusterType': 'string',
    'ClusterVersion': 'string',
    'AutomatedSnapshotRetentionPeriod': 123,
    'ClusterIdentifier': 'string'
  },
  'ClusterVersion': 'string',
  'AllowVersionUpgrade': True|False,
  'NumberOfNodes': 123,
  'PubliclyAccessible': True|False,
  'Encrypted': True|False,
  'RestoreStatus': {
```



```

        'Status': 'string',
        'CurrentRestoreRateInMegaBytesPerSecond': 123.0,
        'SnapshotSizeInMegaBytes': 123,
        'ProgressInMegaBytes': 123,
        'ElapsedTimeInSeconds': 123,
        'EstimatedTimeToCompletionInSeconds': 123
    },
    'HsmStatus': {
        'HsmClientCertificateIdentifier': 'string',
        'HsmConfigurationIdentifier': 'string',
        'Status': 'string'
    },
    'ClusterSnapshotCopyStatus': {
        'DestinationRegion': 'string',
        'RetentionPeriod': 123,
        'SnapshotCopyGrantName': 'string'
    },
    'ClusterPublicKey': 'string',
    'ClusterNodes': [
        {
            'NodeRole': 'string',
            'PrivateIPAddress': 'string',
            'PublicIPAddress': 'string'
        },
    ],
    'ElasticIpStatus': {
        'ElasticIp': 'string',
        'Status': 'string'
    },
    'ClusterRevisionNumber': 'string',
    'Tags': [
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
    'KmsKeyId': 'string'
}

```

Response Structure

- *(dict)* –

Describes a cluster.

- **Cluster** *(dict)* –

Describes a cluster.

- * **ClusterIdentifier** *(string)* –

The unique identifier of the cluster.

- * **NodeType** *(string)* –

The node type for the nodes in the cluster.

- * **ClusterStatus** *(string)* –

The current state of this cluster. Possible values include available , creating , deleting , rebooting , renaming , and resizing .

- * **ModifyStatus** *(string)* –

The status of a modify operation, if any, initiated for the cluster.

* **MasterUsername** (*string*) –

The master user name for the cluster. This name is used to connect to the database that is specified in **DBName**.

* **DBName** (*string*) –

The name of the initial database that was created when the cluster was created. This same name is returned for the life of the cluster. If an initial database was not specified, a database named “dev” was created by default.

* **Endpoint** (*dict*) –

The connection endpoint.

· **Address** (*string*) –

The DNS address of the Cluster.

· **Port** (*integer*) –

The port that the database engine is listening on.

* **ClusterCreateTime** (*datetime*) –

The date and time that the cluster was created.

* **AutomatedSnapshotRetentionPeriod** (*integer*) –

The number of days that automatic cluster snapshots are retained.

* **ClusterSecurityGroups** (*list*) –

A list of cluster security group that are associated with the cluster. Each security group is represented by an element that contains `ClusterSecurityGroup.Name` and `ClusterSecurityGroup.Status` subelements.

Cluster security groups are used when the cluster is not created in a VPC. Clusters that are created in a VPC use VPC security groups, which are listed by the **VpcSecurityGroups** parameter.

· (*dict*) –

Describes a security group.

· **ClusterSecurityGroupName** (*string*) –

The name of the cluster security group.

· **Status** (*string*) –

The status of the cluster security group.

* **VpcSecurityGroups** (*list*) –

A list of Virtual Private Cloud (VPC) security groups that are associated with the cluster. This parameter is returned only if the cluster is in a VPC.

· (*dict*) –

Describes the members of a VPC security group.

· **VpcSecurityGroupId** (*string*) –

· **Status** (*string*) –

* **ClusterParameterGroups** (*list*) –

The list of cluster parameter groups that are associated with this cluster. Each parameter group in the list is returned with its status.

· (*dict*) –

Describes the status of a parameter group.

- **ParameterGroupName** (*string*) –

The name of the cluster parameter group.

- **ParameterApplyStatus** (*string*) –

The status of parameter updates.

- **ClusterParameterStatusList** (*list*) –

The list of parameter statuses.

For more information about parameters and parameter groups, go to [Amazon Redshift Parameter Groups](#) in the *Amazon Redshift Cluster Management Guide*.

- (*dict*) –

Describes the status of a parameter group.

- **ParameterName** (*string*) –

The name of the parameter.

- **ParameterApplyStatus** (*string*) –

The status of the parameter that indicates whether the parameter is in sync with the database, waiting for a cluster reboot, or encountered an error when being applied.

The following are possible statuses and descriptions.

- `in-sync` : The parameter value is in sync with the database.
- `pending-reboot` : The parameter value will be applied after the cluster reboots.
- `applying` : The parameter value is being applied to the database.
- `invalid-parameter` : Cannot apply the parameter value because it has an invalid value or syntax.
- `apply-deferred` : The parameter contains static property changes. The changes are deferred until the cluster reboots.
- `apply-error` : Cannot connect to the cluster. The parameter change will be applied after the cluster reboots.
- `unknown-error` : Cannot apply the parameter change right now. The change will be applied after the cluster reboots.

- **ParameterApplyErrorDescription** (*string*) –

The error that prevented the parameter from being applied to the database.

- * **ClusterSubnetGroupName** (*string*) –

The name of the subnet group that is associated with the cluster. This parameter is valid only when the cluster is in a VPC.

- * **VpcId** (*string*) –

The identifier of the VPC the cluster is in, if the cluster is in a VPC.

- * **AvailabilityZone** (*string*) –

The name of the Availability Zone in which the cluster is located.

- * **PreferredMaintenanceWindow** (*string*) –

The weekly time range (in UTC) during which system maintenance can occur.

- * **PendingModifiedValues** (*dict*) –

If present, changes to the cluster are pending. Specific pending changes are identified by subelements.

- **MasterUserPassword** (*string*) –

The pending or in-progress change of the master user password for the cluster.

- **NodeType** (*string*) –

The pending or in-progress change of the cluster's node type.

- **NumberOfNodes** (*integer*) –

The pending or in-progress change of the number of nodes in the cluster.

- **ClusterType** (*string*) –

The pending or in-progress change of the cluster type.

- **ClusterVersion** (*string*) –

The pending or in-progress change of the service version.

- **AutomatedSnapshotRetentionPeriod** (*integer*) –

The pending or in-progress change of the automated snapshot retention period.

- **ClusterIdentifier** (*string*) –

The pending or in-progress change of the new identifier for the cluster.

- * **ClusterVersion** (*string*) –

The version ID of the Amazon Redshift engine that is running on the cluster.

- * **AllowVersionUpgrade** (*boolean*) –

If `true`, major version upgrades will be applied automatically to the cluster during the maintenance window.

- * **NumberOfNodes** (*integer*) –

The number of compute nodes in the cluster.

- * **PubliclyAccessible** (*boolean*) –

If `true`, the cluster can be accessed from a public network.

- * **Encrypted** (*boolean*) –

If `true`, data in the cluster is encrypted at rest.

- * **RestoreStatus** (*dict*) –

Describes the status of a cluster restore action. Returns null if the cluster was not created by restoring a snapshot.

- **Status** (*string*) –

The status of the restore action. Returns starting, restoring, completed, or failed.

- **CurrentRestoreRateInMegaBytesPerSecond** (*float*) –

The number of megabytes per second being transferred from the backup storage. Returns the average rate for a completed backup.

- **SnapshotSizeInMegaBytes** (*integer*) –

The size of the set of snapshot data used to restore the cluster.

- **ProgressInMegaBytes** (*integer*) –

The number of megabytes that have been transferred from snapshot storage.

- **ElapsedTimeInSeconds** (*integer*) –

The amount of time an in-progress restore has been running, or the amount of time it took a completed restore to finish.

- **EstimatedTimeToCompletionInSeconds** (*integer*) –

The estimate of the time remaining before the restore will complete. Returns 0 for a completed restore.

- * **HsmStatus** (*dict*) –

Reports whether the Amazon Redshift cluster has finished applying any HSM settings changes specified in a modify cluster command.

Values: active, applying

- **HsmClientCertificateIdentifier** (*string*) –

Specifies the name of the HSM client certificate the Amazon Redshift cluster uses to retrieve the data encryption keys stored in an HSM.

- **HsmConfigurationIdentifier** (*string*) –

Specifies the name of the HSM configuration that contains the information the Amazon Redshift cluster can use to retrieve and store keys in an HSM.

- **Status** (*string*) –

Reports whether the Amazon Redshift cluster has finished applying any HSM settings changes specified in a modify cluster command.

Values: active, applying

- * **ClusterSnapshotCopyStatus** (*dict*) –

Returns the destination region and retention period that are configured for cross-region snapshot copy.

- **DestinationRegion** (*string*) –

The destination region that snapshots are automatically copied to when cross-region snapshot copy is enabled.

- **RetentionPeriod** (*integer*) –

The number of days that automated snapshots are retained in the destination region after they are copied from a source region.

- **SnapshotCopyGrantName** (*string*) –

The name of the snapshot copy grant.

- * **ClusterPublicKey** (*string*) –

The public key for the cluster.

- * **ClusterNodes** (*list*) –

The nodes in a cluster.

- (*dict*) –

The identifier of a node in a cluster.

- **NodeRole** (*string*) –

Whether the node is a leader node or a compute node.

- **PrivateIPAddress** (*string*) –
The private IP address of a node within a cluster.
- **PublicIPAddress** (*string*) –
The public IP address of a node within a cluster.
- * **ElasticIpStatus** (*dict*) –
Describes the status of the elastic IP (EIP) address.
 - **ElasticIp** (*string*) –
The elastic IP (EIP) address for the cluster.
 - **Status** (*string*) –
Describes the status of the elastic IP (EIP) address.
- * **ClusterRevisionNumber** (*string*) –
The specific revision number of the database in the cluster.
- * **Tags** (*list*) –
The list of tags for the cluster.
 - (*dict*) –
A tag consisting of a name/value pair for a resource.
 - **Key** (*string*) –
The key, or name, for the resource tag.
 - **Value** (*string*) –
The value for the resource tag.
- * **KmsKeyId** (*string*) –
The AWS Key Management Service (KMS) key ID of the encryption key used to encrypt data in the cluster.

Paginators

The available paginators are:

- `Redshift.Paginator.DescribeClusterParameterGroups`
- `Redshift.Paginator.DescribeClusterParameters`
- `Redshift.Paginator.DescribeClusterSecurityGroups`
- `Redshift.Paginator.DescribeClusterSnapshots`
- `Redshift.Paginator.DescribeClusterSubnetGroups`
- `Redshift.Paginator.DescribeClusterVersions`
- `Redshift.Paginator.DescribeClusters`
- `Redshift.Paginator.DescribeDefaultClusterParameters`
- `Redshift.Paginator.DescribeEventSubscriptions`
- `Redshift.Paginator.DescribeEvents`
- `Redshift.Paginator.DescribeHsmClientCertificates`
- `Redshift.Paginator.DescribeHsmConfigurations`
- `Redshift.Paginator.DescribeOrderableClusterOptions`

- `Redshift.Paginator.DescribeReservedNodeOfferings`
- `Redshift.Paginator.DescribeReservedNodes`

class `Redshift.Paginator.DescribeClusterParameterGroups`

```
paginator = client.get_paginator('describe_cluster_parameter_groups')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Redshift.Client.describe_cluster_parameter_groups()`.

Request Syntax

```
response_iterator = paginator.paginate(
    ParameterGroupName='string',
    TagKeys=[
        'string',
    ],
    TagValues=[
        'string',
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **ParameterGroupName** (*string*) – The name of a specific parameter group for which to return details. By default, details about all parameter groups and the default parameter group are returned.
- **TagKeys** (*list*) – A tag key or keys for which you want to return all matching cluster parameter groups that are associated with the specified key or keys. For example, suppose that you have parameter groups that are tagged with keys called `owner` and `environment`. If you specify both of these tag keys in the request, Amazon Redshift returns a response with the parameter groups that have either or both of these tag keys associated with them.
 - (*string*) –
- **TagValues** (*list*) – A tag value or values for which you want to return all matching cluster parameter groups that are associated with the specified tag value or values. For example, suppose that you have parameter groups that are tagged with values called `admin` and `test`. If you specify both of these tag values in the request, Amazon Redshift returns a response with the parameter groups that have either or both of these tag values associated with them.
 - (*string*) –
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'ParameterGroups': [
    {
      'ParameterGroupName': 'string',
      'ParameterGroupFamily': 'string',
      'Description': 'string',
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        },
      ],
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output from the `DescribeClusterParameterGroups` action.

- **ParameterGroups** (*list*) –

A list of `ClusterParameterGroup` instances. Each instance describes one cluster parameter group.

- * (*dict*) –

Describes a parameter group.

- **ParameterGroupName** (*string*) –

The name of the cluster parameter group.

- **ParameterGroupFamily** (*string*) –

The name of the cluster parameter group family that this cluster parameter group is compatible with.

- **Description** (*string*) –

The description of the parameter group.

- **Tags** (*list*) –

The list of tags for the cluster parameter group.

- (*dict*) –

A tag consisting of a name/value pair for a resource.

- **Key** (*string*) –

The key, or name, for the resource tag.

- **Value** (*string*) –

The value for the resource tag.

- **NextToken** (*string*) –

A token to resume pagination.

class `Redshift.Paginator.DescribeClusterParameters`

```
paginator = client.get_paginator('describe_cluster_parameters')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Redshift.Client.describe_cluster_parameters()`.

Request Syntax

```
response_iterator = paginator.paginate(
    ParameterGroupName='string',
    Source='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **ParameterGroupName** (*string*) – [REQUIRED]

The name of a cluster parameter group for which to return details.

- **Source** (*string*) – The parameter types to return. Specify `user` to show parameters that are different from the default. Similarly, specify `engine-default` to show parameters that are the same as the default parameter group.

Default: All parameter types returned.

Valid Values: `user` | `engine-default`

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
  'Parameters': [
    {
      'ParameterName': 'string',
      'ParameterValue': 'string',
      'Description': 'string',
      'Source': 'string',
      'DataType': 'string',
```

```
        'AllowedValues': 'string',
        'ApplyType': 'static'|'dynamic',
        'IsModifiable': True|False,
        'MinimumEngineVersion': 'string'
    },
],
'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output from the DescribeClusterParameters action.

- **Parameters** (*list*) –

A list of Parameter instances. Each instance lists the parameters of one cluster parameter group.

- * (*dict*) –

Describes a parameter in a cluster parameter group.

- **ParameterName** (*string*) –

The name of the parameter.

- **ParameterValue** (*string*) –

The value of the parameter.

- **Description** (*string*) –

A description of the parameter.

- **Source** (*string*) –

The source of the parameter value, such as “engine-default” or “user”.

- **DataType** (*string*) –

The data type of the parameter.

- **AllowedValues** (*string*) –

The valid range of values for the parameter.

- **ApplyType** (*string*) –

Specifies how to apply the parameter. Supported value: `static`.

- **IsModifiable** (*boolean*) –

If `true`, the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.

- **MinimumEngineVersion** (*string*) –

The earliest engine version to which the parameter can apply.

- **NextToken** (*string*) –

A token to resume pagination.

class Redshift.Paginator.**DescribeClusterSecurityGroups**

```
paginator = client.get_paginator('describe_cluster_security_groups')
```

paginate (**kwargs)

Creates an iterator that will paginate through responses from `Redshift.Client.describe_cluster_security_groups()`.

Request Syntax

```
response_iterator = paginator.paginate(
    ClusterSecurityGroupName='string',
    TagKeys=[
        'string',
    ],
    TagValues=[
        'string',
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **ClusterSecurityGroupName** (*string*) – The name of a cluster security group for which you are requesting details. You can specify either the **Marker** parameter or a **ClusterSecurityGroupName** parameter, but not both.

Example: `securitygroup1`

- **TagKeys** (*list*) – A tag key or keys for which you want to return all matching cluster security groups that are associated with the specified key or keys. For example, suppose that you have security groups that are tagged with keys called `owner` and `environment`. If you specify both of these tag keys in the request, Amazon Redshift returns a response with the security groups that have either or both of these tag keys associated with them.
 - (*string*) –
- **TagValues** (*list*) – A tag value or values for which you want to return all matching cluster security groups that are associated with the specified tag value or values. For example, suppose that you have security groups that are tagged with values called `admin` and `test`. If you specify both of these tag values in the request, Amazon Redshift returns a response with the security groups that have either or both of these tag values associated with them.
 - (*string*) –
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict
Returns

Response Syntax

```
{
  'ClusterSecurityGroups': [
    {
      'ClusterSecurityGroupName': 'string',
      'Description': 'string',
      'EC2SecurityGroups': [
        {
          'Status': 'string',
          'EC2SecurityGroupName': 'string',
          'EC2SecurityGroupOwnerId': 'string',
          'Tags': [
            {
              'Key': 'string',
              'Value': 'string'
            }
          ],
        },
      ],
    },
  ],
  'IPRanges': [
    {
      'Status': 'string',
      'CIDRIP': 'string',
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        }
      ],
    },
  ],
  'Tags': [
    {
      'Key': 'string',
      'Value': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output from the DescribeClusterSecurityGroups action.

- **ClusterSecurityGroups** (*list*) –

A list of ClusterSecurityGroup instances.

- * (*dict*) –

Describes a security group.

- **ClusterSecurityGroupName** (*string*) –

The name of the cluster security group to which the operation was applied.

- **Description** (*string*) –

A description of the security group.

- **EC2SecurityGroups** (*list*) –
A list of EC2 security groups that are permitted to access clusters associated with this cluster security group.
- (*dict*) –
Describes an Amazon EC2 security group.
- **Status** (*string*) –
The status of the EC2 security group.
- **EC2SecurityGroupName** (*string*) –
The name of the EC2 Security Group.
- **EC2SecurityGroupOwnerId** (*string*) –
The AWS ID of the owner of the EC2 security group specified in the `EC2SecurityGroupName` field.
- **Tags** (*list*) –
The list of tags for the EC2 security group.
- (*dict*) –
A tag consisting of a name/value pair for a resource.
- **Key** (*string*) –
The key, or name, for the resource tag.
- **Value** (*string*) –
The value for the resource tag.
- **IPRanges** (*list*) –
A list of IP ranges (CIDR blocks) that are permitted to access clusters associated with this cluster security group.
- (*dict*) –
Describes an IP range used in a security group.
- **Status** (*string*) –
The status of the IP range, for example, “authorized”.
- **CIDRIP** (*string*) –
The IP range in Classless Inter-Domain Routing (CIDR) notation.
- **Tags** (*list*) –
The list of tags for the IP range.
- (*dict*) –
A tag consisting of a name/value pair for a resource.
- **Key** (*string*) –
The key, or name, for the resource tag.
- **Value** (*string*) –
The value for the resource tag.
- **Tags** (*list*) –
The list of tags for the cluster security group.
- (*dict*) –
A tag consisting of a name/value pair for a resource.
- **Key** (*string*) –
The key, or name, for the resource tag.

- **Value** (*string*) –
The value for the resource tag.
- **NextToken** (*string*) –
A token to resume pagination.

class Redshift.Paginator.**DescribeClusterSnapshots**

```
paginator = client.get_paginator('describe_cluster_snapshots')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Redshift.Client.describe_cluster_snapshots()`.

Request Syntax

```
response_iterator = paginator.paginate(
    ClusterIdentifier='string',
    SnapshotIdentifier='string',
    SnapshotType='string',
    StartTime=datetime(2015, 1, 1),
    EndTime=datetime(2015, 1, 1),
    OwnerAccount='string',
    TagKeys=[
        'string',
    ],
    TagValues=[
        'string',
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **ClusterIdentifier** (*string*) – The identifier of the cluster for which information about snapshots is requested.
- **SnapshotIdentifier** (*string*) – The snapshot identifier of the snapshot about which to return information.
- **SnapshotType** (*string*) – The type of snapshots for which you are requesting information. By default, snapshots of all types are returned.

Valid Values: `automated|manual`

- **StartTime** (*datetime*) – A value that requests only snapshots created at or after the specified time. The time value is specified in ISO 8601 format. For more information about ISO 8601, go to the [ISO8601 Wikipedia page](#).

Example: `2012-07-16T18:00:00Z`

- **EndTime** (*datetime*) – A time value that requests only snapshots created at or before the specified time. The time value is specified in ISO 8601 format. For more information about ISO 8601, go to the [ISO8601 Wikipedia page](#).

Example: `2012-07-16T18:00:00Z`

- **OwnerAccount** (*string*) – The AWS customer account used to create or copy the snapshot. Use this field to filter the results to snapshots owned by a particular

account. To describe snapshots you own, either specify your AWS customer account, or do not specify the parameter.

- **TagKeys** (*list*) – A tag key or keys for which you want to return all matching cluster snapshots that are associated with the specified key or keys. For example, suppose that you have snapshots that are tagged with keys called `owner` and `environment`. If you specify both of these tag keys in the request, Amazon Redshift returns a response with the snapshots that have either or both of these tag keys associated with them.
 - (*string*) –
- **TagValues** (*list*) – A tag value or values for which you want to return all matching cluster snapshots that are associated with the specified tag value or values. For example, suppose that you have snapshots that are tagged with values called `admin` and `test`. If you specify both of these tag values in the request, Amazon Redshift returns a response with the snapshots that have either or both of these tag values associated with them.
 - (*string*) –
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Snapshots': [
    {
      'SnapshotIdentifier': 'string',
      'ClusterIdentifier': 'string',
      'SnapshotCreateTime': datetime(2015, 1, 1),
      'Status': 'string',
      'Port': 123,
      'AvailabilityZone': 'string',
      'ClusterCreateTime': datetime(2015, 1, 1),
      'MasterUsername': 'string',
      'ClusterVersion': 'string',
      'SnapshotType': 'string',
      'NodeType': 'string',
      'NumberOfNodes': 123,
      'DBName': 'string',
      'VpcId': 'string',
      'Encrypted': True|False,
      'KmsKeyId': 'string',
      'EncryptedWithHSM': True|False,
      'AccountsWithRestoreAccess': [
        {
```

```
        'AccountId': 'string'
    },
],
'OwnerAccount': 'string',
'TotalBackupSizeInMegaBytes': 123.0,
'ActualIncrementalBackupSizeInMegaBytes': 123.0,
'BackupProgressInMegaBytes': 123.0,
'CurrentBackupRateInMegaBytesPerSecond': 123.0,
'EstimatedSecondsToCompletion': 123,
'ElapsedTimeInSeconds': 123,
'SourceRegion': 'string',
'Tags': [
    {
        'Key': 'string',
        'Value': 'string'
    },
],
'RestorableNodeTypes': [
    'string',
]
},
],
'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output from the DescribeClusterSnapshots action.

- **Snapshots** (*list*) –

A list of Snapshot instances.

- * (*dict*) –

Describes a snapshot.

- **SnapshotIdentifier** (*string*) –

The snapshot identifier that is provided in the request.

- **ClusterIdentifier** (*string*) –

The identifier of the cluster for which the snapshot was taken.

- **SnapshotCreateTime** (*datetime*) –

The time (UTC) when Amazon Redshift began the snapshot. A snapshot contains a copy of the cluster data as of this exact time.

- **Status** (*string*) –

The snapshot status. The value of the status depends on the API operation used.

- CreateClusterSnapshot and CopyClusterSnapshot returns status as “creating”.
- DescribeClusterSnapshots returns status as “creating”, “available”, “final snapshot”, or “failed”.
- DeleteClusterSnapshot returns status as “deleted”.
- **Port** (*integer*) –

The port that the cluster is listening on.

- **AvailabilityZone** (*string*) –

- The Availability Zone in which the cluster was created.
- **ClusterCreateTime** (*datetime*) –
 - The time (UTC) when the cluster was originally created.
- **MasterUsername** (*string*) –
 - The master user name for the cluster.
- **ClusterVersion** (*string*) –
 - The version ID of the Amazon Redshift engine that is running on the cluster.
- **SnapshotType** (*string*) –
 - The snapshot type. Snapshots created using `CreateClusterSnapshot` and `CopyClusterSnapshot` will be of type “manual”.
- **NodeType** (*string*) –
 - The node type of the nodes in the cluster.
- **NumberOfNodes** (*integer*) –
 - The number of nodes in the cluster.
- **DBName** (*string*) –
 - The name of the database that was created when the cluster was created.
- **VpcId** (*string*) –
 - The VPC identifier of the cluster if the snapshot is from a cluster in a VPC. Otherwise, this field is not in the output.
- **Encrypted** (*boolean*) –
 - If `true`, the data in the snapshot is encrypted at rest.
- **KmsKeyId** (*string*) –
 - The AWS Key Management Service (KMS) key ID of the encryption key that was used to encrypt data in the cluster from which the snapshot was taken.
- **EncryptedWithHSM** (*boolean*) –
 - A boolean that indicates whether the snapshot data is encrypted using the HSM keys of the source cluster. `true` indicates that the data is encrypted using HSM keys.
- **AccountsWithRestoreAccess** (*list*) –
 - A list of the AWS customer accounts authorized to restore the snapshot. Returns `null` if no accounts are authorized. Visible only to the snapshot owner.
- (*dict*) –
 - Describes an AWS customer account authorized to restore a snapshot.
- **AccountId** (*string*) –
 - The identifier of an AWS customer account authorized to restore a snapshot.
- **OwnerAccount** (*string*) –
 - For manual snapshots, the AWS customer account used to create or copy the snapshot. For automatic snapshots, the owner of the cluster. The owner can perform all snapshot actions, such as sharing a manual snapshot.

- **TotalBackupSizeInMegaBytes** (*float*) –
The size of the complete set of backup data that would be used to restore the cluster.
- **ActualIncrementalBackupSizeInMegaBytes** (*float*) –
The size of the incremental backup.
- **BackupProgressInMegaBytes** (*float*) –
The number of megabytes that have been transferred to the snapshot backup.
- **CurrentBackupRateInMegaBytesPerSecond** (*float*) –
The number of megabytes per second being transferred to the snapshot backup. Returns 0 for a completed backup.
- **EstimatedSecondsToCompletion** (*integer*) –
The estimate of the time remaining before the snapshot backup will complete. Returns 0 for a completed backup.
- **ElapsedTimeInSeconds** (*integer*) –
The amount of time an in-progress snapshot backup has been running, or the amount of time it took a completed backup to finish.
- **SourceRegion** (*string*) –
The source region from which the snapshot was copied.
- **Tags** (*list*) –
The list of tags for the cluster snapshot.
- (*dict*) –
A tag consisting of a name/value pair for a resource.
- **Key** (*string*) –
The key, or name, for the resource tag.
- **Value** (*string*) –
The value for the resource tag.
- **RestorableNodeTypes** (*list*) –
The list of node types that this cluster snapshot is able to restore into.
- (*string*) –
- **NextToken** (*string*) –
A token to resume pagination.

class Redshift.Paginator.**DescribeClusterSubnetGroups**

```
paginator = client.get_paginator('describe_cluster_subnet_groups')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Redshift.Client.describe_cluster_subnet_groups()`.

Request Syntax

```
response_iterator = paginator.paginate(  
    ClusterSubnetGroupName='string',
```

```

TagKeys=[
    'string',
],
TagValues=[
    'string',
],
PaginationConfig={
    'MaxItems': 123,
    'PageSize': 123,
    'StartingToken': 'string'
}
)

```

Parameters

- **ClusterSubnetGroupName** (*string*) – The name of the cluster subnet group for which information is requested.
- **TagKeys** (*list*) – A tag key or keys for which you want to return all matching cluster subnet groups that are associated with the specified key or keys. For example, suppose that you have subnet groups that are tagged with keys called `owner` and `environment`. If you specify both of these tag keys in the request, Amazon Redshift returns a response with the subnet groups that have either or both of these tag keys associated with them.
 - (*string*) –
- **TagValues** (*list*) – A tag value or values for which you want to return all matching cluster subnet groups that are associated with the specified tag value or values. For example, suppose that you have subnet groups that are tagged with values called `admin` and `test`. If you specify both of these tag values in the request, Amazon Redshift returns a response with the subnet groups that have either or both of these tag values associated with them.
 - (*string*) –
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```

{
    'ClusterSubnetGroups': [
        {
            'ClusterSubnetGroupName': 'string',
            'Description': 'string',
            'VpcId': 'string',
            'SubnetGroupStatus': 'string',
            'Subnets': [

```

```
{
    {
        'SubnetIdentifier': 'string',
        'SubnetAvailabilityZone': {
            'Name': 'string'
        },
        'SubnetStatus': 'string'
    },
],
'Tags': [
    {
        'Key': 'string',
        'Value': 'string'
    },
],
},
],
'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output from the DescribeClusterSubnetGroups action.

- **ClusterSubnetGroups** (*list*) –

A list of ClusterSubnetGroup instances.

- * (*dict*) –

Describes a subnet group.

- **ClusterSubnetGroupName** (*string*) –

The name of the cluster subnet group.

- **Description** (*string*) –

The description of the cluster subnet group.

- **VpcId** (*string*) –

The VPC ID of the cluster subnet group.

- **SubnetGroupStatus** (*string*) –

The status of the cluster subnet group. Possible values are Complete, Incomplete and Invalid.

- **Subnets** (*list*) –

A list of the VPC Subnet elements.

- (*dict*) –

Describes a subnet.

- **SubnetIdentifier** (*string*) –

The identifier of the subnet.

- **SubnetAvailabilityZone** (*dict*) –

Describes an availability zone.

- **Name** (*string*) –

The name of the availability zone.

- **SubnetStatus** (*string*) –

The status of the subnet.

- **Tags** (*list*) –

The list of tags for the cluster subnet group.

· *(dict)* –

A tag consisting of a name/value pair for a resource.

· **Key** (*string*) –

The key, or name, for the resource tag.

· **Value** (*string*) –

The value for the resource tag.

– **NextToken** (*string*) –

A token to resume pagination.

class Redshift.Paginator.**DescribeClusterVersions**

```
paginator = client.get_paginator('describe_cluster_versions')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Redshift.Client.describe_cluster_versions()`.

Request Syntax

```
response_iterator = paginator.paginate(
    ClusterVersion='string',
    ClusterParameterGroupFamily='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **ClusterVersion** (*string*) – The specific cluster version to return.

Example: 1.0

- **ClusterParameterGroupFamily** (*string*) – The name of a specific cluster parameter group family to return details for.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

– **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

– **PageSize** (*integer*) –

The size of each page.

– **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'ClusterVersions': [
    {
      'ClusterVersion': 'string',
      'ClusterParameterGroupFamily': 'string',
      'Description': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output from the DescribeClusterVersions action.

- **ClusterVersions** (*list*) –

A list of Version elements.

- * (*dict*) –

Describes a cluster version, including the parameter group family and description of the version.

- **ClusterVersion** (*string*) –

The version number used by the cluster.

- **ClusterParameterGroupFamily** (*string*) –

The name of the cluster parameter group family for the cluster.

- **Description** (*string*) –

The description of the cluster version.

- **NextToken** (*string*) –

A token to resume pagination.

class Redshift.Paginator.**DescribeClusters**

```
paginator = client.get_paginator('describe_clusters')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Redshift.Client.describe_clusters()`.

Request Syntax

```
response_iterator = paginator.paginate(
    ClusterIdentifier='string',
    TagKeys=[
        'string',
    ],
    TagValues=[
        'string',
    ],
```

```

PaginationConfig={
    'MaxItems': 123,
    'PageSize': 123,
    'StartingToken': 'string'
}
)

```

Parameters

- **ClusterIdentifier** (*string*) – The unique identifier of a cluster whose properties you are requesting. This parameter is case sensitive.

The default is that all clusters defined for an account are returned.

- **TagKeys** (*list*) – A tag key or keys for which you want to return all matching clusters that are associated with the specified key or keys. For example, suppose that you have clusters that are tagged with keys called `owner` and `environment`. If you specify both of these tag keys in the request, Amazon Redshift returns a response with the clusters that have either or both of these tag keys associated with them.

– (*string*) –

- **TagValues** (*list*) – A tag value or values for which you want to return all matching clusters that are associated with the specified tag value or values. For example, suppose that you have clusters that are tagged with values called `admin` and `test`. If you specify both of these tag values in the request, Amazon Redshift returns a response with the clusters that have either or both of these tag values associated with them.

– (*string*) –

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

– **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

– **PageSize** (*integer*) –

The size of each page.

– **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```

{
    'Clusters': [
        {
            'ClusterIdentifier': 'string',
            'NodeType': 'string',
            'ClusterStatus': 'string',
            'ModifyStatus': 'string',
            'MasterUsername': 'string',
            'DBName': 'string',
            'Endpoint': {
                'Address': 'string',
                'Port': 123
            }
        }
    ]
}

```

```
    },
    'ClusterCreateTime': datetime(2015, 1, 1),
    'AutomatedSnapshotRetentionPeriod': 123,
    'ClusterSecurityGroups': [
        {
            'ClusterSecurityGroupName': 'string',
            'Status': 'string'
        },
    ],
    'VpcSecurityGroups': [
        {
            'VpcSecurityGroupId': 'string',
            'Status': 'string'
        },
    ],
    'ClusterParameterGroups': [
        {
            'ParameterGroupName': 'string',
            'ParameterApplyStatus': 'string',
            'ClusterParameterStatusList': [
                {
                    'ParameterName': 'string',
                    'ParameterApplyStatus': 'string',
                    'ParameterApplyErrorDescription': 'string'
                },
            ],
        },
    ],
    'ClusterSubnetGroupName': 'string',
    'VpcId': 'string',
    'AvailabilityZone': 'string',
    'PreferredMaintenanceWindow': 'string',
    'PendingModifiedValues': {
        'MasterUserPassword': 'string',
        'NodeType': 'string',
        'NumberOfNodes': 123,
        'ClusterType': 'string',
        'ClusterVersion': 'string',
        'AutomatedSnapshotRetentionPeriod': 123,
        'ClusterIdentifier': 'string'
    },
    'ClusterVersion': 'string',
    'AllowVersionUpgrade': True|False,
    'NumberOfNodes': 123,
    'PubliclyAccessible': True|False,
    'Encrypted': True|False,
    'RestoreStatus': {
        'Status': 'string',
        'CurrentRestoreRateInMegaBytesPerSecond': 123.0,
        'SnapshotSizeInMegaBytes': 123,
        'ProgressInMegaBytes': 123,
        'ElapsedTimeInSeconds': 123,
        'EstimatedTimeToCompletionInSeconds': 123
    },
    'HsmStatus': {
        'HsmClientCertificateIdentifier': 'string',
        'HsmConfigurationIdentifier': 'string',
        'Status': 'string'
    }
```



```

    },
    'ClusterSnapshotCopyStatus': {
        'DestinationRegion': 'string',
        'RetentionPeriod': 123,
        'SnapshotCopyGrantName': 'string'
    },
    'ClusterPublicKey': 'string',
    'ClusterNodes': [
        {
            'NodeRole': 'string',
            'PrivateIPAddress': 'string',
            'PublicIPAddress': 'string'
        },
    ],
    'ElasticIpStatus': {
        'ElasticIp': 'string',
        'Status': 'string'
    },
    'ClusterRevisionNumber': 'string',
    'Tags': [
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
    'KmsKeyId': 'string'
  },
  'NextToken': 'string'
}

```

Response Structure

- *(dict)* –

Contains the output from the DescribeClusters action.

- **Clusters** *(list)* –

A list of Cluster objects, where each object describes one cluster.

- * *(dict)* –

Describes a cluster.

- **ClusterIdentifier** *(string)* –

The unique identifier of the cluster.

- **NodeType** *(string)* –

The node type for the nodes in the cluster.

- **ClusterStatus** *(string)* –

The current state of this cluster. Possible values include `available`, `creating`, `deleting`, `rebooting`, `renaming`, and `resizing`.

- **ModifyStatus** *(string)* –

The status of a modify operation, if any, initiated for the cluster.

- **MasterUsername** *(string)* –

The master user name for the cluster. This name is used to connect to the database that is specified in **DBName**.

- **DBName** (*string*) –

The name of the initial database that was created when the cluster was created. This same name is returned for the life of the cluster. If an initial database was not specified, a database named “dev” was created by default.

- **Endpoint** (*dict*) –

The connection endpoint.

- **Address** (*string*) –

The DNS address of the Cluster.

- **Port** (*integer*) –

The port that the database engine is listening on.

- **ClusterCreateTime** (*datetime*) –

The date and time that the cluster was created.

- **AutomatedSnapshotRetentionPeriod** (*integer*) –

The number of days that automatic cluster snapshots are retained.

- **ClusterSecurityGroups** (*list*) –

A list of cluster security group that are associated with the cluster. Each security group is represented by an element that contains `ClusterSecurityGroup.Name` and `ClusterSecurityGroup.Status` subelements.

Cluster security groups are used when the cluster is not created in a VPC. Clusters that are created in a VPC use VPC security groups, which are listed by the **VpcSecurityGroups** parameter.

- (*dict*) –

Describes a security group.

- **ClusterSecurityGroupName** (*string*) –

The name of the cluster security group.

- **Status** (*string*) –

The status of the cluster security group.

- **VpcSecurityGroups** (*list*) –

A list of Virtual Private Cloud (VPC) security groups that are associated with the cluster. This parameter is returned only if the cluster is in a VPC.

- (*dict*) –

Describes the members of a VPC security group.

- **VpcSecurityGroupId** (*string*) –

- **Status** (*string*) –

- **ClusterParameterGroups** (*list*) –

The list of cluster parameter groups that are associated with this cluster. Each parameter group in the list is returned with its status.

- (*dict*) –

Describes the status of a parameter group.

- **ParameterGroupName** (*string*) –
The name of the cluster parameter group.
- **ParameterApplyStatus** (*string*) –
The status of parameter updates.
- **ClusterParameterStatusList** (*list*) –
The list of parameter statuses.

For more information about parameters and parameter groups, go to [Amazon Redshift Parameter Groups](#) in the *Amazon Redshift Cluster Management Guide*.
- (*dict*) –
Describes the status of a parameter group.
- **ParameterName** (*string*) –
The name of the parameter.
- **ParameterApplyStatus** (*string*) –
The status of the parameter that indicates whether the parameter is in sync with the database, waiting for a cluster reboot, or encountered an error when being applied.

The following are possible statuses and descriptions.
 - `in-sync` : The parameter value is in sync with the database.
 - `pending-reboot` : The parameter value will be applied after the cluster reboots.
 - `applying` : The parameter value is being applied to the database.
 - `invalid-parameter` : Cannot apply the parameter value because it has an invalid value or syntax.
 - `apply-deferred` : The parameter contains static property changes. The changes are deferred until the cluster reboots.
 - `apply-error` : Cannot connect to the cluster. The parameter change will be applied after the cluster reboots.
 - `unknown-error` : Cannot apply the parameter change right now. The change will be applied after the cluster reboots.
- **ParameterApplyErrorDescription** (*string*) –
The error that prevented the parameter from being applied to the database.
- **ClusterSubnetGroupName** (*string*) –
The name of the subnet group that is associated with the cluster. This parameter is valid only when the cluster is in a VPC.
- **VpcId** (*string*) –
The identifier of the VPC the cluster is in, if the cluster is in a VPC.
- **AvailabilityZone** (*string*) –
The name of the Availability Zone in which the cluster is located.
- **PreferredMaintenanceWindow** (*string*) –
The weekly time range (in UTC) during which system maintenance can occur.

- **PendingModifiedValues** (*dict*) –
If present, changes to the cluster are pending. Specific pending changes are identified by subelements.
- **MasterUserPassword** (*string*) –
The pending or in-progress change of the master user password for the cluster.
- **NodeType** (*string*) –
The pending or in-progress change of the cluster's node type.
- **NumberOfNodes** (*integer*) –
The pending or in-progress change of the number of nodes in the cluster.
- **ClusterType** (*string*) –
The pending or in-progress change of the cluster type.
- **ClusterVersion** (*string*) –
The pending or in-progress change of the service version.
- **AutomatedSnapshotRetentionPeriod** (*integer*) –
The pending or in-progress change of the automated snapshot retention period.
- **ClusterIdentifier** (*string*) –
The pending or in-progress change of the new identifier for the cluster.
- **ClusterVersion** (*string*) –
The version ID of the Amazon Redshift engine that is running on the cluster.
- **AllowVersionUpgrade** (*boolean*) –
If `true`, major version upgrades will be applied automatically to the cluster during the maintenance window.
- **NumberOfNodes** (*integer*) –
The number of compute nodes in the cluster.
- **PubliclyAccessible** (*boolean*) –
If `true`, the cluster can be accessed from a public network.
- **Encrypted** (*boolean*) –
If `true`, data in the cluster is encrypted at rest.
- **RestoreStatus** (*dict*) –
Describes the status of a cluster restore action. Returns null if the cluster was not created by restoring a snapshot.
- **Status** (*string*) –
The status of the restore action. Returns starting, restoring, completed, or failed.
- **CurrentRestoreRateInMegaBytesPerSecond** (*float*) –
The number of megabytes per second being transferred from the backup storage. Returns the average rate for a completed backup.
- **SnapshotSizeInMegaBytes** (*integer*) –
The size of the set of snapshot data used to restore the cluster.

- **ProgressInMegaBytes** (*integer*) –
The number of megabytes that have been transferred from snapshot storage.
- **ElapsedTimeInSeconds** (*integer*) –
The amount of time an in-progress restore has been running, or the amount of time it took a completed restore to finish.
- **EstimatedTimeToCompletionInSeconds** (*integer*) –
The estimate of the time remaining before the restore will complete. Returns 0 for a completed restore.
- **HsmStatus** (*dict*) –
Reports whether the Amazon Redshift cluster has finished applying any HSM settings changes specified in a modify cluster command.

Values: active, applying
- **HsmClientCertificateIdentifier** (*string*) –
Specifies the name of the HSM client certificate the Amazon Redshift cluster uses to retrieve the data encryption keys stored in an HSM.
- **HsmConfigurationIdentifier** (*string*) –
Specifies the name of the HSM configuration that contains the information the Amazon Redshift cluster can use to retrieve and store keys in an HSM.
- **Status** (*string*) –
Reports whether the Amazon Redshift cluster has finished applying any HSM settings changes specified in a modify cluster command.

Values: active, applying
- **ClusterSnapshotCopyStatus** (*dict*) –
Returns the destination region and retention period that are configured for cross-region snapshot copy.
- **DestinationRegion** (*string*) –
The destination region that snapshots are automatically copied to when cross-region snapshot copy is enabled.
- **RetentionPeriod** (*integer*) –
The number of days that automated snapshots are retained in the destination region after they are copied from a source region.
- **SnapshotCopyGrantName** (*string*) –
The name of the snapshot copy grant.
- **ClusterPublicKey** (*string*) –
The public key for the cluster.
- **ClusterNodes** (*list*) –
The nodes in a cluster.
- (*dict*) –
The identifier of a node in a cluster.

- **NodeRole** (*string*) –
Whether the node is a leader node or a compute node.
 - **PrivateIPAddress** (*string*) –
The private IP address of a node within a cluster.
 - **PublicIPAddress** (*string*) –
The public IP address of a node within a cluster.
 - **ElasticIpStatus** (*dict*) –
Describes the status of the elastic IP (EIP) address.
 - **ElasticIp** (*string*) –
The elastic IP (EIP) address for the cluster.
 - **Status** (*string*) –
Describes the status of the elastic IP (EIP) address.
 - **ClusterRevisionNumber** (*string*) –
The specific revision number of the database in the cluster.
 - **Tags** (*list*) –
The list of tags for the cluster.
 - (*dict*) –
A tag consisting of a name/value pair for a resource.
 - **Key** (*string*) –
The key, or name, for the resource tag.
 - **Value** (*string*) –
The value for the resource tag.
 - **KmsKeyId** (*string*) –
The AWS Key Management Service (KMS) key ID of the encryption key used to encrypt data in the cluster.
- **NextToken** (*string*) –
A token to resume pagination.

class Redshift.Paginator.**DescribeDefaultClusterParameters**

```
paginator = client.get_paginator('describe_default_cluster_parameters')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Redshift.Client.describe_default_cluster_parameters()`.

Request Syntax

```
response_iterator = paginator.paginate(  
    ParameterGroupFamily='string',  
    PaginationConfig={  
        'MaxItems': 123,  
        'PageSize': 123,  
        'StartingToken': 'string'  
    }  
)
```

Parameters

- **ParameterGroupFamily** (*string*) – [REQUIRED]

The name of the cluster parameter group family.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'DefaultClusterParameters': {
    'ParameterGroupFamily': 'string',
    'Marker': 'string',
    'Parameters': [
      {
        'ParameterName': 'string',
        'ParameterValue': 'string',
        'Description': 'string',
        'Source': 'string',
        'DataType': 'string',
        'AllowedValues': 'string',
        'ApplyType': 'static'|'dynamic',
        'IsModifiable': True|False,
        'MinimumEngineVersion': 'string'
      },
    ],
  },
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Describes the default cluster parameters for a parameter group family.

- **DefaultClusterParameters** (*dict*) –

Describes the default cluster parameters for a parameter group family.

- * **ParameterGroupFamily** (*string*) –

The name of the cluster parameter group family to which the engine default parameters apply.

- * **Marker** (*string*) –

A value that indicates the starting point for the next set of response records in a subsequent request. If a value is returned in a response, you can retrieve the next set of records by providing this returned marker value in the `Marker` parameter and retrying the command.

If the `Marker` field is empty, all response records have been retrieved for the request.

* **Parameters** (*list*) –

The list of cluster default parameters.

· (*dict*) –

Describes a parameter in a cluster parameter group.

· **ParameterName** (*string*) –

The name of the parameter.

· **ParameterValue** (*string*) –

The value of the parameter.

· **Description** (*string*) –

A description of the parameter.

· **Source** (*string*) –

The source of the parameter value, such as “engine-default” or “user”.

· **DataType** (*string*) –

The data type of the parameter.

· **AllowedValues** (*string*) –

The valid range of values for the parameter.

· **ApplyType** (*string*) –

Specifies how to apply the parameter. Supported value: `static`.

· **IsModifiable** (*boolean*) –

If `true`, the parameter can be modified. Some parameters have security or operational implications that prevent them from being changed.

· **MinimumEngineVersion** (*string*) –

The earliest engine version to which the parameter can apply.

– **NextToken** (*string*) –

A token to resume pagination.

class `Redshift.Paginator.DescribeEventSubscriptions`

```
paginator = client.get_paginator('describe_event_subscriptions')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Redshift.Client.describe_event_subscriptions()`.

Request Syntax

```
response_iterator = paginator.paginate(
    SubscriptionName='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
```



```
    }
)
```

Parameters

- **SubscriptionName** (*string*) – The name of the Amazon Redshift event notification subscription to be described.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'EventSubscriptionsList': [
    {
      'CustomerAwsId': 'string',
      'CustSubscriptionId': 'string',
      'SnsTopicArn': 'string',
      'Status': 'string',
      'SubscriptionCreationTime': datetime(2015, 1, 1),
      'SourceType': 'string',
      'SourceIdsList': [
        'string',
      ],
      'EventCategoriesList': [
        'string',
      ],
      'Severity': 'string',
      'Enabled': True|False,
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        },
      ],
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **EventSubscriptionsList** (*list*) –
A list of event subscriptions.
 - * (*dict*) –

- **CustomerAwsId** (*string*) –

The AWS customer account associated with the Amazon Redshift event notification subscription.

- **CustSubscriptionId** (*string*) –

The name of the Amazon Redshift event notification subscription.

- **SnsTopicArn** (*string*) –

The Amazon Resource Name (ARN) of the Amazon SNS topic used by the event notification subscription.

- **Status** (*string*) –

The status of the Amazon Redshift event notification subscription.

Constraints:

- Can be one of the following: active | no-permission | topic-not-exist

- The status “no-permission” indicates that Amazon Redshift no longer has permission to post to the Amazon SNS topic. The status “topic-not-exist” indicates that the topic was deleted after the subscription was created.

- **SubscriptionCreationTime** (*datetime*) –

The date and time the Amazon Redshift event notification subscription was created.

- **SourceType** (*string*) –

The source type of the events returned the Amazon Redshift event notification, such as cluster, or cluster-snapshot.

- **SourceIdsList** (*list*) –

A list of the sources that publish events to the Amazon Redshift event notification subscription.

- (*string*) –

- **EventCategoriesList** (*list*) –

The list of Amazon Redshift event categories specified in the event notification subscription.

Values: Configuration, Management, Monitoring, Security

- (*string*) –

- **Severity** (*string*) –

The event severity specified in the Amazon Redshift event notification subscription.

Values: ERROR, INFO

- **Enabled** (*boolean*) –

A Boolean value indicating whether the subscription is enabled. `true` indicates the subscription is enabled.

- **Tags** (*list*) –

The list of tags for the event subscription.

- (*dict*) –

A tag consisting of a name/value pair for a resource.

- **Key** (*string*) –
The key, or name, for the resource tag.
- **Value** (*string*) –
The value for the resource tag.
- **NextToken** (*string*) –
A token to resume pagination.

class Redshift.Paginator.**DescribeEvents**

```
paginator = client.get_paginator('describe_events')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Redshift.Client.describe_events()`.

Request Syntax

```
response_iterator = paginator.paginate(
    SourceIdentifier='string',
    SourceType='cluster'|'cluster-parameter-group'|'cluster-security-group'|'cluster-snapsh
    StartTime=datetime(2015, 1, 1),
    EndTime=datetime(2015, 1, 1),
    Duration=123,
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **SourceIdentifier** (*string*) – The identifier of the event source for which events will be returned. If this parameter is not specified, then all sources are included in the response.

Constraints:

If *SourceIdentifier* is supplied, *SourceType* must also be provided.

- Specify a cluster identifier when *SourceType* is `cluster`.
- Specify a cluster security group name when *SourceType* is `cluster-security-group`.
- Specify a cluster parameter group name when *SourceType* is `cluster-parameter-group`.
- Specify a cluster snapshot identifier when *SourceType* is `cluster-snapshot`.

- **SourceType** (*string*) – The event source to retrieve events for. If no value is specified, all events are returned.

Constraints:

If *SourceType* is supplied, *SourceIdentifier* must also be provided.

- Specify `cluster` when *SourceIdentifier* is a cluster identifier.
- Specify `cluster-security-group` when *SourceIdentifier* is a cluster security group name.
- Specify `cluster-parameter-group` when *SourceIdentifier* is a cluster parameter group name.

- Specify `cluster-snapshot` when *SourceIdentifier* is a cluster snapshot identifier.
- **StartTime** (*datetime*) – The beginning of the time interval to retrieve events for, specified in ISO 8601 format. For more information about ISO 8601, go to the [ISO8601 Wikipedia page](#).

Example: 2009-07-08T18:00Z

- **EndTime** (*datetime*) – The end of the time interval for which to retrieve events, specified in ISO 8601 format. For more information about ISO 8601, go to the [ISO8601 Wikipedia page](#).

Example: 2009-07-08T18:00Z

- **Duration** (*integer*) – The number of minutes prior to the time of the request for which to retrieve events. For example, if the request is sent at 18:00 and you specify a duration of 60, then only events which have occurred after 17:00 will be returned.

Default: 60

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Events': [
    {
      'SourceIdentifier': 'string',
      'SourceType': 'cluster'|'cluster-parameter-group'|'cluster-security-gr...',
      'Message': 'string',
      'EventCategories': [
        'string',
      ],
      'Severity': 'string',
      'Date': datetime(2015, 1, 1),
      'EventId': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output from the `DescribeEvents` action.

- **Events** (*list*) –

A list of Event instances.

* (*dict*) –

Describes an event.

- **SourceIdentifier** (*string*) –

The identifier for the source of the event.

- **SourceType** (*string*) –

The source type for this event.

- **Message** (*string*) –

The text of this event.

- **EventCategories** (*list*) –

A list of the event categories.

Values: Configuration, Management, Monitoring, Security

- (*string*) –

- **Severity** (*string*) –

The severity of the event.

Values: ERROR, INFO

- **Date** (*datetime*) –

The date and time of the event.

- **EventId** (*string*) –

The identifier of the event.

- **NextToken** (*string*) –

A token to resume pagination.

class Redshift.Paginator.**DescribeHsmClientCertificates**

```
paginator = client.get_paginator('describe_hsm_client_certificates')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Redshift.Client.describe_hsm_client_certificates()`.

Request Syntax

```
response_iterator = paginator.paginate(
    HsmClientCertificateIdentifier='string',
    TagKeys=[
        'string',
    ],
    TagValues=[
        'string',
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **HsmClientCertificateIdentifier** (*string*) – The identifier of a specific HSM client certificate for which you want information. If no identifier is specified, information is returned for all HSM client certificates owned by your AWS customer account.
- **TagKeys** (*list*) – A tag key or keys for which you want to return all matching HSM client certificates that are associated with the specified key or keys. For example, suppose that you have HSM client certificates that are tagged with keys called `owner` and `environment`. If you specify both of these tag keys in the request, Amazon Redshift returns a response with the HSM client certificates that have either or both of these tag keys associated with them.
 - (*string*) –
- **TagValues** (*list*) – A tag value or values for which you want to return all matching HSM client certificates that are associated with the specified tag value or values. For example, suppose that you have HSM client certificates that are tagged with values called `admin` and `test`. If you specify both of these tag values in the request, Amazon Redshift returns a response with the HSM client certificates that have either or both of these tag values associated with them.
 - (*string*) –
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –

The size of each page.
 - **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'HsmClientCertificates': [
    {
      'HsmClientCertificateIdentifier': 'string',
      'HsmClientCertificatePublicKey': 'string',
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        },
      ],
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **HsmClientCertificates** (*list*) –

A list of the identifiers for one or more HSM client certificates used by

Amazon Redshift clusters to store and retrieve database encryption keys in an HSM.

* (*dict*) –

Returns information about an HSM client certificate. The certificate is stored in a secure Hardware Storage Module (HSM), and used by the Amazon Redshift cluster to encrypt data files.

- **HsmClientCertificateIdentifier** (*string*) –

The identifier of the HSM client certificate.

- **HsmClientCertificatePublicKey** (*string*) –

The public key that the Amazon Redshift cluster will use to connect to the HSM. You must register the public key in the HSM.

- **Tags** (*list*) –

The list of tags for the HSM client certificate.

- (*dict*) –

A tag consisting of a name/value pair for a resource.

- **Key** (*string*) –

The key, or name, for the resource tag.

- **Value** (*string*) –

The value for the resource tag.

- **NextToken** (*string*) –

A token to resume pagination.

class Redshift.Paginator.**DescribeHsmConfigurations**

```
paginator = client.get_paginator('describe_hsm_configurations')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Redshift.Client.describe_hsm_configurations()`.

Request Syntax

```
response_iterator = paginator.paginate(
    HsmConfigurationIdentifier='string',
    TagKeys=[
        'string',
    ],
    TagValues=[
        'string',
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **HsmConfigurationIdentifier** (*string*) – The identifier of a specific Amazon Redshift HSM configuration to be described. If no identifier is specified, information is returned for all HSM configurations owned by your AWS customer account.
- **TagKeys** (*list*) – A tag key or keys for which you want to return all matching HSM configurations that are associated with the specified key or keys. For example, suppose that you have HSM configurations that are tagged with keys called `owner` and `environment`. If you specify both of these tag keys in the request, Amazon Redshift returns a response with the HSM configurations that have either or both of these tag keys associated with them.
 - (*string*) –
- **TagValues** (*list*) – A tag value or values for which you want to return all matching HSM configurations that are associated with the specified tag value or values. For example, suppose that you have HSM configurations that are tagged with values called `admin` and `test`. If you specify both of these tag values in the request, Amazon Redshift returns a response with the HSM configurations that have either or both of these tag values associated with them.
 - (*string*) –
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –

The size of each page.
 - **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'HsmConfigurations': [
    {
      'HsmConfigurationIdentifier': 'string',
      'Description': 'string',
      'HsmIpAddress': 'string',
      'HsmPartitionName': 'string',
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        },
      ],
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

– **HsmConfigurations** (*list*) –

A list of Amazon Redshift HSM configurations.

* (*dict*) –

Returns information about an HSM configuration, which is an object that describes to Amazon Redshift clusters the information they require to connect to an HSM where they can store database encryption keys.

· **HsmConfigurationIdentifier** (*string*) –

The name of the Amazon Redshift HSM configuration.

· **Description** (*string*) –

A text description of the HSM configuration.

· **HsmIpAddress** (*string*) –

The IP address that the Amazon Redshift cluster must use to access the HSM.

· **HsmPartitionName** (*string*) –

The name of the partition in the HSM where the Amazon Redshift clusters will store their database encryption keys.

· **Tags** (*list*) –

The list of tags for the HSM configuration.

· (*dict*) –

A tag consisting of a name/value pair for a resource.

· **Key** (*string*) –

The key, or name, for the resource tag.

· **Value** (*string*) –

The value for the resource tag.

– **NextToken** (*string*) –

A token to resume pagination.

class Redshift.Paginator.**DescribeOrderableClusterOptions**

```
paginator = client.get_paginator('describe_orderable_cluster_options')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Redshift.Client.describe_orderable_cluster_options()`.

Request Syntax

```
response_iterator = paginator.paginate(
    ClusterVersion='string',
    NodeType='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **ClusterVersion** (*string*) – The version filter value. Specify this parameter to show only the available offerings matching the specified version.

Default: All versions.

Constraints: Must be one of the version returned from DescribeClusterVersions

- **NodeType** (*string*) – The node type filter value. Specify this parameter to show only the available offerings matching the specified node type.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'OrderableClusterOptions': [
    {
      'ClusterVersion': 'string',
      'ClusterType': 'string',
      'NodeType': 'string',
      'AvailabilityZones': [
        {
          'Name': 'string'
        },
      ],
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output from the DescribeOrderableClusterOptions action.

- **OrderableClusterOptions** (*list*) –

An OrderableClusterOption structure containing information about orderable options for the Cluster.

- * (*dict*) –

Describes an orderable cluster option.

- **ClusterVersion** (*string*) –

The version of the orderable cluster.

- **ClusterType** (*string*) –

The cluster type, for example `multi-node`.

- **NodeType** (*string*) –
The node type for the orderable cluster.
- **AvailabilityZones** (*list*) –
A list of availability zones for the orderable cluster.
- (*dict*) –
Describes an availability zone.
- **Name** (*string*) –
The name of the availability zone.
- **NextToken** (*string*) –
A token to resume pagination.

class Redshift.Paginator.**DescribeReservedNodeOfferings**

```
paginator = client.get_paginator('describe_reserved_node_offerings')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Redshift.Client.describe_reserved_node_offerings()`.

Request Syntax

```
response_iterator = paginator.paginate(
    ReservedNodeOfferingId='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **ReservedNodeOfferingId** (*string*) – The unique identifier for the offering.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
    'ReservedNodeOfferings': [
        {
```

```
        'ReservedNodeOfferingId': 'string',
        'NodeType': 'string',
        'Duration': 123,
        'FixedPrice': 123.0,
        'UsagePrice': 123.0,
        'CurrencyCode': 'string',
        'OfferingType': 'string',
        'RecurringCharges': [
            {
                'RecurringChargeAmount': 123.0,
                'RecurringChargeFrequency': 'string'
            },
        ],
    },
    ],
    'NextToken': 'string'
}
```

Response Structure

- *(dict)* –

Contains the output from the DescribeReservedNodeOfferings action.

- **ReservedNodeOfferings** (*list*) –

A list of reserved node offerings.

- * *(dict)* –

Describes a reserved node offering.

- **ReservedNodeOfferingId** (*string*) –

The offering identifier.

- **NodeType** (*string*) –

The node type offered by the reserved node offering.

- **Duration** (*integer*) –

The duration, in seconds, for which the offering will reserve the node.

- **FixedPrice** (*float*) –

The upfront fixed charge you will pay to purchase the specific reserved node offering.

- **UsagePrice** (*float*) –

The rate you are charged for each hour the cluster that is using the offering is running.

- **CurrencyCode** (*string*) –

The currency code for the compute nodes offering.

- **OfferingType** (*string*) –

The anticipated utilization of the reserved node, as defined in the reserved node offering.

- **RecurringCharges** (*list*) –

The charge to your account regardless of whether you are creating any clusters using the node offering. Recurring charges are only in effect for heavy-utilization reserved nodes.

- *(dict)* –

Describes a recurring charge.

- **RecurringChargeAmount** (*float*) –

The amount charged per the period of time specified by the recurring charge frequency.

- **RecurringChargeFrequency** (*string*) –

The frequency at which the recurring charge amount is applied.

- **NextToken** (*string*) –

A token to resume pagination.

class Redshift.Paginator.**DescribeReservedNodes**

```
paginator = client.get_paginator('describe_reserved_nodes')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Redshift.Client.describe_reserved_nodes()`.

Request Syntax

```
response_iterator = paginator.paginate(
    ReservedNodeId='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **ReservedNodeId** (*string*) – Identifier for the node reservation.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –

The size of each page.
 - **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
    'ReservedNodes': [
        {
            'ReservedNodeId': 'string',
            'ReservedNodeOfferingId': 'string',
            'NodeType': 'string',
```

```
        'StartTime': datetime(2015, 1, 1),
        'Duration': 123,
        'FixedPrice': 123.0,
        'UsagePrice': 123.0,
        'CurrencyCode': 'string',
        'NodeCount': 123,
        'State': 'string',
        'OfferingType': 'string',
        'RecurringCharges': [
            {
                'RecurringChargeAmount': 123.0,
                'RecurringChargeFrequency': 'string'
            },
        ],
    },
],
'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the output from the DescribeReservedNodes action.

- **ReservedNodes** (*list*) –

The list of reserved nodes.

- * (*dict*) –

Describes a reserved node. You can call the DescribeReservedNodeOfferings API to obtain the available reserved node offerings.

- **ReservedNodeId** (*string*) –

The unique identifier for the reservation.

- **ReservedNodeOfferingId** (*string*) –

The identifier for the reserved node offering.

- **NodeType** (*string*) –

The node type of the reserved node.

- **StartTime** (*datetime*) –

The time the reservation started. You purchase a reserved node offering for a duration. This is the start time of that duration.

- **Duration** (*integer*) –

The duration of the node reservation in seconds.

- **FixedPrice** (*float*) –

The fixed cost Amazon Redshift charges you for this reserved node.

- **UsagePrice** (*float*) –

The hourly rate Amazon Redshift charges you for this reserved node.

- **CurrencyCode** (*string*) –

The currency code for the reserved cluster.

- **NodeCount** (*integer*) –

The number of reserved compute nodes.

- **State** (*string*) –

The state of the reserved compute node.

Possible Values:

- pending-payment-This reserved node has recently been purchased, and the sale has been approved, but payment has not yet been confirmed.
- active-This reserved node is owned by the caller and is available for use.
- payment-failed-Payment failed for the purchase attempt.
- **OfferingType** (*string*) –

The anticipated utilization of the reserved node, as defined in the reserved node offering.

- **RecurringCharges** (*list*) –

The recurring charges for the reserved node.

- (*dict*) –

Describes a recurring charge.

- **RecurringChargeAmount** (*float*) –

The amount charged per the period of time specified by the recurring charge frequency.

- **RecurringChargeFrequency** (*string*) –

The frequency at which the recurring charge amount is applied.

- **NextToken** (*string*) –

A token to resume pagination.

Waiters

The available waiters are:

- `Redshift.Waiter.ClusterAvailable`
- `Redshift.Waiter.ClusterDeleted`
- `Redshift.Waiter.SnapshotAvailable`

class `Redshift.Waiter.ClusterAvailable`

```
waiter = client.get_waiter('cluster_available')
```

wait (***kwargs*)

Polls `Redshift.Client.describe_clusters()` every 60 seconds until a successful state is reached. An error is returned after 30 failed checks.

Request Syntax

```
waiter.wait(
    ClusterIdentifier='string',
    MaxRecords=123,
    Marker='string',
    TagKeys=[
```

```
        'string',
    ],
    TagValues=[
        'string',
    ]
)
```

Parameters

- **ClusterIdentifier** (*string*) – The unique identifier of a cluster whose properties you are requesting. This parameter is case sensitive.

The default is that all clusters defined for an account are returned.

- **MaxRecords** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified `MaxRecords` value, a value is returned in a `marker` field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: 100

Constraints: minimum 20, maximum 100.

- **Marker** (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeClusters` request exceed the value specified in `MaxRecords`, AWS returns a value in the `Marker` field of the response. You can retrieve the next set of response records by providing the returned marker value in the `Marker` parameter and retrying the request.

Constraints: You can specify either the **ClusterIdentifier** parameter or the **Marker** parameter, but not both.

- **TagKeys** (*list*) – A tag key or keys for which you want to return all matching clusters that are associated with the specified key or keys. For example, suppose that you have clusters that are tagged with keys called `owner` and `environment`. If you specify both of these tag keys in the request, Amazon Redshift returns a response with the clusters that have either or both of these tag keys associated with them.

– (*string*) –

- **TagValues** (*list*) – A tag value or values for which you want to return all matching clusters that are associated with the specified tag value or values. For example, suppose that you have clusters that are tagged with values called `admin` and `test`. If you specify both of these tag values in the request, Amazon Redshift returns a response with the clusters that have either or both of these tag values associated with them.

– (*string*) –

Returns None

class `Redshift.Waiter.ClusterDeleted`

```
waiter = client.get_waiter('cluster_deleted')
```

wait (***kwargs*)

Polls `Redshift.Client.describe_clusters()` every 60 seconds until a successful state is reached. An error is returned after 30 failed checks.

Request Syntax


```

waiter.wait(
    ClusterIdentifier='string',
    MaxRecords=123,
    Marker='string',
    TagKeys=[
        'string',
    ],
    TagValues=[
        'string',
    ]
)

```

Parameters

- **ClusterIdentifier** (*string*) – The unique identifier of a cluster whose properties you are requesting. This parameter is case sensitive.

The default is that all clusters defined for an account are returned.

- **MaxRecords** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified `MaxRecords` value, a value is returned in a `marker` field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: 100

Constraints: minimum 20, maximum 100.

- **Marker** (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeClusters` request exceed the value specified in `MaxRecords`, AWS returns a value in the `Marker` field of the response. You can retrieve the next set of response records by providing the returned marker value in the `Marker` parameter and retrying the request.

Constraints: You can specify either the **ClusterIdentifier** parameter or the **Marker** parameter, but not both.

- **TagKeys** (*list*) – A tag key or keys for which you want to return all matching clusters that are associated with the specified key or keys. For example, suppose that you have clusters that are tagged with keys called `owner` and `environment`. If you specify both of these tag keys in the request, Amazon Redshift returns a response with the clusters that have either or both of these tag keys associated with them.

– (*string*) –

- **TagValues** (*list*) – A tag value or values for which you want to return all matching clusters that are associated with the specified tag value or values. For example, suppose that you have clusters that are tagged with values called `admin` and `test`. If you specify both of these tag values in the request, Amazon Redshift returns a response with the clusters that have either or both of these tag values associated with them.

– (*string*) –

Returns None

```
class Redshift.Waiter.SnapshotAvailable
```

```

waiter = client.get_waiter('snapshot_available')

wait(**kwargs)

```

Polls `Redshift.Client.describe_cluster_snapshots()` every 15 seconds until a successful state is reached. An error is returned after 20 failed checks.

Request Syntax

```
waiter.wait(
    ClusterIdentifier='string',
    SnapshotIdentifier='string',
    SnapshotType='string',
    StartTime=datetime(2015, 1, 1),
    EndTime=datetime(2015, 1, 1),
    MaxRecords=123,
    Marker='string',
    OwnerAccount='string',
    TagKeys=[
        'string',
    ],
    TagValues=[
        'string',
    ]
)
```

Parameters

- **ClusterIdentifier** (*string*) – The identifier of the cluster for which information about snapshots is requested.
- **SnapshotIdentifier** (*string*) – The snapshot identifier of the snapshot about which to return information.
- **SnapshotType** (*string*) – The type of snapshots for which you are requesting information. By default, snapshots of all types are returned.

Valid Values: `automated|manual`

- **StartTime** (*datetime*) – A value that requests only snapshots created at or after the specified time. The time value is specified in ISO 8601 format. For more information about ISO 8601, go to the [ISO8601 Wikipedia page](#).

Example: `2012-07-16T18:00:00Z`

- **EndTime** (*datetime*) – A time value that requests only snapshots created at or before the specified time. The time value is specified in ISO 8601 format. For more information about ISO 8601, go to the [ISO8601 Wikipedia page](#).

Example: `2012-07-16T18:00:00Z`

- **MaxRecords** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified `MaxRecords` value, a value is returned in a `marker` field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: 100

Constraints: minimum 20, maximum 100.

- **Marker** (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeClusterSnapshots` request exceed the value specified in `MaxRecords`, AWS returns a value in the `Marker` field of the response. You can retrieve the next set of response records by providing the returned marker value in the `Marker` parameter and retrying the request.
- **OwnerAccount** (*string*) – The AWS customer account used to create or copy the snapshot. Use this field to filter the results to snapshots owned by a particular

account. To describe snapshots you own, either specify your AWS customer account, or do not specify the parameter.

- **TagKeys** (*list*) – A tag key or keys for which you want to return all matching cluster snapshots that are associated with the specified key or keys. For example, suppose that you have snapshots that are tagged with keys called `owner` and `environment`. If you specify both of these tag keys in the request, Amazon Redshift returns a response with the snapshots that have either or both of these tag keys associated with them.
 - (*string*) –
- **TagValues** (*list*) – A tag value or values for which you want to return all matching cluster snapshots that are associated with the specified tag value or values. For example, suppose that you have snapshots that are tagged with values called `admin` and `test`. If you specify both of these tag values in the request, Amazon Redshift returns a response with the snapshots that have either or both of these tag values associated with them.
 - (*string*) –

Returns None

Route53

Table of Contents

- [Route53](#)
 - [Client](#)
 - [Paginators](#)
 - [Waiters](#)

Client

`class Route53.Client`

A low-level client representing Amazon Route 53:

```
import boto3
```

```
client = boto3.client('route53')
```

These are the available methods:

- `associate_vpc_with_hosted_zone()`
- `can_paginate()`
- `change_resource_record_sets()`
- `change_tags_for_resource()`
- `create_health_check()`
- `create_hosted_zone()`
- `create_reusable_delegation_set()`
- `delete_health_check()`
- `delete_hosted_zone()`
- `delete_reusable_delegation_set()`
- `disassociate_vpc_from_hosted_zone()`
- `generate_presigned_url()`
- `get_change()`
- `get_checker_ip_ranges()`

- `get_geo_location()`
- `get_health_check()`
- `get_health_check_count()`
- `get_health_check_last_failure_reason()`
- `get_health_check_status()`
- `get_hosted_zone()`
- `get_hosted_zone_count()`
- `get_paginator()`
- `get_reusable_delegation_set()`
- `get_waiter()`
- `list_geo_locations()`
- `list_health_checks()`
- `list_hosted_zones()`
- `list_hosted_zones_by_name()`
- `list_resource_record_sets()`
- `list_reusable_delegation_sets()`
- `list_tags_for_resource()`
- `list_tags_for_resources()`
- `update_health_check()`
- `update_hosted_zone_comment()`

associate_vpc_with_hosted_zone (***kwargs*)

This action associates a VPC with an hosted zone.

To associate a VPC with an hosted zone, send a POST request to the `2013-04-01/hostedzone/*hosted zone ID* /associatevpc` resource. The request body must include an XML document with a `AssociateVPCWithHostedZoneRequest` element. The response returns the `AssociateVPCWithHostedZoneResponse` element that contains `ChangeInfo` for you to track the progress of the `AssociateVPCWithHostedZoneRequest` you made. See `GetChange` operation for how to track the progress of your change.

Request Syntax

```
response = client.associate_vpc_with_hosted_zone(
    HostedZoneId='string',
    VPC={
        'VPCRegion': 'us-east-1'|'us-west-1'|'us-west-2'|'eu-west-1'|'eu-central-1'|'ap-sou',
        'VPCId': 'string'
    },
    Comment='string'
)
```

Parameters

- **HostedZoneId** (*string*) – [REQUIRED]

The ID of the hosted zone you want to associate your VPC with.

Note that you cannot associate a VPC with a hosted zone that doesn't have an existing VPC association.

- **VPC** (*dict*) – [REQUIRED]

The VPC that you want your hosted zone to be associated with.

- **VPCRegion** (*string*) –
- **VPCId** (*string*) –

A VPC ID

- **Comment** (*string*) – *Optional:* Any comments you want to include about a `AssociateVPCWithHostedZoneRequest`.

Return type dict

Returns

Response Syntax

```
{
  'ChangeInfo': {
    'Id': 'string',
    'Status': 'PENDING'|'INSYNC',
    'SubmittedAt': datetime(2015, 1, 1),
    'Comment': 'string'
  }
}
```

Response Structure

- (*dict*) –

A complex type containing the response information for the request.

- **ChangeInfo** (*dict*) –

A complex type that contains the ID, the status, and the date and time of your AssociateVPCWithHostedZoneRequest .

- * **Id** (*string*) –

The ID of the request. Use this ID to track when the change has completed across all Amazon Route 53 DNS servers.

- * **Status** (*string*) –

The current state of the request. PENDING indicates that this request has not yet been applied to all Amazon Route 53 DNS servers.

Valid Values: PENDING | INSYNC

- * **SubmittedAt** (*datetime*) –

The date and time the change was submitted, in the format YYYY-MM-DDThh:mm:ssZ , as specified in the ISO 8601 standard (for example, 2009-11-19T19:37:58Z). The Z after the time indicates that the time is listed in Coordinated Universal Time (UTC), which is synonymous with Greenwich Mean Time in this context.

- * **Comment** (*string*) –

A complex type that describes change information about changes made to your hosted zone.

This element contains an ID that you use when performing a GetChange action to get detailed information about the change.

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

change_resource_record_sets (***kwargs*)

Use this action to create or change your authoritative DNS information. To use this action, send a POST request to the `2013-04-01/hostedzone/*hosted Zone ID* /rrset` resource. The request body must include an XML document with a `ChangeResourceRecordSetsRequest` element.

Changes are a list of change items and are considered transactional. For more information on transactional changes, also known as change batches, see [Creating, Changing, and Deleting Resource Record Sets Using the Route 53 API](#) in the *Amazon Route 53 Developer Guide* .

Warning: Due to the nature of transactional changes, you cannot delete the same resource record set more than once in a single change batch. If you attempt to delete the same change batch more than once, Route 53 returns an `InvalidChangeBatch` error.

In response to a `ChangeResourceRecordSets` request, your DNS data is changed on all Route 53 DNS servers. Initially, the status of a change is `PENDING` . This means the change has not yet propagated to all the authoritative Route 53 DNS servers. When the change is propagated to all hosts, the change returns a status of `INSYNC` .

Note the following limitations on a `ChangeResourceRecordSets` request:

- A request cannot contain more than 100 Change elements.
- A request cannot contain more than 1000 ResourceRecord elements.

The sum of the number of characters (including spaces) in all Value elements in a request cannot exceed 32,000 characters.

Request Syntax

```
response = client.change_resource_record_sets(
    HostedZoneId='string',
    ChangeBatch={
        'Comment': 'string',
        'Changes': [
            {
                'Action': 'CREATE'|'DELETE'|'UPSERT',
                'ResourceRecordSet': {
                    'Name': 'string',
                    'Type': 'SOA'|'A'|'TXT'|'NS'|'CNAME'|'MX'|'PTR'|'SRV'|'SPF'|'AAAA',
                    'SetIdentifier': 'string',
                    'Weight': 123,
                    'Region': 'us-east-1'|'us-west-1'|'us-west-2'|'eu-west-1'|'eu-central-1',
                    'GeoLocation': {
                        'ContinentCode': 'string',
                        'CountryCode': 'string',
                        'SubdivisionCode': 'string'
                    },
                    'Failover': 'PRIMARY'|'SECONDARY',
                    'TTL': 123,
                    'ResourceRecords': [
                        {
                            'Value': 'string'
                        },
                    ],
                    'AliasTarget': {
                        'HostedZoneId': 'string',
                        'DNSName': 'string',
                        'EvaluateTargetHealth': True|False
                    },
                    'HealthCheckId': 'string'
                },
            },
        ],
    },
)
```

Parameters

- **HostedZoneId** (*string*) – [REQUIRED]

The ID of the hosted zone that contains the resource record sets that you want to change.

- **ChangeBatch** (*dict*) – [REQUIRED]

A complex type that contains an optional comment and the `Changes` element.

- **Comment** (*string*) –

Optional: Any comments you want to include about a change batch request.

- **Changes** (*list*) – [REQUIRED]

A complex type that contains one `Change` element for each resource record set that you want to create or delete.

* (*dict*) –

A complex type that contains the information for each change in a change batch request.

- **Action** (*string*) – [REQUIRED]

The action to perform.

Valid values: `CREATE` | `DELETE` | `UPSERT`

- **ResourceRecordSet** (*dict*) – [REQUIRED]

Information about the resource record set to create or delete.

- **Name** (*string*) – [REQUIRED]

The domain name of the current resource record set.

- **Type** (*string*) – [REQUIRED]

The type of the current resource record set.

- **SetIdentifier** (*string*) –

Weighted, Latency, Geo, and Failover resource record sets only: An identifier that differentiates among multiple resource record sets that have the same combination of DNS name and type.

- **Weight** (*integer*) –

Weighted resource record sets only: Among resource record sets that have the same combination of DNS name and type, a value that determines what portion of traffic for the current resource record set is routed to the associated location.

- **Region** (*string*) –

Latency-based resource record sets only: Among resource record sets that have the same combination of DNS name and type, a value that specifies the AWS region for the current resource record set.

- **GeoLocation** (*dict*) –

Geo location resource record sets only: Among resource record sets that have the same combination of DNS name and type, a value that specifies the geo location for the current resource record set.

- **ContinentCode** (*string*) –

The code for a continent geo location. Note: only continent locations have a continent code.

Valid values: AF | AN | AS | EU | OC | NA | SA

Constraint: Specifying `ContinentCode` with either `CountryCode` or `SubdivisionCode` returns an `InvalidInput` error.

- **CountryCode** (*string*) –

The code for a country geo location. The default location uses ‘*’ for the country code and will match all locations that are not matched by a geo location.

The default geo location uses a * for the country code. All other country codes follow the ISO 3166 two-character code.

- **SubdivisionCode** (*string*) –

The code for a country’s subdivision (e.g., a province of Canada). A subdivision code is only valid with the appropriate country code.

Constraint: Specifying `SubdivisionCode` without `CountryCode` returns an `InvalidInput` error.

- **Failover** (*string*) –

Failover resource record sets only: Among resource record sets that have the same combination of DNS name and type, a value that indicates whether the current resource record set is a primary or secondary resource record set. A failover set may contain at most one resource record set marked as primary and one resource record set marked as secondary. A resource record set marked as primary will be returned if any of the following are true: (1) an associated health check is passing, (2) if the resource record set is an alias with the evaluate target health and at least one target resource record set is healthy, (3) both the primary and secondary resource record set are failing health checks or (4) there is no secondary resource record set. A secondary resource record set will be returned if: (1) the primary is failing a health check and either the secondary is passing a health check or has no associated health check, or (2) there is no primary resource record set.

Valid values: PRIMARY | SECONDARY

- **TTL** (*integer*) –

The cache time to live for the current resource record set.

- **ResourceRecords** (*list*) –

A complex type that contains the resource records for the current resource record set.

- (*dict*) –

A complex type that contains the value of the `Value` element for the current resource record set.

- **Value** (*string*) – [REQUIRED]

The value of the `Value` element for the current resource record set.

- **AliasTarget** (*dict*) –

Alias resource record sets only: Information about the AWS resource to which you are redirecting traffic.

- **HostedZoneId** (*string*) – **[REQUIRED]**

Alias resource record sets only: The value of the hosted zone ID for the AWS resource.

For more information and an example, see [Creating Alias Resource Record Sets](#) in the *Amazon Route 53 Developer Guide*

- **DNSName** (*string*) – **[REQUIRED]**

Alias resource record sets only: The external DNS name associated with the AWS Resource.

For more information and an example, see [Creating Alias Resource Record Sets](#) in the *Amazon Route 53 Developer Guide*

- **EvaluateTargetHealth** (*boolean*) – **[REQUIRED]**

Alias resource record sets only: A boolean value that indicates whether this Resource Record Set should respect the health status of any health checks associated with the ALIAS target record which it is linked to.

For more information and an example, see [Creating Alias Resource Record Sets](#) in the *Amazon Route 53 Developer Guide*

- **HealthCheckId** (*string*) –

Health Check resource record sets only, not required for alias resource record sets: An identifier that is used to identify health check associated with the resource record set.

Return type dict

Returns

Response Syntax

```
{
  'ChangeInfo': {
    'Id': 'string',
    'Status': 'PENDING'|'INSYNC',
    'SubmittedAt': datetime(2015, 1, 1),
    'Comment': 'string'
  }
}
```

Response Structure

- (*dict*) –

A complex type containing the response for the request.

- **ChangeInfo** (*dict*) –

A complex type that contains information about changes made to your hosted zone.

This element contains an ID that you use when performing a `GetChange` action to get detailed information about the change.

* **Id** (*string*) –

The ID of the request. Use this ID to track when the change has completed across all Amazon Route 53 DNS servers.

* **Status** (*string*) –

The current state of the request. `PENDING` indicates that this request has not yet been applied to all Amazon Route 53 DNS servers.

Valid Values: `PENDING` | `INSYNC`

* **SubmittedAt** (*datetime*) –

The date and time the change was submitted, in the format `YYYY-MM-DDThh:mm:ssZ`, as specified in the ISO 8601 standard (for example, `2009-11-19T19:37:58Z`). The `Z` after the time indicates that the time is listed in Coordinated Universal Time (UTC), which is synonymous with Greenwich Mean Time in this context.

* **Comment** (*string*) –

A complex type that describes change information about changes made to your hosted zone.

This element contains an ID that you use when performing a `GetChange` action to get detailed information about the change.

change_tags_for_resource (***kwargs*)

Request Syntax

```
response = client.change_tags_for_resource(
    ResourceType='healthcheck'|'hostedzone',
    ResourceId='string',
    AddTags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ],
    RemoveTagKeys=[
        'string',
    ]
)
```

Parameters

- **ResourceType** (*string*) – [REQUIRED]

The type of the resource.

- The resource type for health checks is `healthcheck`.
- The resource type for hosted zones is `hostedzone`.

- **ResourceId** (*string*) – [REQUIRED]

The ID of the resource for which you want to add, change, or delete tags.

- **AddTags** (*list*) – A complex type that contains a list of `Tag` elements. Each `Tag` element identifies a tag that you want to add or update for the specified resource.
 - (*dict*) –

A single tag containing a key and value.

* **Key** (*string*) –

The key for a `Tag`.

* **Value** (*string*) –

The value for a Tag .

- **RemoveTagKeys** (*list*) – A list of Tag keys that you want to remove from the specified resource.
- (*string*) –

Return type dict

Returns

Response Syntax

```
{}
```

Response Structure

- (*dict*) –

Empty response for the request.

create_health_check (***kwargs*)

This action creates a new health check.

To create a new health check, send a POST request to the 2013-04-01/healthcheck resource. The request body must include an XML document with a CreateHealthCheckRequest element. The response returns the CreateHealthCheckResponse element that contains metadata about the health check.

Request Syntax

```
response = client.create_health_check(
    CallerReference='string',
    HealthCheckConfig={
        'IPAddress': 'string',
        'Port': 123,
        'Type': 'HTTP'|'HTTPS'|'HTTP_STR_MATCH'|'HTTPS_STR_MATCH'|'TCP'|'CALCULATED',
        'ResourcePath': 'string',
        'FullyQualifiedDomainName': 'string',
        'SearchString': 'string',
        'RequestInterval': 123,
        'FailureThreshold': 123,
        'MeasureLatency': True|False,
        'Inverted': True|False,
        'HealthThreshold': 123,
        'ChildHealthChecks': [
            'string',
        ]
    }
)
```

Parameters

- **CallerReference** (*string*) – **[REQUIRED]**

A unique string that identifies the request and that allows failed CreateHealthCheck requests to be retried without the risk of executing the operation twice. You must use a unique CallerReference string every time you create a health check. CallerReference can be any unique string; you might choose to use a string that identifies your project.

Valid characters are any Unicode code points that are legal in an XML 1.0 document. The UTF-8 encoding of the value must be less than 128 bytes.

- **HealthCheckConfig** (*dict*) – [REQUIRED]

A complex type that contains health check configuration.

- **IPAddress** (*string*) –

IP Address of the instance being checked.

- **Port** (*integer*) –

Port on which connection will be opened to the instance to health check. For HTTP and HTTP_STR_MATCH this defaults to 80 if the port is not specified. For HTTPS and HTTPS_STR_MATCH this defaults to 443 if the port is not specified.

- **Type** (*string*) – [REQUIRED]

The type of health check to be performed. Currently supported types are TCP, HTTP, HTTPS, HTTP_STR_MATCH, and HTTPS_STR_MATCH.

- **ResourcePath** (*string*) –

Path to ping on the instance to check the health. Required for HTTP, HTTPS, HTTP_STR_MATCH, and HTTPS_STR_MATCH health checks, HTTP request is issued to the instance on the given port and path.

- **FullyQualifiedDomainName** (*string*) –

Fully qualified domain name of the instance to be health checked.

- **SearchString** (*string*) –

A string to search for in the body of a health check response. Required for HTTP_STR_MATCH and HTTPS_STR_MATCH health checks.

- **RequestInterval** (*integer*) –

The number of seconds between the time that Route 53 gets a response from your endpoint and the time that it sends the next health-check request.

Each Route 53 health checker makes requests at this interval. Valid values are 10 and 30. The default value is 30.

- **FailureThreshold** (*integer*) –

The number of consecutive health checks that an endpoint must pass or fail for Route 53 to change the current status of the endpoint from unhealthy to healthy or vice versa.

Valid values are integers between 1 and 10. For more information, see “How Amazon Route 53 Determines Whether an Endpoint Is Healthy” in the Amazon Route 53 Developer Guide.

- **MeasureLatency** (*boolean*) –

A Boolean value that indicates whether you want Route 53 to measure the latency between health checkers in multiple AWS regions and your endpoint and to display CloudWatch latency graphs in the Route 53 console.

- **Inverted** (*boolean*) –

A boolean value that indicates whether the status of health check should be inverted. For example, if a health check is healthy but `Inverted` is `True`, then Route 53 considers the health check to be unhealthy.

- **HealthThreshold** (*integer*) –

The minimum number of child health checks that must be healthy for Route 53 to consider the parent health check to be healthy. Valid values are integers between 0 and 256, inclusive.

- **ChildHealthChecks** (*list*) –

For a specified parent health check, a list of `HealthCheckId` values for the associated child health checks.

* (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'HealthCheck': {
    'Id': 'string',
    'CallerReference': 'string',
    'HealthCheckConfig': {
      'IPAddress': 'string',
      'Port': 123,
      'Type': 'HTTP'|'HTTPS'|'HTTP_STR_MATCH'|'HTTPS_STR_MATCH'|'TCP'|'CALCU
      'ResourcePath': 'string',
      'FullyQualifiedDomainName': 'string',
      'SearchString': 'string',
      'RequestInterval': 123,
      'FailureThreshold': 123,
      'MeasureLatency': True|False,
      'Inverted': True|False,
      'HealthThreshold': 123,
      'ChildHealthChecks': [
        'string',
      ]
    },
    'HealthCheckVersion': 123
  },
  'Location': 'string'
}
```

Response Structure

- (*dict*) –

A complex type containing the response information for the new health check.

- **HealthCheck** (*dict*) –

A complex type that contains identifying information about the health check.

- * **Id** (*string*) –

The ID of the specified health check.

- * **CallerReference** (*string*) –

A unique string that identifies the request to create the health check.

- * **HealthCheckConfig** (*dict*) –

A complex type that contains the health check configuration.

- **IPAddress** (*string*) –

IP Address of the instance being checked.

- **Port** (*integer*) –

Port on which connection will be opened to the instance to health check. For HTTP and HTTP_STR_MATCH this defaults to 80 if the port is not specified. For HTTPS and HTTPS_STR_MATCH this defaults to 443 if the port is not specified.

- **Type** (*string*) –

The type of health check to be performed. Currently supported types are TCP, HTTP, HTTPS, HTTP_STR_MATCH, and HTTPS_STR_MATCH.

- **ResourcePath** (*string*) –

Path to ping on the instance to check the health. Required for HTTP, HTTPS, HTTP_STR_MATCH, and HTTPS_STR_MATCH health checks, HTTP request is issued to the instance on the given port and path.

- **FullyQualifiedDomainName** (*string*) –

Fully qualified domain name of the instance to be health checked.

- **SearchString** (*string*) –

A string to search for in the body of a health check response. Required for HTTP_STR_MATCH and HTTPS_STR_MATCH health checks.

- **RequestInterval** (*integer*) –

The number of seconds between the time that Route 53 gets a response from your endpoint and the time that it sends the next health-check request.

Each Route 53 health checker makes requests at this interval. Valid values are 10 and 30. The default value is 30.

- **FailureThreshold** (*integer*) –

The number of consecutive health checks that an endpoint must pass or fail for Route 53 to change the current status of the endpoint from unhealthy to healthy or vice versa.

Valid values are integers between 1 and 10. For more information, see “How Amazon Route 53 Determines Whether an Endpoint Is Healthy” in the Amazon Route 53 Developer Guide.

- **MeasureLatency** (*boolean*) –

A Boolean value that indicates whether you want Route 53 to measure the latency between health checkers in multiple AWS regions and your endpoint and to display CloudWatch latency graphs in the Route 53 console.

- **Inverted** (*boolean*) –

A boolean value that indicates whether the status of health check should be inverted. For example, if a health check is healthy but `Inverted` is `True`, then Route 53 considers the health check to be unhealthy.

- **HealthThreshold** (*integer*) –

The minimum number of child health checks that must be healthy for Route 53 to consider the parent health check to be healthy. Valid values are integers between 0 and 256, inclusive.

- **ChildHealthChecks** (*list*) –

For a specified parent health check, a list of `HealthCheckId` values for the associated child health

- **VPC** (*dict*) – The VPC that you want your hosted zone to be associated with. By providing this parameter, your newly created hosted cannot be resolved anywhere other than the given VPC.
 - **VPCRegion** (*string*) –
 - **VPCId** (*string*) –

A VPC ID

- **CallerReference** (*string*) – **[REQUIRED]**

A unique string that identifies the request and that allows failed `CreateHostedZone` requests to be retried without the risk of executing the operation twice. You must use a unique `CallerReference` string every time you create a hosted zone. `CallerReference` can be any unique string; you might choose to use a string that identifies your project, such as `DNSMigration_01`.

Valid characters are any Unicode code points that are legal in an XML 1.0 document. The UTF-8 encoding of the value must be less than 128 bytes.

- **HostedZoneConfig** (*dict*) – A complex type that contains an optional comment about your hosted zone.
 - **Comment** (*string*) –

An optional comment about your hosted zone. If you don't want to specify a comment, you can omit the `HostedZoneConfig` and `Comment` elements from the XML document.

- **PrivateZone** (*boolean*) –

A value that indicates whether this is a private hosted zone. The value is returned in the response; do not specify it in the request.

- **DelegationSetId** (*string*) – The delegation set id of the reusable delegation set whose NS records you want to assign to the new hosted zone.

Return type dict

Returns

Response Syntax

```
{
  'HostedZone': {
    'Id': 'string',
    'Name': 'string',
    'CallerReference': 'string',
    'Config': {
      'Comment': 'string',
      'PrivateZone': True|False
    },
    'ResourceRecordSetCount': 123
  },
  'ChangeInfo': {
    'Id': 'string',
    'Status': 'PENDING'|'INSYNC',
    'SubmittedAt': datetime(2015, 1, 1),
    'Comment': 'string'
  },
  'DelegationSet': {
    'Id': 'string',
    'CallerReference': 'string',
    'NameServers': [
      'string',
```



```

    ]
  },
  'VPC': {
    'VPCRegion': 'us-east-1'|'us-west-1'|'us-west-2'|'eu-west-1'|'eu-central-1'
    'VPCId': 'string'
  },
  'Location': 'string'
}

```

Response Structure

- (*dict*) –

A complex type containing the response information for the new hosted zone.

- **HostedZone** (*dict*) –

A complex type that contains identifying information about the hosted zone.

- * **Id** (*string*) –

The ID of the specified hosted zone.

- * **Name** (*string*) –

The name of the domain. This must be a fully-specified domain, for example, `www.example.com`. The trailing dot is optional; Route 53 assumes that the domain name is fully qualified. This means that Route 53 treats `www.example.com` (without a trailing dot) and `www.example.com.` (with a trailing dot) as identical.

This is the name you have registered with your DNS registrar. You should ask your registrar to change the authoritative name servers for your domain to the set of `NameServers` elements returned in `DelegationSet`.

- * **CallerReference** (*string*) –

A unique string that identifies the request to create the hosted zone.

- * **Config** (*dict*) –

A complex type that contains the `Comment` element.

- **Comment** (*string*) –

An optional comment about your hosted zone. If you don't want to specify a comment, you can omit the `HostedZoneConfig` and `Comment` elements from the XML document.

- **PrivateZone** (*boolean*) –

A value that indicates whether this is a private hosted zone. The value is returned in the response; do not specify it in the request.

- * **ResourceRecordSetCount** (*integer*) –

Total number of resource record sets in the hosted zone.

- **ChangeInfo** (*dict*) –

A complex type that contains information about the request to create a hosted zone. This includes an ID that you use when you call the `GetChange` action to get the current status of the change request.

- * **Id** (*string*) –

The ID of the request. Use this ID to track when the change has completed across all Amazon Route 53 DNS servers.

* **Status** (*string*) –

The current state of the request. `PENDING` indicates that this request has not yet been applied to all Amazon Route 53 DNS servers.

Valid Values: `PENDING` | `INSYNC`

* **SubmittedAt** (*datetime*) –

The date and time the change was submitted, in the format `YYYY-MM-DDThh:mm:ssZ`, as specified in the ISO 8601 standard (for example, `2009-11-19T19:37:58Z`). The `Z` after the time indicates that the time is listed in Coordinated Universal Time (UTC), which is synonymous with Greenwich Mean Time in this context.

* **Comment** (*string*) –

A complex type that describes change information about changes made to your hosted zone.

This element contains an ID that you use when performing a `GetChange` action to get detailed information about the change.

– **DelegationSet** (*dict*) –

A complex type that contains name server information.

* **Id** (*string*) –

* **CallerReference** (*string*) –

* **NameServers** (*list*) –

A complex type that contains the authoritative name servers for the hosted zone. Use the method provided by your domain registrar to add an NS record to your domain for each `NameServer` that is assigned to your hosted zone.

· (*string*) –

– **VPC** (*dict*) –

* **VPCRegion** (*string*) –

* **VPCId** (*string*) –

A VPC ID

– **Location** (*string*) –

The unique URL representing the new hosted zone.

create_reusable_delegation_set (***kwargs*)

This action creates a reusable delegationSet.

To create a new reusable delegationSet, send a `POST` request to the `2013-04-01/delegationset` resource. The request body must include an XML document with a `CreateReusableDelegationSetRequest` element. The response returns the `CreateReusableDelegationSetResponse` element that contains metadata about the delegationSet.

If the optional parameter `HostedZoneId` is specified, it marks the delegationSet associated with that particular hosted zone as reusable.

Request Syntax

```
response = client.create_reusable_delegation_set(
    CallerReference='string',
```

```
HostedZoneId='string'
)
```

Parameters

- **CallerReference** (*string*) – [REQUIRED]

A unique string that identifies the request and that allows failed `CreateReusableDelegationSet` requests to be retried without the risk of executing the operation twice. You must use a unique `CallerReference` string every time you create a reusable delegation set. `CallerReference` can be any unique string; you might choose to use a string that identifies your project, such as `DNSMigration_01`.

Valid characters are any Unicode code points that are legal in an XML 1.0 document. The UTF-8 encoding of the value must be less than 128 bytes.

- **HostedZoneId** (*string*) – The ID of the hosted zone whose delegation set you want to mark as reusable. It is an optional parameter.

Return type dict

Returns

Response Syntax

```
{
  'DelegationSet': {
    'Id': 'string',
    'CallerReference': 'string',
    'NameServers': [
      'string',
    ]
  },
  'Location': 'string'
}
```

Response Structure

- (*dict*) –
 - **DelegationSet** (*dict*) –

A complex type that contains name server information.

- * **Id** (*string*) –
- * **CallerReference** (*string*) –
- * **NameServers** (*list*) –

A complex type that contains the authoritative name servers for the hosted zone. Use the method provided by your domain registrar to add an NS record to your domain for each `NameServer` that is assigned to your hosted zone.

· (*string*) –

- **Location** (*string*) –

The unique URL representing the new reusable delegation set.

delete_health_check (**kwargs)

This action deletes a health check. To delete a health check, send a `DELETE` request to the `2013-04-01/healthcheck/*health check ID* resource`.

Warning: You can delete a health check only if there are no resource record sets associated with this health check. If resource record sets are associated with this health check, you must disassociate them before you can delete your health check. If you try to delete a health check that is associated with resource record sets, Route 53 will deny your request with a `HealthCheckInUse` error. For information about disassociating the records from your health check, see `ChangeResourceRecordSets`.

Request Syntax

```
response = client.delete_health_check(  
    HealthCheckId='string'  
)
```

Parameters `HealthCheckId` (*string*) – [REQUIRED]

The ID of the health check to delete.

Return type dict

Returns

Response Syntax

```
{}
```

Response Structure

- (*dict*) –

Empty response for the request.

`delete_hosted_zone` (***kwargs*)

This action deletes a hosted zone. To delete a hosted zone, send a `DELETE` request to the `2013-04-01/hostedzone/*hosted zone ID* resource`.

For more information about deleting a hosted zone, see [Deleting a Hosted Zone](#) in the *Amazon Route 53 Developer Guide*.

Warning: You can delete a hosted zone only if there are no resource record sets other than the default SOA record and NS resource record sets. If your hosted zone contains other resource record sets, you must delete them before you can delete your hosted zone. If you try to delete a hosted zone that contains other resource record sets, Route 53 will deny your request with a `HostedZoneNotEmpty` error. For information about deleting records from your hosted zone, see `ChangeResourceRecordSets`.

Request Syntax

```
response = client.delete_hosted_zone(  
    Id='string'  
)
```

Parameters `Id` (*string*) – [REQUIRED]

The ID of the hosted zone you want to delete.

Return type dict

Returns

Response Syntax

```
{
  'ChangeInfo': {
    'Id': 'string',
    'Status': 'PENDING'|'INSYNC',
    'SubmittedAt': datetime(2015, 1, 1),
    'Comment': 'string'
  }
}
```

Response Structure

- (*dict*) –

A complex type containing the response information for the request.

– **ChangeInfo** (*dict*) –

A complex type that contains the ID, the status, and the date and time of your delete request.

* **Id** (*string*) –

The ID of the request. Use this ID to track when the change has completed across all Amazon Route 53 DNS servers.

* **Status** (*string*) –

The current state of the request. `PENDING` indicates that this request has not yet been applied to all Amazon Route 53 DNS servers.

Valid Values: `PENDING|INSYNC`

* **SubmittedAt** (*datetime*) –

The date and time the change was submitted, in the format `YYYY-MM-DDThh:mm:ssZ`, as specified in the ISO 8601 standard (for example, `2009-11-19T19:37:58Z`). The `Z` after the time indicates that the time is listed in Coordinated Universal Time (UTC), which is synonymous with Greenwich Mean Time in this context.

* **Comment** (*string*) –

A complex type that describes change information about changes made to your hosted zone.

This element contains an ID that you use when performing a `GetChange` action to get detailed information about the change.

delete_reusable_delegation_set (***kwargs*)

This action deletes a reusable delegation set. To delete a reusable delegation set, send a `DELETE` request to the `2013-04-01/delegationset/*delegation set ID*` resource.

Warning: You can delete a reusable delegation set only if there are no associated hosted zones. If your reusable delegation set contains associated hosted zones, you must delete them before you can delete your reusable delegation set. If you try to delete a reusable delegation set that contains associated hosted zones, Route 53 will deny your request with a `DelegationSetInUse` error.

Request Syntax

```
response = client.delete_reusable_delegation_set(
    Id='string'
)
```

Parameters **Id** (*string*) – [REQUIRED]

The ID of the reusable delegation set you want to delete.

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –

Empty response for the request.

disassociate_vpc_from_hosted_zone (**kwargs)

This action disassociates a VPC from an hosted zone.

To disassociate a VPC to a hosted zone, send a POST request to the 2013-04-01/hostedzone/*hosted zone ID* /disassociatevpc resource. The request body must include an XML document with a DisassociateVPCFromHostedZoneRequest element. The response returns the DisassociateVPCFromHostedZoneResponse element that contains ChangeInfo for you to track the progress of the DisassociateVPCFromHostedZoneRequest you made. See GetChange operation for how to track the progress of your change.

Request Syntax

```
response = client.disassociate_vpc_from_hosted_zone(
    HostedZoneId='string',
    VPC={
        'VPCRegion': 'us-east-1'|'us-west-1'|'us-west-2'|'eu-west-1'|'eu-central-1'|'ap-sou
        'VPCId': 'string'
    },
    Comment='string'
)
```

Parameters

- **HostedZoneId** (*string*) – **[REQUIRED]**

The ID of the hosted zone you want to disassociate your VPC from.

Note that you cannot disassociate the last VPC from a hosted zone.

- **VPC** (*dict*) – **[REQUIRED]**

The VPC that you want your hosted zone to be disassociated from.

- **VPCRegion** (*string*) –
- **VPCId** (*string*) –

A VPC ID

- **Comment** (*string*) – *Optional*: Any comments you want to include about a DisassociateVPCFromHostedZoneRequest.

Return type dict

Returns

Response Syntax

```
{
    'ChangeInfo': {
        'Id': 'string',
        'Status': 'PENDING'|'INSYNC',
        'SubmittedAt': datetime(2015, 1, 1),
```

```

        'Comment': 'string'
    }
}

```

Response Structure

- (*dict*) –

A complex type containing the response information for the request.

- **ChangeInfo** (*dict*) –

A complex type that contains the ID, the status, and the date and time of your `DisassociateVPCFromHostedZoneRequest`.

- * **Id** (*string*) –

The ID of the request. Use this ID to track when the change has completed across all Amazon Route 53 DNS servers.

- * **Status** (*string*) –

The current state of the request. `PENDING` indicates that this request has not yet been applied to all Amazon Route 53 DNS servers.

Valid Values: `PENDING` | `INSYNC`

- * **SubmittedAt** (*datetime*) –

The date and time the change was submitted, in the format `YYYY-MM-DDThh:mm:ssZ`, as specified in the ISO 8601 standard (for example, `2009-11-19T19:37:58Z`). The `Z` after the time indicates that the time is listed in Coordinated Universal Time (UTC), which is synonymous with Greenwich Mean Time in this context.

- * **Comment** (*string*) –

A complex type that describes change information about changes made to your hosted zone.

This element contains an ID that you use when performing a `GetChange` action to get detailed information about the change.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_change (***kwargs*)

This action returns the current status of a change batch request. The status is one of the following values:

- `PENDING` indicates that the changes in this request have not replicated to all Route 53 DNS servers. This is the initial status of all change batch requests.
- `INSYNC` indicates that the changes have replicated to all Amazon Route 53 DNS servers.

Request Syntax

```

response = client.get_change(
    Id='string'
)

```

Parameters `Id` (*string*) – [REQUIRED]

The ID of the change batch request. The value that you specify here is the value that `ChangeResourceRecordSets` returned in the `Id` element when you submitted the request.

Return type `dict`**Returns****Response Syntax**

```
{
  'ChangeInfo': {
    'Id': 'string',
    'Status': 'PENDING'|'INSYNC',
    'SubmittedAt': datetime(2015, 1, 1),
    'Comment': 'string'
  }
}
```

Response Structure

- (*dict*) –

A complex type that contains the `ChangeInfo` element.

- **ChangeInfo** (*dict*) –

A complex type that contains information about the specified change batch, including the change batch ID, the status of the change, and the date and time of the request.

- * **Id** (*string*) –

The ID of the request. Use this ID to track when the change has completed across all Amazon Route 53 DNS servers.

- * **Status** (*string*) –

The current state of the request. `PENDING` indicates that this request has not yet been applied to all Amazon Route 53 DNS servers.

Valid Values: `PENDING|INSYNC`

- * **SubmittedAt** (*datetime*) –

The date and time the change was submitted, in the format `YYYY-MM-DDThh:mm:ssZ`, as specified in the ISO 8601 standard (for example, `2009-11-19T19:37:58Z`). The `Z` after the time indicates that the time is listed in Coordinated Universal Time (UTC), which is synonymous with Greenwich Mean Time in this context.

- * **Comment** (*string*) –

A complex type that describes change information about changes made to your hosted zone.

This element contains an ID that you use when performing a `GetChange` action to get detailed information about the change.

get_checker_ip_ranges()

To retrieve a list of the IP ranges used by Amazon Route 53 health checkers to check the health of your resources, send a `GET` request to the `2013-04-01/checkeripranges` resource. You can use these IP addresses to configure router and firewall rules to allow health checkers to check the health of your resources.

Request Syntax


```
response = client.get_checker_ip_ranges()
```

Return type dict

Returns

Response Syntax

```
{
    'CheckerIpRanges': [
        'string',
    ]
}
```

Response Structure

- (*dict*) –

A complex type that contains the `CheckerIpRanges` element.

- **CheckerIpRanges** (*list*) –

A complex type that contains sorted list of IP ranges in CIDR format for Amazon Route 53 health checkers.

- * (*string*) –

get_geo_location (***kwargs*)

To retrieve a single geo location, send a GET request to the `2013-04-01/geolocation` resource with one of these options: `continentcode` | `countrycode` | `countrycode` and `subdivisioncode`.

Request Syntax

```
response = client.get_geo_location(
    ContinentCode='string',
    CountryCode='string',
    SubdivisionCode='string'
)
```

Parameters

- **ContinentCode** (*string*) – The code for a continent geo location. Note: only continent locations have a continent code.

Valid values: `AF` | `AN` | `AS` | `EU` | `OC` | `NA` | `SA`

Constraint: Specifying `ContinentCode` with either `CountryCode` or `SubdivisionCode` returns an `InvalidInput` error.

- **CountryCode** (*string*) – The code for a country geo location. The default location uses `*` for the country code and will match all locations that are not matched by a geo location.

The default geo location uses a `*` for the country code. All other country codes follow the ISO 3166 two-character code.

- **SubdivisionCode** (*string*) – The code for a country's subdivision (e.g., a province of Canada). A subdivision code is only valid with the appropriate country code.

Constraint: Specifying `SubdivisionCode` without `CountryCode` returns an `InvalidInput` error.

Return type dict

Returns

Response Syntax

```
{
  'GeoLocationDetails': {
    'ContinentCode': 'string',
    'ContinentName': 'string',
    'CountryCode': 'string',
    'CountryName': 'string',
    'SubdivisionCode': 'string',
    'SubdivisionName': 'string'
  }
}
```

Response Structure

- (*dict*) –

A complex type containing information about the specified geo location.

- **GeoLocationDetails** (*dict*) –

A complex type that contains the information about the specified geo location.

- * **ContinentCode** (*string*) –

The code for a continent geo location. Note: only continent locations have a continent code.

- * **ContinentName** (*string*) –

The name of the continent. This element is only present if `ContinentCode` is also present.

- * **CountryCode** (*string*) –

The code for a country geo location. The default location uses '*' for the country code and will match all locations that are not matched by a geo location.

The default geo location uses a * for the country code. All other country codes follow the ISO 3166 two-character code.

- * **CountryName** (*string*) –

The name of the country. This element is only present if `CountryCode` is also present.

- * **SubdivisionCode** (*string*) –

The code for a country's subdivision (e.g., a province of Canada). A subdivision code is only valid with the appropriate country code.

- * **SubdivisionName** (*string*) –

The name of the subdivision. This element is only present if `SubdivisionCode` is also present.

get_health_check (***kwargs*)

To retrieve the health check, send a GET request to the 2013-04-01/healthcheck/*healthcheck ID* resource.

Request Syntax

```
response = client.get_health_check(
    HealthCheckId='string'
)
```

Parameters **HealthCheckId** (*string*) – [REQUIRED]

The ID of the health check to retrieve.

Return type dict

Returns

Response Syntax

```
{
  'HealthCheck': {
    'Id': 'string',
    'CallerReference': 'string',
    'HealthCheckConfig': {
      'IPAddress': 'string',
      'Port': 123,
      'Type': 'HTTP'|'HTTPS'|'HTTP_STR_MATCH'|'HTTPS_STR_MATCH'|'TCP'|'CALCU
      'ResourcePath': 'string',
      'FullyQualifiedDomainName': 'string',
      'SearchString': 'string',
      'RequestInterval': 123,
      'FailureThreshold': 123,
      'MeasureLatency': True|False,
      'Inverted': True|False,
      'HealthThreshold': 123,
      'ChildHealthChecks': [
        'string',
      ]
    },
    'HealthCheckVersion': 123
  }
}
```

Response Structure

- (*dict*) –

A complex type containing information about the specified health check.

- **HealthCheck** (*dict*) –

A complex type that contains the information about the specified health check.

- * **Id** (*string*) –

The ID of the specified health check.

- * **CallerReference** (*string*) –

A unique string that identifies the request to create the health check.

- * **HealthCheckConfig** (*dict*) –

A complex type that contains the health check configuration.

- **IPAddress** (*string*) –

IP Address of the instance being checked.

- **Port** (*integer*) –

Port on which connection will be opened to the instance to health check. For HTTP and HTTP_STR_MATCH this defaults to 80 if the port is not specified. For HTTPS and HTTPS_STR_MATCH this defaults to 443 if the port is not specified.

- **Type** (*string*) –

The type of health check to be performed. Currently supported types are TCP, HTTP, HTTPS, HTTP_STR_MATCH, and HTTPS_STR_MATCH.

- **ResourcePath** (*string*) –

Path to ping on the instance to check the health. Required for HTTP, HTTPS, HTTP_STR_MATCH, and HTTPS_STR_MATCH health checks, HTTP request is issued to the instance on the given port and path.

- **FullyQualifiedDomainName** (*string*) –

Fully qualified domain name of the instance to be health checked.

- **SearchString** (*string*) –

A string to search for in the body of a health check response. Required for HTTP_STR_MATCH and HTTPS_STR_MATCH health checks.

- **RequestInterval** (*integer*) –

The number of seconds between the time that Route 53 gets a response from your endpoint and the time that it sends the next health-check request.

Each Route 53 health checker makes requests at this interval. Valid values are 10 and 30. The default value is 30.

- **FailureThreshold** (*integer*) –

The number of consecutive health checks that an endpoint must pass or fail for Route 53 to change the current status of the endpoint from unhealthy to healthy or vice versa.

Valid values are integers between 1 and 10. For more information, see “How Amazon Route 53 Determines Whether an Endpoint Is Healthy” in the Amazon Route 53 Developer Guide.

- **MeasureLatency** (*boolean*) –

A Boolean value that indicates whether you want Route 53 to measure the latency between health checkers in multiple AWS regions and your endpoint and to display CloudWatch latency graphs in the Route 53 console.

- **Inverted** (*boolean*) –

A boolean value that indicates whether the status of health check should be inverted. For example, if a health check is healthy but `Inverted` is `True`, then Route 53 considers the health check to be unhealthy.

- **HealthThreshold** (*integer*) –

The minimum number of child health checks that must be healthy for Route 53 to consider the parent health check to be healthy. Valid values are integers between 0 and 256, inclusive.

- **ChildHealthChecks** (*list*) –

For a specified parent health check, a list of `HealthCheckId` values for the associated child health checks.

· (*string*) –
 * **HealthCheckVersion** (*integer*) –

The version of the health check. You can optionally pass this value in a call to `UpdateHealthCheck` to prevent overwriting another change to the health check.

get_health_check_count()

To retrieve a count of all your health checks, send a GET request to the `2013-04-01/healthcheckcount` resource.

Request Syntax

```
response = client.get_health_check_count()
```

Return type dict

Returns

Response Syntax

```
{
    'HealthCheckCount': 123
}
```

Response Structure

• (*dict*) –

A complex type that contains the count of health checks associated with the current AWS account.

– **HealthCheckCount** (*integer*) –

The number of health checks associated with the current AWS account.

get_health_check_last_failure_reason(kwargs)**

If you want to learn why a health check is currently failing or why it failed most recently (if at all), you can get the failure reason for the most recent failure. Send a GET request to the `2013-04-01/healthcheck/*health check ID* /lastfailurereason` resource.

Request Syntax

```
response = client.get_health_check_last_failure_reason(
    HealthCheckId='string'
)
```

Parameters **HealthCheckId** (*string*) – **[REQUIRED]**

The ID of the health check for which you want to retrieve the reason for the most recent failure.

Return type dict

Returns

Response Syntax

```
{
    'HealthCheckObservations': [
        {
            'IPAddress': 'string',
            'StatusReport': {
                'Status': 'string',
                'CheckedTime': datetime(2015, 1, 1)
            }
        }
    ]
}
```

```
        }
    },
]
}
```

Response Structure

- *(dict)* –

A complex type that contains information about the most recent failure for the specified health check.

- **HealthCheckObservations** (*list*) –

A list that contains one `HealthCheckObservation` element for each Route 53 health checker.

- * *(dict)* –

A complex type that contains the IP address of a Route 53 health checker and the reason for the health check status.

- **IPAddress** (*string*) –

The IP address of the Route 53 health checker that performed the health check.

- **StatusReport** (*dict*) –

A complex type that contains information about the health check status for the current observation.

- **Status** (*string*) –

The observed health check status.

- **CheckedTime** (*datetime*) –

The date and time the health check status was observed, in the format `YYYY-MM-DDThh:mm:ssZ`, as specified in the ISO 8601 standard (for example, `2009-11-19T19:37:58Z`). The `Z` after the time indicates that the time is listed in Coordinated Universal Time (UTC), which is synonymous with Greenwich Mean Time in this context.

`get_health_check_status` (**kwargs)

To retrieve the health check status, send a GET request to the `2013-04-01/healthcheck/*healthcheck ID* /status` resource. You can use this call to get a health check's current status.

Request Syntax

```
response = client.get_health_check_status(
    HealthCheckId='string'
)
```

Parameters **HealthCheckId** (*string*) – [REQUIRED]

The ID of the health check for which you want to retrieve the most recent status.

Return type dict

Returns

Response Syntax

```
{
    'HealthCheckObservations': [
        {
            'IPAddress': 'string',
```

```

        'StatusReport': {
            'Status': 'string',
            'CheckedTime': datetime(2015, 1, 1)
        },
    ],
}

```

Response Structure

- (*dict*) –

A complex type that contains information about the status of the specified health check.

- **HealthCheckObservations** (*list*) –

A list that contains one `HealthCheckObservation` element for each Route 53 health checker.

- * (*dict*) –

A complex type that contains the IP address of a Route 53 health checker and the reason for the health check status.

- **IPAddress** (*string*) –

The IP address of the Route 53 health checker that performed the health check.

- **StatusReport** (*dict*) –

A complex type that contains information about the health check status for the current observation.

- **Status** (*string*) –

The observed health check status.

- **CheckedTime** (*datetime*) –

The date and time the health check status was observed, in the format `YYYY-MM-DDThh:mm:ssZ`, as specified in the ISO 8601 standard (for example, `2009-11-19T19:37:58Z`). The `Z` after the time indicates that the time is listed in Coordinated Universal Time (UTC), which is synonymous with Greenwich Mean Time in this context.

`get_hosted_zone` (***kwargs*)

To retrieve the delegation set for a hosted zone, send a GET request to the `2013-04-01/hostedzone/*hosted zone ID*` resource. The delegation set is the four Route 53 name servers that were assigned to the hosted zone when you created it.

Request Syntax

```

response = client.get_hosted_zone(
    Id='string'
)

```

Parameters `Id` (*string*) – [REQUIRED]

The ID of the hosted zone for which you want to get a list of the name servers in the delegation set.

Return type dict

Returns

Response Syntax

```
{
  'HostedZone': {
    'Id': 'string',
    'Name': 'string',
    'CallerReference': 'string',
    'Config': {
      'Comment': 'string',
      'PrivateZone': True|False
    },
    'ResourceRecordSetCount': 123
  },
  'DelegationSet': {
    'Id': 'string',
    'CallerReference': 'string',
    'NameServers': [
      'string',
    ]
  },
  'VPCs': [
    {
      'VPCRegion': 'us-east-1'|'us-west-1'|'us-west-2'|'eu-west-1'|'eu-centr
      'VPCId': 'string'
    },
  ]
}
```

Response Structure

- (*dict*) –

A complex type containing information about the specified hosted zone.

- **HostedZone** (*dict*) –

A complex type that contains the information about the specified hosted zone.

- * **Id** (*string*) –

The ID of the specified hosted zone.

- * **Name** (*string*) –

The name of the domain. This must be a fully-specified domain, for example, `www.example.com`. The trailing dot is optional; Route 53 assumes that the domain name is fully qualified. This means that Route 53 treats `www.example.com` (without a trailing dot) and `www.example.com.` (with a trailing dot) as identical.

This is the name you have registered with your DNS registrar. You should ask your registrar to change the authoritative name servers for your domain to the set of `NameServers` elements returned in `DelegationSet`.

- * **CallerReference** (*string*) –

A unique string that identifies the request to create the hosted zone.

- * **Config** (*dict*) –

A complex type that contains the `Comment` element.

- **Comment** (*string*) –

An optional comment about your hosted zone. If you don't want to specify a comment, you can omit the

HostedZoneConfig and Comment elements from the XML document.

- **PrivateZone** (*boolean*) –

A value that indicates whether this is a private hosted zone. The value is returned in the response; do not specify it in the request.

- * **ResourceRecordSetCount** (*integer*) –

Total number of resource record sets in the hosted zone.

- **DelegationSet** (*dict*) –

A complex type that contains information about the name servers for the specified hosted zone.

- * **Id** (*string*) –
- * **CallerReference** (*string*) –
- * **NameServers** (*list*) –

A complex type that contains the authoritative name servers for the hosted zone. Use the method provided by your domain registrar to add an NS record to your domain for each NameServer that is assigned to your hosted zone.

- (*string*) –

- **VPCs** (*list*) –

A complex type that contains information about VPCs associated with the specified hosted zone.

- * (*dict*) –
- **VPCRegion** (*string*) –
- **VPCId** (*string*) –

A VPC ID

get_hosted_zone_count()

To retrieve a count of all your hosted zones, send a GET request to the 2013-04-01/hostedzonecount resource.

Request Syntax

```
response = client.get_hosted_zone_count()
```

Return type dict

Returns

Response Syntax

```
{
    'HostedZoneCount': 123
}
```

Response Structure

- (*dict*) –

A complex type that contains the count of hosted zones associated with the current AWS account.

- **HostedZoneCount** (*integer*) –

The number of hosted zones associated with the current AWS account.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type L{botocore.paginate.Paginator}

Returns A paginator object.

get_reusable_delegation_set (***kwargs*)

To retrieve the reusable delegation set, send a GET request to the `2013-04-01/delegationset/*delegation set ID* resource`.

Request Syntax

```
response = client.get_reusable_delegation_set(
    Id='string'
)
```

Parameters *Id* (*string*) – [REQUIRED]

The ID of the reusable delegation set for which you want to get a list of the name server.

Return type dict

Returns

Response Syntax

```
{
  'DelegationSet': {
    'Id': 'string',
    'CallerReference': 'string',
    'NameServers': [
      'string',
    ]
  }
}
```

Response Structure

- (*dict*) –

A complex type containing information about the specified reusable delegation set.

- **DelegationSet** (*dict*) –

A complex type that contains the information about the nameservers for the specified delegation set ID.

- * **Id** (*string*) –
- * **CallerReference** (*string*) –
- * **NameServers** (*list*) –

A complex type that contains the authoritative name servers for the hosted zone. Use the method provided by your domain registrar to add an NS record to your domain for each `NameServer` that is assigned to your hosted zone.

- (*string*) –

`get_waiter(waiter_name)`

`list_geo_locations(**kwargs)`

To retrieve a list of supported geo locations, send a GET request to the 2013-04-01/geolocations resource. The response to this request includes a `GeoLocationDetailsList` element with zero, one, or multiple `GeoLocationDetails` child elements. The list is sorted by country code, and then subdivision code, followed by continents at the end of the list.

By default, the list of geo locations is displayed on a single page. You can control the length of the page that is displayed by using the `MaxItems` parameter. If the list is truncated, `IsTruncated` will be set to `true` and a combination of `NextContinentCode`, `NextCountryCode`, `NextSubdivisionCode` will be populated. You can pass these as parameters to `StartContinentCode`, `StartCountryCode`, `StartSubdivisionCode` to control the geo location that the list begins with.

Request Syntax

```
response = client.list_geo_locations(
    StartContinentCode='string',
    StartCountryCode='string',
    StartSubdivisionCode='string',
    MaxItems='string'
)
```

Parameters

- **StartContinentCode** (*string*) – The first continent code in the lexicographic ordering of geo locations that you want the `ListGeoLocations` request to list. For non-continent geo locations, this should be null.

Valid values: AF | AN | AS | EU | OC | NA | SA

Constraint: Specifying `ContinentCode` with either `CountryCode` or `SubdivisionCode` returns an `InvalidInput` error.

- **StartCountryCode** (*string*) – The first country code in the lexicographic ordering of geo locations that you want the `ListGeoLocations` request to list.

The default geo location uses a `*` for the country code. All other country codes follow the ISO 3166 two-character code.

- **StartSubdivisionCode** (*string*) – The first subdivision code in the lexicographic ordering of geo locations that you want the `ListGeoLocations` request to list.

Constraint: Specifying `SubdivisionCode` without `CountryCode` returns an `InvalidInput` error.

- **MaxItems** (*string*) – The maximum number of geo locations you want in the response body.

Return type dict

Returns

Response Syntax

```
{
  'GeoLocationDetailsList': [
    {
      'ContinentCode': 'string',
      'ContinentName': 'string',
      'CountryCode': 'string',
      'CountryName': 'string',
      'SubdivisionCode': 'string',
```

```
        'SubdivisionName': 'string'
    },
],
'IsTruncated': True|False,
'NextContinentCode': 'string',
'NextCountryCode': 'string',
'NextSubdivisionCode': 'string',
'MaxItems': 'string'
}
```

Response Structure

- (*dict*) –

A complex type that contains information about the geo locations that are returned by the request and information about the response.

- **GeoLocationDetailsList** (*list*) –

A complex type that contains information about the geo locations that are returned by the request.

- * (*dict*) –

A complex type that contains information about a `GeoLocation`

- **ContinentCode** (*string*) –

The code for a continent geo location. Note: only continent locations have a continent code.

- **ContinentName** (*string*) –

The name of the continent. This element is only present if `ContinentCode` is also present.

- **CountryCode** (*string*) –

The code for a country geo location. The default location uses “*” for the country code and will match all locations that are not matched by a geo location.

The default geo location uses a * for the country code. All other country codes follow the ISO 3166 two-character code.

- **CountryName** (*string*) –

The name of the country. This element is only present if `CountryCode` is also present.

- **SubdivisionCode** (*string*) –

The code for a country’s subdivision (e.g., a province of Canada). A subdivision code is only valid with the appropriate country code.

- **SubdivisionName** (*string*) –

The name of the subdivision. This element is only present if `SubdivisionCode` is also present.

- **IsTruncated** (*boolean*) –

A flag that indicates whether there are more geo locations to be listed. If your results were truncated, you can make a follow-up request for the next page of results by using the values included in the `ListGeoLocationsResponse$NextContinentCode`, `ListGeoLocationsResponse$NextCountryCode` and `ListGeoLocationsResponse$NextSubdivisionCode` elements.

Valid Values: `true` | `false`

– **NextContinentCode** (*string*) –

If the results were truncated, the continent code of the next geo location in the list. This element is present only if `ListGeoLocationsResponse$IsTruncated` is `true` and the next geo location to list is a continent location.

– **NextCountryCode** (*string*) –

If the results were truncated, the country code of the next geo location in the list. This element is present only if `ListGeoLocationsResponse$IsTruncated` is `true` and the next geo location to list is not a continent location.

– **NextSubdivisionCode** (*string*) –

If the results were truncated, the subdivision code of the next geo location in the list. This element is present only if `ListGeoLocationsResponse$IsTruncated` is `true` and the next geo location has a subdivision.

– **MaxItems** (*string*) –

The maximum number of records you requested. The maximum value of `MaxItems` is 100.

list_health_checks (***kwargs*)

To retrieve a list of your health checks, send a GET request to the `2013-04-01/healthcheck` resource. The response to this request includes a `HealthChecks` element with zero, one, or multiple `HealthCheck` child elements. By default, the list of health checks is displayed on a single page. You can control the length of the page that is displayed by using the `MaxItems` parameter. You can use the `Marker` parameter to control the health check that the list begins with.

Note: Amazon Route 53 returns a maximum of 100 items. If you set `MaxItems` to a value greater than 100, Amazon Route 53 returns only the first 100.

Request Syntax

```
response = client.list_health_checks(
    Marker='string',
    MaxItems='string'
)
```

Parameters

- **Marker** (*string*) – If the request returned more than one page of results, submit another request and specify the value of `NextMarker` from the last response in the `marker` parameter to get the next page of results.
- **MaxItems** (*string*) – Specify the maximum number of health checks to return per page of results.

Return type dict

Returns

Response Syntax

```
{
  'HealthChecks': [
    {
      'Id': 'string',
      'CallerReference': 'string',
      'HealthCheckConfig': {
```

```
        'IPAddress': 'string',
        'Port': 123,
        'Type': 'HTTP'|'HTTPS'|'HTTP_STR_MATCH'|'HTTPS_STR_MATCH'|'TCP'|'O
        'ResourcePath': 'string',
        'FullyQualifiedDomainName': 'string',
        'SearchString': 'string',
        'RequestInterval': 123,
        'FailureThreshold': 123,
        'MeasureLatency': True|False,
        'Inverted': True|False,
        'HealthThreshold': 123,
        'ChildHealthChecks': [
            'string',
        ]
    },
    'HealthCheckVersion': 123
},
],
'Marker': 'string',
'IsTruncated': True|False,
'NextMarker': 'string',
'MaxItems': 'string'
}
```

Response Structure

- (*dict*) –

A complex type that contains the response for the request.

- **HealthChecks** (*list*) –

A complex type that contains information about the health checks associated with the current AWS account.

- * (*dict*) –

A complex type that contains identifying information about the health check.

- **Id** (*string*) –

The ID of the specified health check.

- **CallerReference** (*string*) –

A unique string that identifies the request to create the health check.

- **HealthCheckConfig** (*dict*) –

A complex type that contains the health check configuration.

- **IPAddress** (*string*) –

IP Address of the instance being checked.

- **Port** (*integer*) –

Port on which connection will be opened to the instance to health check. For HTTP and HTTP_STR_MATCH this defaults to 80 if the port is not specified. For HTTPS and HTTPS_STR_MATCH this defaults to 443 if the port is not specified.

- **Type** (*string*) –

The type of health check to be performed. Currently supported types are TCP, HTTP, HTTPS, HTTP_STR_MATCH,

and HTTPS_STR_MATCH.

- **ResourcePath** (*string*) –

Path to ping on the instance to check the health. Required for HTTP, HTTPS, HTTP_STR_MATCH, and HTTPS_STR_MATCH health checks, HTTP request is issued to the instance on the given port and path.

- **FullyQualifiedDomainName** (*string*) –

Fully qualified domain name of the instance to be health checked.

- **SearchString** (*string*) –

A string to search for in the body of a health check response. Required for HTTP_STR_MATCH and HTTPS_STR_MATCH health checks.

- **RequestInterval** (*integer*) –

The number of seconds between the time that Route 53 gets a response from your endpoint and the time that it sends the next health-check request.

Each Route 53 health checker makes requests at this interval. Valid values are 10 and 30. The default value is 30.

- **FailureThreshold** (*integer*) –

The number of consecutive health checks that an endpoint must pass or fail for Route 53 to change the current status of the endpoint from unhealthy to healthy or vice versa.

Valid values are integers between 1 and 10. For more information, see “How Amazon Route 53 Determines Whether an Endpoint Is Healthy” in the Amazon Route 53 Developer Guide.

- **MeasureLatency** (*boolean*) –

A Boolean value that indicates whether you want Route 53 to measure the latency between health checkers in multiple AWS regions and your endpoint and to display CloudWatch latency graphs in the Route 53 console.

- **Inverted** (*boolean*) –

A boolean value that indicates whether the status of health check should be inverted. For example, if a health check is healthy but `Inverted` is `True`, then Route 53 considers the health check to be unhealthy.

- **HealthThreshold** (*integer*) –

The minimum number of child health checks that must be healthy for Route 53 to consider the parent health check to be healthy. Valid values are integers between 0 and 256, inclusive.

- **ChildHealthChecks** (*list*) –

For a specified parent health check, a list of `HealthCheckId` values for the associated child health checks.

- (*string*) –

- **HealthCheckVersion** (*integer*) –

The version of the health check. You can optionally pass this value in a call to `UpdateHealthCheck` to prevent overwriting another change to the health check.

– **Marker** (*string*) –

If the request returned more than one page of results, submit another request and specify the value of `NextMarker` from the last response in the `marker` parameter to get the next page of results.

– **IsTruncated** (*boolean*) –

A flag indicating whether there are more health checks to be listed. If your results were truncated, you can make a follow-up request for the next page of results by using the `Marker` element.

Valid Values: `true` | `false`

– **NextMarker** (*string*) –

Indicates where to continue listing health checks. If `ListHealthChecksResponse$IsTruncated` is `true`, make another request to `ListHealthChecks` and include the value of the `NextMarker` element in the `Marker` element to get the next page of results.

– **MaxItems** (*string*) –

The maximum number of health checks to be included in the response body. If the number of health checks associated with this AWS account exceeds `MaxItems`, the value of `ListHealthChecksResponse$IsTruncated` in the response is `true`. Call `ListHealthChecks` again and specify the value of `ListHealthChecksResponse$NextMarker` in the `ListHostedZonesRequest$Marker` element to get the next page of results.

list_hosted_zones (***kwargs*)

To retrieve a list of your hosted zones, send a GET request to the `2013-04-01/hostedzone` resource. The response to this request includes a `HostedZones` element with zero, one, or multiple `HostedZone` child elements. By default, the list of hosted zones is displayed on a single page. You can control the length of the page that is displayed by using the `MaxItems` parameter. You can use the `Marker` parameter to control the hosted zone that the list begins with.

Note: Amazon Route 53 returns a maximum of 100 items. If you set `MaxItems` to a value greater than 100, Amazon Route 53 returns only the first 100.

Request Syntax

```
response = client.list_hosted_zones(
    Marker='string',
    MaxItems='string',
    DelegationSetId='string'
)
```

Parameters

- **Marker** (*string*) – If the request returned more than one page of results, submit another request and specify the value of `NextMarker` from the last response in the `marker` parameter to get the next page of results.
- **MaxItems** (*string*) – Specify the maximum number of hosted zones to return per page of results.
- **DelegationSetId** (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'HostedZones': [
    {
      'Id': 'string',
      'Name': 'string',
      'CallerReference': 'string',
      'Config': {
        'Comment': 'string',
        'PrivateZone': True|False
      },
      'ResourceRecordSetCount': 123
    },
  ],
  'Marker': 'string',
  'IsTruncated': True|False,
  'NextMarker': 'string',
  'MaxItems': 'string'
}
```

Response Structure

- (*dict*) –

A complex type that contains the response for the request.

- **HostedZones** (*list*) –

A complex type that contains information about the hosted zones associated with the current AWS account.

- * (*dict*) –

A complex type that contain information about the specified hosted zone.

- **Id** (*string*) –

The ID of the specified hosted zone.

- **Name** (*string*) –

The name of the domain. This must be a fully-specified domain, for example, `www.example.com`. The trailing dot is optional; Route 53 assumes that the domain name is fully qualified. This means that Route 53 treats `www.example.com` (without a trailing dot) and `www.example.com.` (with a trailing dot) as identical.

This is the name you have registered with your DNS registrar. You should ask your registrar to change the authoritative name servers for your domain to the set of `NameServers` elements returned in `DelegationSet`.

- **CallerReference** (*string*) –

A unique string that identifies the request to create the hosted zone.

- **Config** (*dict*) –

A complex type that contains the `Comment` element.

- **Comment** (*string*) –

An optional comment about your hosted zone. If you don't want to specify a comment, you can omit the

HostedZoneConfig and Comment elements from the XML document.

- **PrivateZone** (*boolean*) –

A value that indicates whether this is a private hosted zone. The value is returned in the response; do not specify it in the request.

- **ResourceRecordSetCount** (*integer*) –

Total number of resource record sets in the hosted zone.

- **Marker** (*string*) –

If the request returned more than one page of results, submit another request and specify the value of NextMarker from the last response in the marker parameter to get the next page of results.

- **IsTruncated** (*boolean*) –

A flag indicating whether there are more hosted zones to be listed. If your results were truncated, you can make a follow-up request for the next page of results by using the Marker element.

Valid Values: true|false

- **NextMarker** (*string*) –

Indicates where to continue listing hosted zones. If ListHostedZonesResponse\$IsTruncated is true, make another request to ListHostedZones and include the value of the NextMarker element in the Marker element to get the next page of results.

- **MaxItems** (*string*) –

The maximum number of hosted zones to be included in the response body. If the number of hosted zones associated with this AWS account exceeds MaxItems, the value of ListHostedZonesResponse\$IsTruncated in the response is true. Call ListHostedZones again and specify the value of ListHostedZonesResponse\$NextMarker in the ListHostedZones-Request\$Marker element to get the next page of results.

list_hosted_zones_by_name (***kwargs*)

To retrieve a list of your hosted zones in lexicographic order, send a GET request to the 2013-04-01/hostedzonesbyname resource. The response to this request includes a HostedZones element with zero or more HostedZone child elements lexicographically ordered by DNS name. By default, the list of hosted zones is displayed on a single page. You can control the length of the page that is displayed by using the MaxItems parameter. You can use the DNSName and HostedZoneId parameters to control the hosted zone that the list begins with.

Note: Amazon Route 53 returns a maximum of 100 items. If you set MaxItems to a value greater than 100, Amazon Route 53 returns only the first 100.

Request Syntax

```
response = client.list_hosted_zones_by_name(  
    DNSName='string',  
    HostedZoneId='string',  
    MaxItems='string'  
)
```

Parameters

- **DNSName** (*string*) – The first name in the lexicographic ordering of domain names that you want the `ListHostedZonesByNameRequest` request to list.

If the request returned more than one page of results, submit another request and specify the value of `NextDNSName` and `NextHostedZoneId` from the last response in the `DNSName` and `HostedZoneId` parameters to get the next page of results.

- **HostedZoneId** (*string*) – If the request returned more than one page of results, submit another request and specify the value of `NextDNSName` and `NextHostedZoneId` from the last response in the `DNSName` and `HostedZoneId` parameters to get the next page of results.
- **MaxItems** (*string*) – Specify the maximum number of hosted zones to return per page of results.

Return type dict

Returns

Response Syntax

```
{
  'HostedZones': [
    {
      'Id': 'string',
      'Name': 'string',
      'CallerReference': 'string',
      'Config': {
        'Comment': 'string',
        'PrivateZone': True|False
      },
      'ResourceRecordSetCount': 123
    },
  ],
  'DNSName': 'string',
  'HostedZoneId': 'string',
  'IsTruncated': True|False,
  'NextDNSName': 'string',
  'NextHostedZoneId': 'string',
  'MaxItems': 'string'
}
```

Response Structure

- (*dict*) –

A complex type that contains the response for the request.

- **HostedZones** (*list*) –

A complex type that contains information about the hosted zones associated with the current AWS account.

- * (*dict*) –

A complex type that contain information about the specified hosted zone.

- **Id** (*string*) –

The ID of the specified hosted zone.

- **Name** (*string*) –

The name of the domain. This must be a fully-specified domain, for example, `www.example.com`. The trailing dot is

optional; Route 53 assumes that the domain name is fully qualified. This means that Route 53 treats `www.example.com` (without a trailing dot) and `www.example.com.` (with a trailing dot) as identical.

This is the name you have registered with your DNS registrar. You should ask your registrar to change the authoritative name servers for your domain to the set of `NameServers` elements returned in `DelegationSet`.

- **CallerReference** (*string*) –

A unique string that identifies the request to create the hosted zone.

- **Config** (*dict*) –

A complex type that contains the `Comment` element.

- **Comment** (*string*) –

An optional comment about your hosted zone. If you don't want to specify a comment, you can omit the `HostedZoneConfig` and `Comment` elements from the XML document.

- **PrivateZone** (*boolean*) –

A value that indicates whether this is a private hosted zone. The value is returned in the response; do not specify it in the request.

- **ResourceRecordSetCount** (*integer*) –

Total number of resource record sets in the hosted zone.

- **DNSName** (*string*) –

The `DNSName` value sent in the request.

- **HostedZoneId** (*string*) –

The `HostedZoneId` value sent in the request.

- **IsTruncated** (*boolean*) –

A flag indicating whether there are more hosted zones to be listed. If your results were truncated, you can make a follow-up request for the next page of results by using the `NextDNSName` and `NextHostedZoneId` elements.

Valid Values: `true` | `false`

- **NextDNSName** (*string*) –

If `ListHostedZonesByNameResponse$IsTruncated` is `true`, there are more hosted zones associated with the current AWS account. To get the next page of results, make another request to `ListHostedZonesByName`. Specify the value of `ListHostedZonesByNameResponse$NextDNSName` in the `ListHostedZonesByNameRequest$DNSName` element and `ListHostedZonesByNameResponse$NextHostedZoneId` in the `ListHostedZonesByNameRequest$HostedZoneId` element.

- **NextHostedZoneId** (*string*) –

If `ListHostedZonesByNameResponse$IsTruncated` is `true`, there are more hosted zones associated with the current AWS account. To get the next page of results, make another request to

`ListHostedZonesByName` . Specify the value of `ListHostedZonesByNameResponse$NextDNSName` in the `ListHostedZonesByNameRequest$DNSName` element and `ListHostedZonesByNameResponse$NextHostedZoneId` in the `ListHostedZonesByNameRequest$HostedZoneId` element.

– **MaxItems** (*string*) –

The maximum number of hosted zones to be included in the response body. If the number of hosted zones associated with this AWS account exceeds `MaxItems` , the value of `ListHostedZonesByNameResponse$IsTruncated` in the response is `true` . Call `ListHostedZonesByName` again and specify the value of `ListHostedZonesByNameResponse$NextDNSName` and `ListHostedZonesByNameResponse$NextHostedZoneId` elements respectively to get the next page of results.

list_resource_record_sets (***kwargs*)

Imagine all the resource record sets in a zone listed out in front of you. Imagine them sorted lexicographically first by DNS name (with the labels reversed, like “com.amazon.www” for example), and secondarily, lexicographically by record type. This operation retrieves at most `MaxItems` resource record sets from this list, in order, starting at a position specified by the `Name` and `Type` arguments:

- If both `Name` and `Type` are omitted, this means start the results at the first RRSET in the Hosted-Zone.
- If `Name` is specified but `Type` is omitted, this means start the results at the first RRSET in the list whose name is greater than or equal to `Name`.
- If both `Name` and `Type` are specified, this means start the results at the first RRSET in the list whose name is greater than or equal to `Name` and whose type is greater than or equal to `Type`.
- It is an error to specify the `Type` but not the `Name`.

Use `ListResourceRecordSets` to retrieve a single known record set by specifying the record set’s name and type, and setting `MaxItems = 1`

To retrieve all the records in a `HostedZone`, first pause any processes making calls to `ChangeResourceRecordSets`. Initially call `ListResourceRecordSets` without a `Name` and `Type` to get the first page of record sets. For subsequent calls, set `Name` and `Type` to the `NextName` and `NextType` values returned by the previous response.

In the presence of concurrent `ChangeResourceRecordSets` calls, there is no consistency of results across calls to `ListResourceRecordSets`. The only way to get a consistent multi-page snapshot of all RRSETs in a zone is to stop making changes while pagination is in progress.

However, the results from `ListResourceRecordSets` are consistent within a page. If `MakeChange` calls are taking place concurrently, the result of each one will either be completely visible in your results or not at all. You will not see partial changes, or changes that do not ultimately succeed. (This follows from the fact that `MakeChange` is atomic)

The results from `ListResourceRecordSets` are strongly consistent with `ChangeResourceRecordSets`. To be precise, if a single process makes a call to `ChangeResourceRecordSets` and receives a successful response, the effects of that change will be visible in a subsequent call to `ListResourceRecordSets` by that process.

Request Syntax

```
response = client.list_resource_record_sets(
    HostedZoneId='string',
    StartRecordName='string',
    StartRecordType='SOA'|'A'|'TXT'|'NS'|'CNAME'|'MX'|'PTR'|'SRV'|'SPF'|'AAAA',
    StartRecordIdentifier='string',
    MaxItems='string'
)
```

Parameters

- **HostedZoneId** (*string*) – **[REQUIRED]**

The ID of the hosted zone that contains the resource record sets that you want to get.

- **StartRecordName** (*string*) – The first name in the lexicographic ordering of domain names that you want the `ListResourceRecordSets` request to list.
- **StartRecordType** (*string*) – The DNS type at which to begin the listing of resource record sets.

Valid values: A | AAAA | CNAME | MX | NS | PTR | SOA | SPF | SRV | TXT

Values for Weighted Resource Record Sets: A | AAAA | CNAME | TXT

Values for Regional Resource Record Sets: A | AAAA | CNAME | TXT

Values for Alias Resource Record Sets: A | AAAA

Constraint: Specifying type without specifying name returns an `InvalidInput` error.

- **StartRecordIdentifier** (*string*) – *Weighted resource record sets only:* If results were truncated for a given DNS name and type, specify the value of `ListResourceRecordSetsResponse$NextRecordIdentifier` from the previous response to get the next resource record set that has the current DNS name and type.
- **MaxItems** (*string*) – The maximum number of records you want in the response body.

Return type dict

Returns

Response Syntax

```
{
  'ResourceRecordSets': [
    {
      'Name': 'string',
      'Type': 'SOA'|'A'|'TXT'|'NS'|'CNAME'|'MX'|'PTR'|'SRV'|'SPF'|'AAAA',
      'SetIdentifier': 'string',
      'Weight': 123,
      'Region': 'us-east-1'|'us-west-1'|'us-west-2'|'eu-west-1'|'eu-central-1',
      'GeoLocation': {
        'ContinentCode': 'string',
        'CountryCode': 'string',
        'SubdivisionCode': 'string'
      },
      'Failover': 'PRIMARY'|'SECONDARY',
      'TTL': 123,
      'ResourceRecords': [
        {
          'Value': 'string'
        },
      ],
      'AliasTarget': {
        'HostedZoneId': 'string',
        'DNSName': 'string',
        'EvaluateTargetHealth': True|False
      },
      'HealthCheckId': 'string'
    },
  ],
}
```

```

],
'IsTruncated': True|False,
'NextRecordName': 'string',
'NextRecordType': 'SOA'|'A'|'TXT'|'NS'|'CNAME'|'MX'|'PTR'|'SRV'|'SPF'|'AAAA',
'NextRecordIdentifier': 'string',
'MaxItems': 'string'
}

```

Response Structure

- (*dict*) –

A complex type that contains information about the resource record sets that are returned by the request and information about the response.

- **ResourceRecordSets** (*list*) –

A complex type that contains information about the resource record sets that are returned by the request.

- * (*dict*) –

A complex type that contains information about the current resource record set.

- **Name** (*string*) –

The domain name of the current resource record set.

- **Type** (*string*) –

The type of the current resource record set.

- **SetIdentifier** (*string*) –

Weighted, Latency, Geo, and Failover resource record sets only: An identifier that differentiates among multiple resource record sets that have the same combination of DNS name and type.

- **Weight** (*integer*) –

Weighted resource record sets only: Among resource record sets that have the same combination of DNS name and type, a value that determines what portion of traffic for the current resource record set is routed to the associated location.

- **Region** (*string*) –

Latency-based resource record sets only: Among resource record sets that have the same combination of DNS name and type, a value that specifies the AWS region for the current resource record set.

- **GeoLocation** (*dict*) –

Geo location resource record sets only: Among resource record sets that have the same combination of DNS name and type, a value that specifies the geo location for the current resource record set.

- **ContinentCode** (*string*) –

The code for a continent geo location. Note: only continent locations have a continent code.

Valid values: AF | AN | AS | EU | OC | NA | SA

Constraint: Specifying `ContinentCode` with either

`CountryCode` or `SubdivisionCode` returns an `InvalidInput` error.

- **CountryCode** (*string*) –

The code for a country geo location. The default location uses ‘*’ for the country code and will match all locations that are not matched by a geo location.

The default geo location uses a * for the country code. All other country codes follow the ISO 3166 two-character code.

- **SubdivisionCode** (*string*) –

The code for a country’s subdivision (e.g., a province of Canada). A subdivision code is only valid with the appropriate country code.

Constraint: Specifying `SubdivisionCode` without `CountryCode` returns an `InvalidInput` error.

- **Failover** (*string*) –

Failover resource record sets only: Among resource record sets that have the same combination of DNS name and type, a value that indicates whether the current resource record set is a primary or secondary resource record set. A failover set may contain at most one resource record set marked as primary and one resource record set marked as secondary. A resource record set marked as primary will be returned if any of the following are true: (1) an associated health check is passing, (2) if the resource record set is an alias with the evaluate target health and at least one target resource record set is healthy, (3) both the primary and secondary resource record set are failing health checks or (4) there is no secondary resource record set. A secondary resource record set will be returned if: (1) the primary is failing a health check and either the secondary is passing a health check or has no associated health check, or (2) there is no primary resource record set.

Valid values: PRIMARY | SECONDARY

- **TTL** (*integer*) –

The cache time to live for the current resource record set.

- **ResourceRecords** (*list*) –

A complex type that contains the resource records for the current resource record set.

- (*dict*) –

A complex type that contains the value of the `Value` element for the current resource record set.

- **Value** (*string*) –

The value of the `Value` element for the current resource record set.

- **AliasTarget** (*dict*) –

Alias resource record sets only: Information about the AWS resource to which you are redirecting traffic.

- **HostedZoneId** (*string*) –

Alias resource record sets only: The value of the hosted zone ID for the AWS resource.

For more information and an example, see [Creating Alias Resource Record Sets](#) in the *Amazon Route 53 Developer Guide*

· **DNSName** (*string*) –

Alias resource record sets only: The external DNS name associated with the AWS Resource.

For more information and an example, see [Creating Alias Resource Record Sets](#) in the *Amazon Route 53 Developer Guide*

· **EvaluateTargetHealth** (*boolean*) –

Alias resource record sets only: A boolean value that indicates whether this Resource Record Set should respect the health status of any health checks associated with the ALIAS target record which it is linked to.

For more information and an example, see [Creating Alias Resource Record Sets](#) in the *Amazon Route 53 Developer Guide*

· **HealthCheckId** (*string*) –

Health Check resource record sets only, not required for alias resource record sets: An identifier that is used to identify health check associated with the resource record set.

– **IsTruncated** (*boolean*) –

A flag that indicates whether there are more resource record sets to be listed. If your results were truncated, you can make a follow-up request for the next page of results by using the `ListResourceRecordSetsResponse$NextRecordName` element.

Valid Values: `true` | `false`

– **NextRecordName** (*string*) –

If the results were truncated, the name of the next record in the list. This element is present only if `ListResourceRecordSetsResponse$IsTruncated` is true.

– **NextRecordType** (*string*) –

If the results were truncated, the type of the next record in the list. This element is present only if `ListResourceRecordSetsResponse$IsTruncated` is true.

– **NextRecordIdentifier** (*string*) –

Weighted resource record sets only: If results were truncated for a given DNS name and type, the value of `SetIdentifier` for the next resource record set that has the current DNS name and type.

– **MaxItems** (*string*) –

The maximum number of records you requested. The maximum value of `MaxItems` is 100.

list_reusable_delegation_sets (**kwargs)

To retrieve a list of your reusable delegation sets, send a GET request to the

2013-04-01/delegationset resource. The response to this request includes a `DelegationSets` element with zero, one, or multiple `DelegationSet` child elements. By default, the list of delegation sets is displayed on a single page. You can control the length of the page that is displayed by using the `MaxItems` parameter. You can use the `Marker` parameter to control the delegation set that the list begins with.

Note: Amazon Route 53 returns a maximum of 100 items. If you set `MaxItems` to a value greater than 100, Amazon Route 53 returns only the first 100.

Request Syntax

```
response = client.list_reusable_delegation_sets(
    Marker='string',
    MaxItems='string'
)
```

Parameters

- **Marker** (*string*) – If the request returned more than one page of results, submit another request and specify the value of `NextMarker` from the last response in the `marker` parameter to get the next page of results.
- **MaxItems** (*string*) – Specify the maximum number of reusable delegation sets to return per page of results.

Return type dict

Returns

Response Syntax

```
{
    'DelegationSets': [
        {
            'Id': 'string',
            'CallerReference': 'string',
            'NameServers': [
                'string',
            ]
        },
    ],
    'Marker': 'string',
    'IsTruncated': True|False,
    'NextMarker': 'string',
    'MaxItems': 'string'
}
```

Response Structure

- (*dict*) –

A complex type that contains the response for the request.

- **DelegationSets** (*list*) –

A complex type that contains information about the reusable delegation sets associated with the current AWS account.

- * (*dict*) –

A complex type that contains name server information.

- **Id** (*string*) –
- **CallerReference** (*string*) –

- **NameServers** (*list*) –

A complex type that contains the authoritative name servers for the hosted zone. Use the method provided by your domain registrar to add an NS record to your domain for each `NameServer` that is assigned to your hosted zone.

- (*string*) –

- **Marker** (*string*) –

If the request returned more than one page of results, submit another request and specify the value of `NextMarker` from the last response in the `marker` parameter to get the next page of results.

- **IsTruncated** (*boolean*) –

A flag indicating whether there are more reusable delegation sets to be listed. If your results were truncated, you can make a follow-up request for the next page of results by using the `Marker` element.

Valid Values: `true` | `false`

- **NextMarker** (*string*) –

Indicates where to continue listing reusable delegation sets. If `ListReusableDelegationSetsResponse$IsTruncated` is `true`, make another request to `ListReusableDelegationSets` and include the value of the `NextMarker` element in the `Marker` element to get the next page of results.

- **MaxItems** (*string*) –

The maximum number of reusable delegation sets to be included in the response body. If the number of reusable delegation sets associated with this AWS account exceeds `MaxItems`, the value of `ListReusableDelegationSetsResponse$IsTruncated` in the response is `true`. Call `ListReusableDelegationSets` again and specify the value of `ListReusableDelegationSetsResponse$NextMarker` in the `ListReusableDelegationSetsRequest$Marker` element to get the next page of results.

list_tags_for_resource (***kwargs*)

Request Syntax

```
response = client.list_tags_for_resource(
    ResourceType='healthcheck'|'hostedzone',
    ResourceId='string'
)
```

Parameters

- **ResourceType** (*string*) – [REQUIRED]

The type of the resource.

- The resource type for health checks is `healthcheck`.
- The resource type for hosted zones is `hostedzone`.

- **ResourceId** (*string*) – [REQUIRED]

The ID of the resource for which you want to retrieve tags.

Return type dict

Returns

Response Syntax

```
{
  'ResourceTagSet': {
    'ResourceType': 'healthcheck'|'hostedzone',
    'ResourceId': 'string',
    'Tags': [
      {
        'Key': 'string',
        'Value': 'string'
      },
    ],
  }
}
```

Response Structure

- *(dict)* –

A complex type containing tags for the specified resource.

- **ResourceTagSet** (*dict*) –

A ResourceTagSet containing tags associated with the specified resource.

- * **ResourceType** (*string*) –

The type of the resource.

- The resource type for health checks is `healthcheck`.
- The resource type for hosted zones is `hostedzone`.

- * **ResourceId** (*string*) –

The ID for the specified resource.

- * **Tags** (*list*) –

The tags associated with the specified resource.

- *(dict)* –

A single tag containing a key and value.

- **Key** (*string*) –

The key for a Tag.

- **Value** (*string*) –

The value for a Tag.

list_tags_for_resources (***kwargs*)

Request Syntax

```
response = client.list_tags_for_resources(
    ResourceType='healthcheck'|'hostedzone',
    ResourceIds=[
        'string',
    ]
)
```

Parameters

- **ResourceType** (*string*) – [REQUIRED]

The type of the resources.

- The resource type for health checks is `healthcheck`.
- The resource type for hosted zones is `hostedzone`.

- **ResourceIds** (*list*) – [REQUIRED]

A complex type that contains the ResourceId element for each resource for which you want to get a list of tags.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'ResourceTagSets': [
    {
      'ResourceType': 'healthcheck'|'hostedzone',
      'ResourceId': 'string',
      'Tags': [
        {
          'Key': 'string',
          'Value': 'string'
        },
        ...
      ]
    },
    ...
  ]
}
```

Response Structure

- (*dict*) –

A complex type containing tags for the specified resources.

- **ResourceTagSets** (*list*) –

A list of ResourceTagSet s containing tags associated with the specified resources.

- * (*dict*) –

A complex type containing a resource and its associated tags.

- **ResourceType** (*string*) –

The type of the resource.

- The resource type for health checks is healthcheck .

- The resource type for hosted zones is hostedzone .

- **ResourceId** (*string*) –

The ID for the specified resource.

- **Tags** (*list*) –

The tags associated with the specified resource.

- (*dict*) –

A single tag containing a key and value.

- **Key** (*string*) –

The key for a Tag .

- **Value** (*string*) –

The value for a Tag .

update_health_check (***kwargs*)

This action updates an existing health check.

To update a health check, send a POST request to the 2013-04-01/healthcheck/*healthcheck ID* resource. The request body must include an XML document

with an `UpdateHealthCheckRequest` element. The response returns an `UpdateHealthCheckResponse` element, which contains metadata about the health check.

Request Syntax

```
response = client.update_health_check(
    HealthCheckId='string',
    HealthCheckVersion=123,
    IPAddress='string',
    Port=123,
    ResourcePath='string',
    FullyQualifiedDomainName='string',
    SearchString='string',
    FailureThreshold=123,
    Inverted=True|False,
    HealthThreshold=123,
    ChildHealthChecks=[
        'string',
    ]
)
```

Parameters

- **HealthCheckId** (*string*) – [REQUIRED]

The ID of the health check to update.

- **HealthCheckVersion** (*integer*) – Optional. When you specify a health check version, Route 53 compares this value with the current value in the health check, which prevents you from updating the health check when the versions don't match. Using `HealthCheckVersion` lets you prevent overwriting another change to the health check.

- **IPAddress** (*string*) – The IP address of the resource that you want to check.

Specify this value only if you want to change it.

- **Port** (*integer*) – The port on which you want Route 53 to open a connection to perform health checks.

Specify this value only if you want to change it.

- **ResourcePath** (*string*) – The path that you want Amazon Route 53 to request when performing health checks. The path can be any value for which your endpoint will return an HTTP status code of 2xx or 3xx when the endpoint is healthy, for example the file `/docs/route53-health-check.html`.

Specify this value only if you want to change it.

- **FullyQualifiedDomainName** (*string*) – Fully qualified domain name of the instance to be health checked.

Specify this value only if you want to change it.

- **SearchString** (*string*) – If the value of `Type` is `HTTP_STR_MATCH` or `HTTP_STR_MATCH`, the string that you want Route 53 to search for in the response body from the specified resource. If the string appears in the response body, Route 53 considers the resource healthy.

Specify this value only if you want to change it.

- **FailureThreshold** (*integer*) – The number of consecutive health checks that an endpoint must pass or fail for Route 53 to change the current status of the endpoint from unhealthy to healthy or vice versa.

Valid values are integers between 1 and 10. For more information, see “How Amazon Route 53 Determines Whether an Endpoint Is Healthy” in the Amazon

Route 53 Developer Guide.

Specify this value only if you want to change it.

- **Inverted** (*boolean*) – A boolean value that indicates whether the status of health check should be inverted. For example, if a health check is healthy but `Inverted` is `True`, then Route 53 considers the health check to be unhealthy.

Specify this value only if you want to change it.

- **HealthThreshold** (*integer*) – The minimum number of child health checks that must be healthy for Route 53 to consider the parent health check to be healthy. Valid values are integers between 0 and 256, inclusive.

Specify this value only if you want to change it.

- **ChildHealthChecks** (*list*) – For a specified parent health check, a list of `HealthCheckId` values for the associated child health checks.

Specify this value only if you want to change it.

– (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'HealthCheck': {
    'Id': 'string',
    'CallerReference': 'string',
    'HealthCheckConfig': {
      'IPAddress': 'string',
      'Port': 123,
      'Type': 'HTTP'|'HTTPS'|'HTTP_STR_MATCH'|'HTTPS_STR_MATCH'|'TCP'|'CALCU
      'ResourcePath': 'string',
      'FullyQualifiedDomainName': 'string',
      'SearchString': 'string',
      'RequestInterval': 123,
      'FailureThreshold': 123,
      'MeasureLatency': True|False,
      'Inverted': True|False,
      'HealthThreshold': 123,
      'ChildHealthChecks': [
        'string',
      ]
    },
    'HealthCheckVersion': 123
  }
}
```

Response Structure

- (*dict*) –
 - **HealthCheck** (*dict*) –

A complex type that contains identifying information about the health check.

* **Id** (*string*) –

The ID of the specified health check.

* **CallerReference** (*string*) –

A unique string that identifies the request to create the health check.

* **HealthCheckConfig** (*dict*) –

A complex type that contains the health check configuration.

· **IPAddress** (*string*) –

IP Address of the instance being checked.

· **Port** (*integer*) –

Port on which connection will be opened to the instance to health check. For HTTP and HTTP_STR_MATCH this defaults to 80 if the port is not specified. For HTTPS and HTTPS_STR_MATCH this defaults to 443 if the port is not specified.

· **Type** (*string*) –

The type of health check to be performed. Currently supported types are TCP, HTTP, HTTPS, HTTP_STR_MATCH, and HTTPS_STR_MATCH.

· **ResourcePath** (*string*) –

Path to ping on the instance to check the health. Required for HTTP, HTTPS, HTTP_STR_MATCH, and HTTPS_STR_MATCH health checks, HTTP request is issued to the instance on the given port and path.

· **FullyQualifiedDomainName** (*string*) –

Fully qualified domain name of the instance to be health checked.

· **SearchString** (*string*) –

A string to search for in the body of a health check response. Required for HTTP_STR_MATCH and HTTPS_STR_MATCH health checks.

· **RequestInterval** (*integer*) –

The number of seconds between the time that Route 53 gets a response from your endpoint and the time that it sends the next health-check request.

Each Route 53 health checker makes requests at this interval. Valid values are 10 and 30. The default value is 30.

· **FailureThreshold** (*integer*) –

The number of consecutive health checks that an endpoint must pass or fail for Route 53 to change the current status of the endpoint from unhealthy to healthy or vice versa.

Valid values are integers between 1 and 10. For more information, see “How Amazon Route 53 Determines Whether an Endpoint Is Healthy” in the Amazon Route 53 Developer Guide.

· **MeasureLatency** (*boolean*) –

A Boolean value that indicates whether you want Route 53 to measure the latency between health checkers in multiple AWS regions and your endpoint and to display CloudWatch latency graphs in the Route 53 console.

· **Inverted** (*boolean*) –

A boolean value that indicates whether the status of health

check should be inverted. For example, if a health check is healthy but `Inverted` is `True`, then Route 53 considers the health check to be unhealthy.

- **HealthThreshold** (*integer*) –

The minimum number of child health checks that must be healthy for Route 53 to consider the parent health check to be healthy. Valid values are integers between 0 and 256, inclusive.

- **ChildHealthChecks** (*list*) –

For a specified parent health check, a list of `HealthCheckId` values for the associated child health checks.

- (*string*) –

- * **HealthCheckVersion** (*integer*) –

The version of the health check. You can optionally pass this value in a call to `UpdateHealthCheck` to prevent overwriting another change to the health check.

update_hosted_zone_comment (***kwargs*)

To update the hosted zone comment, send a POST request to the `2013-04-01/hostedzone/*hosted zone ID*` resource. The request body must include an XML document with a `UpdateHostedZoneCommentRequest` element. The response to this request includes the modified `HostedZone` element.

Note: The comment can have a maximum length of 256 characters.

Request Syntax

```
response = client.update_hosted_zone_comment(
    Id='string',
    Comment='string'
)
```

Parameters

- **Id** (*string*) – [REQUIRED]

The ID of the hosted zone you want to update.

- **Comment** (*string*) – A comment about your hosted zone.

Return type dict

Returns

Response Syntax

```
{
  'HostedZone': {
    'Id': 'string',
    'Name': 'string',
    'CallerReference': 'string',
    'Config': {
      'Comment': 'string',
      'PrivateZone': True|False
    },
    'ResourceRecordSetCount': 123
  }
}
```

Response Structure

- (*dict*) –

A complex type containing information about the specified hosted zone after the update.

- **HostedZone** (*dict*) –

A complex type that contain information about the specified hosted zone.

- * **Id** (*string*) –

The ID of the specified hosted zone.

- * **Name** (*string*) –

The name of the domain. This must be a fully-specified domain, for example, `www.example.com`. The trailing dot is optional; Route 53 assumes that the domain name is fully qualified. This means that Route 53 treats `www.example.com` (without a trailing dot) and `www.example.com.` (with a trailing dot) as identical.

This is the name you have registered with your DNS registrar. You should ask your registrar to change the authoritative name servers for your domain to the set of `NameServers` elements returned in `DelegationSet`.

- * **CallerReference** (*string*) –

A unique string that identifies the request to create the hosted zone.

- * **Config** (*dict*) –

A complex type that contains the `Comment` element.

- **Comment** (*string*) –

An optional comment about your hosted zone. If you don't want to specify a comment, you can omit the `HostedZoneConfig` and `Comment` elements from the XML document.

- **PrivateZone** (*boolean*) –

A value that indicates whether this is a private hosted zone. The value is returned in the response; do not specify it in the request.

- * **ResourceRecordSetCount** (*integer*) –

Total number of resource record sets in the hosted zone.

Paginators

The available paginators are:

- `Route53.Paginator.ListHealthChecks`
- `Route53.Paginator.ListHostedZones`
- `Route53.Paginator.ListResourceRecordSets`

class `Route53.Paginator.ListHealthChecks`

```
paginator = client.get_paginator('list_health_checks')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Route53.Client.list_health_checks()`.

Request Syntax

```
response_iterator = paginator.paginate(
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns**Response Syntax**

```
{
    'HealthChecks': [
        {
            'Id': 'string',
            'CallerReference': 'string',
            'HealthCheckConfig': {
                'IPAddress': 'string',
                'Port': 123,
                'Type': 'HTTP'|'HTTPS'|'HTTP_STR_MATCH'|'HTTPS_STR_MATCH'|'TCP'|'UDP',
                'ResourcePath': 'string',
                'FullyQualifiedDomainName': 'string',
                'SearchString': 'string',
                'RequestInterval': 123,
                'FailureThreshold': 123,
                'MeasureLatency': True|False,
                'Inverted': True|False,
                'HealthThreshold': 123,
                'ChildHealthChecks': [
                    'string',
                ]
            },
            'HealthCheckVersion': 123
        },
    ],
    'Marker': 'string',
    'IsTruncated': True|False,
```

```
'MaxItems': 'string',  
'NextToken': 'string'  
}
```

Response Structure

- (*dict*) –

A complex type that contains the response for the request.

- **HealthChecks** (*list*) –

A complex type that contains information about the health checks associated with the current AWS account.

- * (*dict*) –

A complex type that contains identifying information about the health check.

- **Id** (*string*) –

The ID of the specified health check.

- **CallerReference** (*string*) –

A unique string that identifies the request to create the health check.

- **HealthCheckConfig** (*dict*) –

A complex type that contains the health check configuration.

- **IPAddress** (*string*) –

IP Address of the instance being checked.

- **Port** (*integer*) –

Port on which connection will be opened to the instance to health check. For HTTP and HTTP_STR_MATCH this defaults to 80 if the port is not specified. For HTTPS and HTTPS_STR_MATCH this defaults to 443 if the port is not specified.

- **Type** (*string*) –

The type of health check to be performed. Currently supported types are TCP, HTTP, HTTPS, HTTP_STR_MATCH, and HTTPS_STR_MATCH.

- **ResourcePath** (*string*) –

Path to ping on the instance to check the health. Required for HTTP, HTTPS, HTTP_STR_MATCH, and HTTPS_STR_MATCH health checks, HTTP request is issued to the instance on the given port and path.

- **FullyQualifiedDomainName** (*string*) –

Fully qualified domain name of the instance to be health checked.

- **SearchString** (*string*) –

A string to search for in the body of a health check response. Required for HTTP_STR_MATCH and HTTPS_STR_MATCH health checks.

- **RequestInterval** (*integer*) –

The number of seconds between the time that Route 53 gets a response from your endpoint and the time that it sends the

next health-check request.

Each Route 53 health checker makes requests at this interval. Valid values are 10 and 30. The default value is 30.

- **FailureThreshold** (*integer*) –

The number of consecutive health checks that an endpoint must pass or fail for Route 53 to change the current status of the endpoint from unhealthy to healthy or vice versa.

Valid values are integers between 1 and 10. For more information, see “How Amazon Route 53 Determines Whether an Endpoint Is Healthy” in the Amazon Route 53 Developer Guide.

- **MeasureLatency** (*boolean*) –

A Boolean value that indicates whether you want Route 53 to measure the latency between health checkers in multiple AWS regions and your endpoint and to display CloudWatch latency graphs in the Route 53 console.

- **Inverted** (*boolean*) –

A boolean value that indicates whether the status of health check should be inverted. For example, if a health check is healthy but `Inverted` is `True`, then Route 53 considers the health check to be unhealthy.

- **HealthThreshold** (*integer*) –

The minimum number of child health checks that must be healthy for Route 53 to consider the parent health check to be healthy. Valid values are integers between 0 and 256, inclusive.

- **ChildHealthChecks** (*list*) –

For a specified parent health check, a list of `HealthCheckId` values for the associated child health checks.

- (*string*) –

- **HealthCheckVersion** (*integer*) –

The version of the health check. You can optionally pass this value in a call to `UpdateHealthCheck` to prevent overwriting another change to the health check.

- **Marker** (*string*) –

If the request returned more than one page of results, submit another request and specify the value of `NextMarker` from the last response in the `marker` parameter to get the next page of results.

- **IsTruncated** (*boolean*) –

A flag indicating whether there are more health checks to be listed. If your results were truncated, you can make a follow-up request for the next page of results by using the `Marker` element.

Valid Values: `true` | `false`

- **MaxItems** (*string*) –

The maximum number of health checks to be included in the response body. If the number of health checks associated with this AWS account exceeds `MaxItems`, the value of `ListHealthChecksResponse$IsTruncated`

in the response is `true`. Call `ListHealthChecks` again and specify the value of `ListHealthChecksResponse$NextMarker` in the `ListHostedZonesRequest$Marker` element to get the next page of results.

- **NextToken** (*string*) –

A token to resume pagination.

class `Route53.Paginator.ListHostedZones`

```
paginator = client.get_paginator('list_hosted_zones')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Route53.Client.list_hosted_zones()`.

Request Syntax

```
response_iterator = paginator.paginate(
    DelegationSetId='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **DelegationSetId** (*string*) –
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
    'HostedZones': [
        {
            'Id': 'string',
            'Name': 'string',
            'CallerReference': 'string',
            'Config': {
                'Comment': 'string',
                'PrivateZone': True|False
            },
            'ResourceRecordSetCount': 123
        }
    ]
}
```

```

    },
    ],
    'Marker': 'string',
    'IsTruncated': True|False,
    'MaxItems': 'string',
    'NextToken': 'string'
}

```

Response Structure

- (*dict*) –

A complex type that contains the response for the request.

- **HostedZones** (*list*) –

A complex type that contains information about the hosted zones associated with the current AWS account.

- * (*dict*) –

A complex type that contain information about the specified hosted zone.

- **Id** (*string*) –

The ID of the specified hosted zone.

- **Name** (*string*) –

The name of the domain. This must be a fully-specified domain, for example, `www.example.com`. The trailing dot is optional; Route 53 assumes that the domain name is fully qualified. This means that Route 53 treats `www.example.com` (without a trailing dot) and `www.example.com.` (with a trailing dot) as identical.

This is the name you have registered with your DNS registrar. You should ask your registrar to change the authoritative name servers for your domain to the set of `NameServers` elements returned in `DelegationSet`.

- **CallerReference** (*string*) –

A unique string that identifies the request to create the hosted zone.

- **Config** (*dict*) –

A complex type that contains the `Comment` element.

- **Comment** (*string*) –

An optional comment about your hosted zone. If you don't want to specify a comment, you can omit the `HostedZoneConfig` and `Comment` elements from the XML document.

- **PrivateZone** (*boolean*) –

A value that indicates whether this is a private hosted zone. The value is returned in the response; do not specify it in the request.

- **ResourceRecordSetCount** (*integer*) –

Total number of resource record sets in the hosted zone.

- **Marker** (*string*) –

If the request returned more than one page of results, submit another request and specify the value of `NextMarker` from the last response in the `marker` parameter to get the next page of results.

– **IsTruncated** (*boolean*) –

A flag indicating whether there are more hosted zones to be listed. If your results were truncated, you can make a follow-up request for the next page of results by using the `Marker` element.

Valid Values: `true` | `false`

– **MaxItems** (*string*) –

The maximum number of hosted zones to be included in the response body. If the number of hosted zones associated with this AWS account exceeds `MaxItems`, the value of `ListHostedZonesResponse$IsTruncated` in the response is `true`. Call `ListHostedZones` again and specify the value of `ListHostedZonesResponse$NextMarker` in the `ListHostedZones-Request$Marker` element to get the next page of results.

– **NextToken** (*string*) –

A token to resume pagination.

class `Route53.Paginator.ListResourceRecordSets`

```
paginator = client.get_paginator('list_resource_record_sets')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Route53.Client.list_resource_record_sets()`.

Request Syntax

```
response_iterator = paginator.paginate(
    HostedZoneId='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

• **HostedZoneId** (*string*) – [REQUIRED]

The ID of the hosted zone that contains the resource record sets that you want to get.

• **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

– **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

– **PageSize** (*integer*) –

The size of each page.

– **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'ResourceRecordSets': [
    {
      'Name': 'string',
      'Type': 'SOA'|'A'|'TXT'|'NS'|'CNAME'|'MX'|'PTR'|'SRV'|'SPF'|'AAAA',
      'SetIdentifier': 'string',
      'Weight': 123,
      'Region': 'us-east-1'|'us-west-1'|'us-west-2'|'eu-west-1'|'eu-central-1',
      'GeoLocation': {
        'ContinentCode': 'string',
        'CountryCode': 'string',
        'SubdivisionCode': 'string'
      },
      'Failover': 'PRIMARY'|'SECONDARY',
      'TTL': 123,
      'ResourceRecords': [
        {
          'Value': 'string'
        },
      ],
      'AliasTarget': {
        'HostedZoneId': 'string',
        'DNSName': 'string',
        'EvaluateTargetHealth': True|False
      },
      'HealthCheckId': 'string'
    },
  ],
  'IsTruncated': True|False,
  'MaxItems': 'string',
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

A complex type that contains information about the resource record sets that are returned by the request and information about the response.

- **ResourceRecordSets** (*list*) –

A complex type that contains information about the resource record sets that are returned by the request.

- * (*dict*) –

A complex type that contains information about the current resource record set.

- **Name** (*string*) –

The domain name of the current resource record set.

- **Type** (*string*) –

The type of the current resource record set.

- **SetIdentifier** (*string*) –

Weighted, Latency, Geo, and Failover resource record sets only: An identifier that differentiates among multiple resource record sets that have the same combination of DNS name and type.

- **Weight** (*integer*) –

Weighted resource record sets only: Among resource record sets that have the same combination of DNS name and type, a value that determines what portion of traffic for the current resource record set is routed to the associated location.

- **Region** (*string*) –

Latency-based resource record sets only: Among resource record sets that have the same combination of DNS name and type, a value that specifies the AWS region for the current resource record set.

- **GeoLocation** (*dict*) –

Geo location resource record sets only: Among resource record sets that have the same combination of DNS name and type, a value that specifies the geo location for the current resource record set.

- **ContinentCode** (*string*) –

The code for a continent geo location. Note: only continent locations have a continent code.

Valid values: AF | AN | AS | EU | OC | NA | SA

Constraint: Specifying `ContinentCode` with either `CountryCode` or `SubdivisionCode` returns an `InvalidInput` error.

- **CountryCode** (*string*) –

The code for a country geo location. The default location uses “*” for the country code and will match all locations that are not matched by a geo location.

The default geo location uses a * for the country code. All other country codes follow the ISO 3166 two-character code.

- **SubdivisionCode** (*string*) –

The code for a country’s subdivision (e.g., a province of Canada). A subdivision code is only valid with the appropriate country code.

Constraint: Specifying `SubdivisionCode` without `CountryCode` returns an `InvalidInput` error.

- **Failover** (*string*) –

Failover resource record sets only: Among resource record sets that have the same combination of DNS name and type, a value that indicates whether the current resource record set is a primary or secondary resource record set. A failover set may contain at most one resource record set marked as primary and one resource record set marked as secondary. A resource record set marked as primary will be returned if any of the following are true: (1) an associated health check is passing,

(2) if the resource record set is an alias with the evaluate target health and at least one target resource record set is healthy, (3) both the primary and secondary resource record set are failing health checks or (4) there is no secondary resource record set. A secondary resource record set will be returned if: (1) the primary is failing a health check and either the secondary is passing a health check or has no associated health check, or (2) there is no primary resource record set.

Valid values: PRIMARY | SECONDARY

- **TTL** (*integer*) –

The cache time to live for the current resource record set.

- **ResourceRecords** (*list*) –

A complex type that contains the resource records for the current resource record set.

- (*dict*) –

A complex type that contains the value of the `Value` element for the current resource record set.

- **Value** (*string*) –

The value of the `Value` element for the current resource record set.

- **AliasTarget** (*dict*) –

Alias resource record sets only: Information about the AWS resource to which you are redirecting traffic.

- **HostedZoneId** (*string*) –

Alias resource record sets only: The value of the hosted zone ID for the AWS resource.

For more information and an example, see [Creating Alias Resource Record Sets](#) in the *Amazon Route 53 Developer Guide*

- **DNSName** (*string*) –

Alias resource record sets only: The external DNS name associated with the AWS Resource.

For more information and an example, see [Creating Alias Resource Record Sets](#) in the *Amazon Route 53 Developer Guide*

- **EvaluateTargetHealth** (*boolean*) –

Alias resource record sets only: A boolean value that indicates whether this Resource Record Set should respect the health status of any health checks associated with the ALIAS target record which it is linked to.

For more information and an example, see [Creating Alias Resource Record Sets](#) in the *Amazon Route 53 Developer Guide*

- **HealthCheckId** (*string*) –

Health Check resource record sets only, not required for alias resource record sets: An identifier that is used to identify

health check associated with the resource record set.

– **IsTruncated** (*boolean*) –

A flag that indicates whether there are more resource record sets to be listed. If your results were truncated, you can make a follow-up request for the next page of results by using the `ListResourceRecordSetsResponse$NextRecordName` element.

Valid Values: `true` | `false`

– **MaxItems** (*string*) –

The maximum number of records you requested. The maximum value of `MaxItems` is 100.

– **NextToken** (*string*) –

A token to resume pagination.

Waiters

The available waiters are:

- `Route53.Waiter.ResourceRecordSetsChanged`

class `Route53.Waiter.ResourceRecordSetsChanged`

```
waiter = client.get_waiter('resource_record_sets_changed')
```

wait (***kwargs*)

Polls `Route53.Client.get_change()` every 30 seconds until a successful state is reached. An error is returned after 60 failed checks.

Request Syntax

```
waiter.wait(  
    Id='string'  
)
```

Parameters `Id` (*string*) – **[REQUIRED]**

The ID of the change batch request. The value that you specify here is the value that `ChangeResourceRecordSets` returned in the `Id` element when you submitted the request.

Returns `None`

Route53Domains

Table of Contents

- `Route53Domains`
 - `Client`
 - `Paginators`

Client

class Route53Domains.**Client**

A low-level client representing Amazon Route 53 Domains:

```
import boto3

client = boto3.client('route53domains')
```

These are the available methods:

- `can_paginate()`
- `check_domain_availability()`
- `delete_tags_for_domain()`
- `disable_domain_auto_renew()`
- `disable_domain_transfer_lock()`
- `enable_domain_auto_renew()`
- `enable_domain_transfer_lock()`
- `generate_presigned_url()`
- `get_domain_detail()`
- `get_operation_detail()`
- `get_paginator()`
- `get_waiter()`
- `list_domains()`
- `list_operations()`
- `list_tags_for_domain()`
- `register_domain()`
- `retrieve_domain_auth_code()`
- `transfer_domain()`
- `update_domain_contact()`
- `update_domain_contact_privacy()`
- `update_domain_nameservers()`
- `update_tags_for_domain()`

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

check_domain_availability (***kwargs*)

This operation checks the availability of one domain name. You can access this API without authenticating. Note that if the availability status of a domain is pending, you must submit another request to determine the availability of the domain name.

Request Syntax

```
response = client.check_domain_availability(
    DomainName='string',
    IdnLangCode='string'
)
```

Parameters

- **DomainName** (*string*) – [REQUIRED]

The name of a domain.

Type: String

Default: None

Constraints: The domain name can contain only the letters a through z, the numbers 0 through 9, and hyphen (-). Internationalized Domain Names are not supported.

Required: Yes

- **IdnLangCode** (*string*) – Reserved for future use.

Return type dict

Returns

Response Syntax

```
{
    'Availability': 'AVAILABLE' | 'AVAILABLE_RESERVED' | 'AVAILABLE_PREORDER' | 'UNAVAIL...
```

Response Structure

- (*dict*) –

The CheckDomainAvailability response includes the following elements.

- **Availability** (*string*) –

Whether the domain name is available for registering.

Note: You can only register domains designated as AVAILABLE .

Type: String

Valid values:

- * AVAILABLE – The domain name is available.
- * AVAILABLE_RESERVED – The domain name is reserved under specific conditions.
- * AVAILABLE_PREORDER – The domain name is available and can be preordered.
- * UNAVAILABLE – The domain name is not available.
- * UNAVAILABLE_PREMIUM – The domain name is not available.
- * UNAVAILABLE_RESTRICTED – The domain name is forbidden.
- * RESERVED – The domain name has been reserved for another person or organization.
- * DONT_KNOW – The TLD registry didn't reply with a definitive answer about whether the domain name is available. Amazon Route 53 can return this response for a variety of reasons, for example, the registry is performing maintenance. Try again later.

delete_tags_for_domain (***kwargs*)

This operation deletes the specified tags for a domain.

All tag operations are eventually consistent; subsequent operations may not immediately represent all issued operations.

Request Syntax

```
response = client.delete_tags_for_domain(
    DomainName='string',
```

```

TagsToDelete=[
    'string',
]
)

```

Parameters

- **DomainName** (*string*) – [REQUIRED]

The domain for which you want to delete one or more tags.

The name of a domain.

Type: String

Default: None

Constraints: The domain name can contain only the letters a through z, the numbers 0 through 9, and hyphen (-). Hyphens are allowed only when they are surrounded by letters, numbers, or other hyphens. You cannot specify a hyphen at the beginning or end of a label. To specify an Internationalized Domain Name, you must convert the name to Punycode.

Required: Yes

- **TagsToDelete** (*list*) – [REQUIRED]

A list of tag keys to delete.

Type: A list that contains the keys of the tags that you want to delete.

Default: None

Required: No

,

– (*string*) –

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –

disable_domain_auto_renew (***kwargs*)

This operation disables automatic renewal of domain registration for the specified domain.

Note: Caution! Amazon Route 53 doesn't have a manual renewal process, so if you disable automatic renewal, registration for the domain will not be renewed when the expiration date passes, and you will lose control of the domain name.

Request Syntax

```

response = client.disable_domain_auto_renew(
    DomainName='string'
)

```

Parameters **DomainName** (*string*) – [REQUIRED]

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –

disable_domain_transfer_lock (***kwargs*)

This operation removes the transfer lock on the domain (specifically the `clientTransferProhibited` status) to allow domain transfers. We recommend you refrain from performing this action unless you intend to transfer the domain to a different registrar. Successful submission returns an operation ID that you can use to track the progress and completion of the action. If the request is not completed successfully, the domain registrant will be notified by email.

Request Syntax

```
response = client.disable_domain_transfer_lock(  
    DomainName='string'  
)
```

Parameters **DomainName** (*string*) – [REQUIRED]

The name of a domain.

Type: String

Default: None

Constraints: The domain name can contain only the letters a through z, the numbers 0 through 9, and hyphen (-). Internationalized Domain Names are not supported.

Required: Yes

Return type dict

Returns

Response Syntax

```
{  
    'OperationId': 'string'  
}
```

Response Structure

- (*dict*) –

The DisableDomainTransferLock response includes the following element.

- **OperationId** (*string*) –

Identifier for tracking the progress of the request. To use this ID to query the operation status, use `GetOperationDetail`.

Type: String

Default: None

Constraints: Maximum 255 characters.

enable_domain_auto_renew (***kwargs*)

This operation configures Amazon Route 53 to automatically renew the specified domain before the domain registration expires. The cost of renewing your domain registration is billed to your AWS account.

The period during which you can renew a domain name varies by TLD. For a list of TLDs and their renewal policies, see “[Renewal, restoration, and deletion times](#)” on the website for our registrar partner, Gandi. Route 53 requires that you renew before the end of the renewal period that is listed on the Gandi website so we can complete processing before the deadline.

Request Syntax

```
response = client.enable_domain_auto_renew(
    DomainName='string'
)
```

Parameters `DomainName` (*string*) – [REQUIRED]

Return type dict

Returns

Response Syntax

```
{}
```

Response Structure

- (*dict*) –

`enable_domain_transfer_lock` (**kwargs)

This operation sets the transfer lock on the domain (specifically the `clientTransferProhibited` status) to prevent domain transfers. Successful submission returns an operation ID that you can use to track the progress and completion of the action. If the request is not completed successfully, the domain registrant will be notified by email.

Request Syntax

```
response = client.enable_domain_transfer_lock(
    DomainName='string'
)
```

Parameters `DomainName` (*string*) – [REQUIRED]

The name of a domain.

Type: String

Default: None

Constraints: The domain name can contain only the letters a through z, the numbers 0 through 9, and hyphen (-). Internationalized Domain Names are not supported.

Required: Yes

Return type dict

Returns

Response Syntax

```
{
    'OperationId': 'string'
}
```

Response Structure

- (*dict*) –

The EnableDomainTransferLock response includes the following elements.

– **OperationId** (*string*) –

Identifier for tracking the progress of the request. To use this ID to query the operation status, use `GetOperationDetail`.

Type: String

Default: None

Constraints: Maximum 255 characters.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_domain_detail (***kwargs*)

This operation returns detailed information about the domain. The domain's contact information is also returned as part of the output.

Request Syntax

```
response = client.get_domain_detail(  
    DomainName='string'  
)
```

Parameters **DomainName** (*string*) – [REQUIRED]

The name of a domain.

Type: String

Default: None

Constraints: The domain name can contain only the letters a through z, the numbers 0 through 9, and hyphen (-). Internationalized Domain Names are not supported.

Required: Yes

Return type dict

Returns

Response Syntax

```
{  
    'DomainName': 'string',  
    'Nameservers': [  
        {  
            'Name': 'string',  
            'GlueIps': [  
                'string',  
            ]  
        },  
    ],  
    'AutoRenew': True|False,  
    'AdminContact': {
```

```

        'FirstName': 'string',
        'LastName': 'string',
        'ContactType': 'PERSON' | 'COMPANY' | 'ASSOCIATION' | 'PUBLIC_BODY' | 'RESELLER',
        'OrganizationName': 'string',
        'AddressLine1': 'string',
        'AddressLine2': 'string',
        'City': 'string',
        'State': 'string',
        'CountryCode': 'AD' | 'AE' | 'AF' | 'AG' | 'AI' | 'AL' | 'AM' | 'AN' | 'AO' | 'AQ' | 'AR' | 'AS',
        'ZipCode': 'string',
        'PhoneNumber': 'string',
        'Email': 'string',
        'Fax': 'string',
        'ExtraParams': [
            {
                'Name': 'DUNS_NUMBER' | 'BRAND_NUMBER' | 'BIRTH_DEPARTMENT' | 'BIRTH_DATE',
                'Value': 'string'
            },
        ],
    },
    'RegistrantContact': {
        'FirstName': 'string',
        'LastName': 'string',
        'ContactType': 'PERSON' | 'COMPANY' | 'ASSOCIATION' | 'PUBLIC_BODY' | 'RESELLER',
        'OrganizationName': 'string',
        'AddressLine1': 'string',
        'AddressLine2': 'string',
        'City': 'string',
        'State': 'string',
        'CountryCode': 'AD' | 'AE' | 'AF' | 'AG' | 'AI' | 'AL' | 'AM' | 'AN' | 'AO' | 'AQ' | 'AR' | 'AS',
        'ZipCode': 'string',
        'PhoneNumber': 'string',
        'Email': 'string',
        'Fax': 'string',
        'ExtraParams': [
            {
                'Name': 'DUNS_NUMBER' | 'BRAND_NUMBER' | 'BIRTH_DEPARTMENT' | 'BIRTH_DATE',
                'Value': 'string'
            },
        ],
    },
    'TechContact': {
        'FirstName': 'string',
        'LastName': 'string',
        'ContactType': 'PERSON' | 'COMPANY' | 'ASSOCIATION' | 'PUBLIC_BODY' | 'RESELLER',
        'OrganizationName': 'string',
        'AddressLine1': 'string',
        'AddressLine2': 'string',
        'City': 'string',
        'State': 'string',
        'CountryCode': 'AD' | 'AE' | 'AF' | 'AG' | 'AI' | 'AL' | 'AM' | 'AN' | 'AO' | 'AQ' | 'AR' | 'AS',
        'ZipCode': 'string',
        'PhoneNumber': 'string',
        'Email': 'string',
        'Fax': 'string',
        'ExtraParams': [
            {
                'Name': 'DUNS_NUMBER' | 'BRAND_NUMBER' | 'BIRTH_DEPARTMENT' | 'BIRTH_DATE',

```

```
        'Value': 'string'
    },
]
},
'AdminPrivacy': True|False,
'RegistrantPrivacy': True|False,
'TechPrivacy': True|False,
'RegistrarName': 'string',
'WhoIsServer': 'string',
'RegistrarUrl': 'string',
'AbuseContactEmail': 'string',
'AbuseContactPhone': 'string',
'RegistryDomainId': 'string',
'CreationDate': datetime(2015, 1, 1),
'UpdatedDate': datetime(2015, 1, 1),
'ExpirationDate': datetime(2015, 1, 1),
'Reseller': 'string',
'DnsSec': 'string',
'StatusList': [
    'string',
]
}
```

Response Structure

- (*dict*) –

The GetDomainDetail response includes the following elements.

- **DomainName** (*string*) –

The name of a domain.

Type: String

- **Nameservers** (*list*) –

The name of the domain.

Type: String

- * (*dict*) –

Nameserver includes the following elements.

- **Name** (*string*) –

The fully qualified host name of the name server.

Type: String

Constraint: Maximum 255 characters

Parent: Nameservers

- **GlueIps** (*list*) –

Glue IP address of a name server entry. Glue IP addresses are required only when the name of the name server is a subdomain of the domain. For example, if your domain is example.com and the name server for the domain is ns.example.com, you need to specify the IP address for ns.example.com.

Type: List of IP addresses.

Constraints: The list can contain only one IPv4 and one IPv6 address.

Parent: Nameservers

- (string) –
- **AutoRenew** (boolean) –
Specifies whether the domain registration is set to renew automatically.
Type: Boolean
- **AdminContact** (dict) –
Provides details about the domain administrative contact.
Type: Complex
Children: FirstName, MiddleName, LastName, ContactType, OrganizationName, AddressLine1, AddressLine2, City, State, CountryCode, ZipCode, PhoneNumber, Email, Fax, ExtraParams
 - * **FirstName** (string) –
First name of contact.
Type: String
Default: None
Constraints: Maximum 255 characters.
Parents: RegistrantContact, AdminContact, TechContact
Required: Yes
 - * **LastName** (string) –
Last name of contact.
Type: String
Default: None
Constraints: Maximum 255 characters.
Parents: RegistrantContact, AdminContact, TechContact
Required: Yes
 - * **ContactType** (string) –
Indicates whether the contact is a person, company, association, or public organization. If you choose an option other than PERSON, you must enter an organization name, and you can't enable privacy protection for the contact.
Type: String
Default: None
Constraints: Maximum 255 characters.
Valid values: PERSON | COMPANY | ASSOCIATION | PUBLIC_BODY
Parents: RegistrantContact, AdminContact, TechContact
Required: Yes

* **OrganizationName** (*string*) –

Name of the organization for contact types other than PERSON .

Type: String

Default: None

Constraints: Maximum 255 characters. Contact type must not be PERSON .

Parents: RegistrantContact , AdminContact , TechContact

Required: No

* **AddressLine1** (*string*) –

First line of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

* **AddressLine2** (*string*) –

Second line of contact's address, if any.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

* **City** (*string*) –

The city of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

* **State** (*string*) –

The state or province of the contact's city.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact ,
TechContact

Required: No

* **CountryCode** (*string*) –

Code for the country of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact ,
TechContact

Required: Yes

* **ZipCode** (*string*) –

The zip or postal code of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact ,
TechContact

Required: No

* **PhoneNumber** (*string*) –

The phone number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format "+[country dialing code].[number including any area code]". For example, a US phone number might appear as "+1.1234567890".

Parents: RegistrantContact , AdminContact ,
TechContact

Required: Yes

* **Email** (*string*) –

Email address of the contact.

Type: String

Default: None

Constraints: Maximum 254 characters.

Parents: RegistrantContact , AdminContact ,
TechContact

Required: Yes

* **Fax** (*string*) –

Fax number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format “+[country dialing code].[number including any area code]”. For example, a US phone number might appear as “+1.1234567890”.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

* **ExtraParams** (*list*) –

A list of name-value pairs for parameters required by certain top-level domains.

Type: Complex

Default: None

Parents: RegistrantContact , AdminContact , TechContact

Children: Name , Value

Required: No

· (*dict*) –

ExtraParam includes the following elements.

· **Name** (*string*) –

Name of the additional parameter required by the top-level domain.

Type: String

Default: None

Valid values: DUNS_NUMBER |
BRAND_NUMBER | BIRTH_DEPARTMENT |
BIRTH_DATE_IN_YYYY_MM_DD | BIRTH_COUNTRY
| BIRTH_CITY | DOCUMENT_NUMBER |
AU_ID_NUMBER | AU_ID_TYPE | CA_LEGAL_TYPE
| ES_IDENTIFICATION | ES_IDENTIFICATION_TYPE
| ES_LEGAL_FORM | FI_BUSINESS_NUMBER |
FI_ID_NUMBER | IT_PIN | RU_PASSPORT_DATA |
SE_ID_NUMBER | SG_ID_NUMBER | VAT_NUMBER

Parent: ExtraParams

Required: Yes

· **Value** (*string*) –

Values corresponding to the additional parameter names required by some top-level domains.

Type: String

Default: None

Constraints: Maximum 2048 characters.

Parent: ExtraParams

Required: Yes

– **RegistrantContact** (*dict*) –

Provides details about the domain registrant.

Type: Complex

Children: `FirstName`, `MiddleName`, `LastName`, `ContactType`, `OrganizationName`, `AddressLine1`, `AddressLine2`, `City`, `State`, `CountryCode`, `ZipCode`, `PhoneNumber`, `Email`, `Fax`, `ExtraParams`

* **FirstName** (*string*) –

First name of contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: `RegistrantContact`, `AdminContact`, `TechContact`

Required: Yes

* **LastName** (*string*) –

Last name of contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: `RegistrantContact`, `AdminContact`, `TechContact`

Required: Yes

* **ContactType** (*string*) –

Indicates whether the contact is a person, company, association, or public organization. If you choose an option other than `PERSON`, you must enter an organization name, and you can't enable privacy protection for the contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Valid values: `PERSON` | `COMPANY` | `ASSOCIATION` | `PUBLIC_BODY`

Parents: `RegistrantContact`, `AdminContact`, `TechContact`

Required: Yes

* **OrganizationName** (*string*) –

Name of the organization for contact types other than `PERSON`.

Type: String

Default: None

Constraints: Maximum 255 characters. Contact type must not be PERSON.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

* **AddressLine1** (*string*) –

First line of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

* **AddressLine2** (*string*) –

Second line of contact's address, if any.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

* **City** (*string*) –

The city of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

* **State** (*string*) –

The state or province of the contact's city.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

* **CountryCode** (*string*) –

Code for the country of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact ,
TechContact

Required: Yes

* **ZipCode** (*string*) –

The zip or postal code of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact ,
TechContact

Required: No

* **PhoneNumber** (*string*) –

The phone number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format “+[country dialing code].[number including any area code]”. For example, a US phone number might appear as “+1.1234567890”.

Parents: RegistrantContact , AdminContact ,
TechContact

Required: Yes

* **Email** (*string*) –

Email address of the contact.

Type: String

Default: None

Constraints: Maximum 254 characters.

Parents: RegistrantContact , AdminContact ,
TechContact

Required: Yes

* **Fax** (*string*) –

Fax number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format “+[country dialing code].[number including any area code]”. For example, a US phone number might appear as “+1.1234567890”.

Parents: RegistrantContact , AdminContact ,
TechContact

Required: No

* **ExtraParams** (*list*) –

A list of name-value pairs for parameters required by certain top-level domains.

Type: Complex

Default: None

Parents: RegistrantContact , AdminContact , TechContact

Children: Name , Value

Required: No

· (*dict*) –

ExtraParam includes the following elements.

· **Name** (*string*) –

Name of the additional parameter required by the top-level domain.

Type: String

Default: None

Valid values: DUNS_NUMBER |
BRAND_NUMBER | BIRTH_DEPARTMENT |
BIRTH_DATE_IN_YYYY_MM_DD | BIRTH_COUNTRY
| BIRTH_CITY | DOCUMENT_NUMBER |
AU_ID_NUMBER | AU_ID_TYPE | CA_LEGAL_TYPE
| ES_IDENTIFICATION | ES_IDENTIFICATION_TYPE
| ES_LEGAL_FORM | FI_BUSINESS_NUMBER |
FI_ID_NUMBER | IT_PIN | RU_PASSPORT_DATA |
SE_ID_NUMBER | SG_ID_NUMBER | VAT_NUMBER

Parent: ExtraParams

Required: Yes

· **Value** (*string*) –

Values corresponding to the additional parameter names required by some top-level domains.

Type: String

Default: None

Constraints: Maximum 2048 characters.

Parent: ExtraParams

Required: Yes

– **TechContact** (*dict*) –

Provides details about the domain technical contact.

Type: Complex

Children: FirstName , MiddleName , LastName , ContactType
, OrganizationName , AddressLine1 , AddressLine2 , City
, State , CountryCode , ZipCode , PhoneNumber , Email , Fax
, ExtraParams

* **FirstName** (*string*) –

First name of contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact ,
TechContact

Required: Yes

* **LastName** (*string*) –

Last name of contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact ,
TechContact

Required: Yes

* **ContactType** (*string*) –

Indicates whether the contact is a person, company, association, or public organization. If you choose an option other than PERSON , you must enter an organization name, and you can't enable privacy protection for the contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Valid values: PERSON | COMPANY | ASSOCIATION |
PUBLIC_BODY

Parents: RegistrantContact , AdminContact ,
TechContact

Required: Yes

* **OrganizationName** (*string*) –

Name of the organization for contact types other than PERSON .

Type: String

Default: None

Constraints: Maximum 255 characters. Contact type must not be
PERSON .

Parents: RegistrantContact , AdminContact ,
TechContact

Required: No

* **AddressLine1** (*string*) –

First line of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact ,
TechContact

Required: Yes

* **AddressLine2** (*string*) –

Second line of contact's address, if any.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact ,
TechContact

Required: No

* **City** (*string*) –

The city of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact ,
TechContact

Required: Yes

* **State** (*string*) –

The state or province of the contact's city.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact ,
TechContact

Required: No

* **CountryCode** (*string*) –

Code for the country of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact ,
TechContact

Required: Yes

* **ZipCode** (*string*) –

The zip or postal code of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact ,
TechContact

Required: No

* **PhoneNumber** (*string*) –

The phone number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format "+[country dialing code].[number including any area code]". For example, a US phone number might appear as "+1.1234567890".

Parents: RegistrantContact , AdminContact ,
TechContact

Required: Yes

* **Email** (*string*) –

Email address of the contact.

Type: String

Default: None

Constraints: Maximum 254 characters.

Parents: RegistrantContact , AdminContact ,
TechContact

Required: Yes

* **Fax** (*string*) –

Fax number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format "+[country dialing code].[number including any area code]". For example, a US phone number might appear as "+1.1234567890".

Parents: RegistrantContact , AdminContact ,
TechContact

Required: No

* **ExtraParams** (*list*) –

A list of name-value pairs for parameters required by certain top-level domains.

Type: Complex

Default: None

Parents: RegistrantContact , AdminContact ,
TechContact

Children: Name , Value

Required: No

· *(dict)* –

ExtraParam includes the following elements.

· **Name** (*string*) –

Name of the additional parameter required by the top-level domain.

Type: String

Default: None

Valid values: DUNS_NUMBER |
BRAND_NUMBER | BIRTH_DEPARTMENT |
BIRTH_DATE_IN_YYYY_MM_DD | BIRTH_COUNTRY
| BIRTH_CITY | DOCUMENT_NUMBER |
AU_ID_NUMBER | AU_ID_TYPE | CA_LEGAL_TYPE
| ES_IDENTIFICATION | ES_IDENTIFICATION_TYPE
| ES_LEGAL_FORM | FI_BUSINESS_NUMBER |
FI_ID_NUMBER | IT_PIN | RU_PASSPORT_DATA |
SE_ID_NUMBER | SG_ID_NUMBER | VAT_NUMBER

Parent: ExtraParams

Required: Yes

· **Value** (*string*) –

Values corresponding to the additional parameter names required by some top-level domains.

Type: String

Default: None

Constraints: Maximum 2048 characters.

Parent: ExtraParams

Required: Yes

– **AdminPrivacy** (*boolean*) –

Specifies whether contact information for the admin contact is concealed from WHOIS queries. If the value is `true`, WHOIS (“who is”) queries will return contact information for our registrar partner, Gandi, instead of the contact information that you enter.

Type: Boolean

– **RegistrantPrivacy** (*boolean*) –

Specifies whether contact information for the registrant contact is concealed from WHOIS queries. If the value is `true`, WHOIS (“who is”) queries will return contact information for our registrar partner, Gandi, instead of the contact information that you enter.

Type: Boolean

- **TechPrivacy** (*boolean*) –

Specifies whether contact information for the tech contact is concealed from WHOIS queries. If the value is `true`, WHOIS (“who is”) queries will return contact information for our registrar partner, Gandi, instead of the contact information that you enter.

Type: Boolean

- **RegistrarName** (*string*) –

Name of the registrar of the domain as identified in the registry. Amazon Route 53 domains are registered by registrar Gandi. The value is "GANDI SAS" .

Type: String

- **WhoIsServer** (*string*) –

The fully qualified name of the WHOIS server that can answer the WHOIS query for the domain.

Type: String

- **RegistrarUrl** (*string*) –

Web address of the registrar.

Type: String

- **AbuseContactEmail** (*string*) –

Email address to contact to report incorrect contact information for a domain, to report that the domain is being used to send spam, to report that someone is cybersquatting on a domain name, or report some other type of abuse.

Type: String

- **AbuseContactPhone** (*string*) –

Phone number for reporting abuse.

Type: String

- **RegistryDomainId** (*string*) –

Reserved for future use.

- **CreationDate** (*datetime*) –

The date when the domain was created as found in the response to a WHOIS query. The date format is Unix time.

- **UpdatedDate** (*datetime*) –

The last updated date of the domain as found in the response to a WHOIS query. The date format is Unix time.

- **ExpirationDate** (*datetime*) –

The date when the registration for the domain is set to expire. The date format is Unix time.

- **Reseller** (*string*) –

Reseller of the domain. Domains registered or transferred using Amazon Route 53 domains will have "Amazon" as the reseller.

Type: String

- **DnsSec** (*string*) –

Reserved for future use.

– **StatusList** (*list*) –

An array of domain name status codes, also known as Extensible Provisioning Protocol (EPP) status codes.

ICANN, the organization that maintains a central database of domain names, has developed a set of domain name status codes that tell you the status of a variety of operations on a domain name, for example, registering a domain name, transferring a domain name to another registrar, renewing the registration for a domain name, and so on. All registrars use this same set of status codes.

For a current list of domain name status codes and an explanation of what each code means, go to the [ICANN website](#) and search for `epp status codes`. (Search on the ICANN website; web searches sometimes return an old version of the document.)

Type: Array of String

* (*string*) –

get_operation_detail (***kwargs*)

This operation returns the current status of an operation that is not completed.

Request Syntax

```
response = client.get_operation_detail(
    OperationId='string'
)
```

Parameters **OperationId** (*string*) – **[REQUIRED]**

The identifier for the operation for which you want to get the status. Amazon Route 53 returned the identifier in the response to the original request.

Type: String

Default: None

Required: Yes

Return type dict

Returns

Response Syntax

```
{
    'OperationId': 'string',
    'Status': 'SUBMITTED' | 'IN_PROGRESS' | 'ERROR' | 'SUCCESSFUL' | 'FAILED',
    'Message': 'string',
    'DomainName': 'string',
    'Type': 'REGISTER_DOMAIN' | 'DELETE_DOMAIN' | 'TRANSFER_IN_DOMAIN' | 'UPDATE_DOMAIN',
    'SubmittedDate': datetime(2015, 1, 1)
}
```

Response Structure

- (*dict*) –

The GetOperationDetail response includes the following elements.

– **OperationId** (*string*) –

The identifier for the operation.

Type: String

- **Status** (*string*) –
The current status of the requested operation in the system.
Type: String
- **Message** (*string*) –
Detailed information on the status including possible errors.
Type: String
- **DomainName** (*string*) –
The name of a domain.
Type: String
- **Type** (*string*) –
The type of operation that was requested.
Type: String
- **SubmittedDate** (*datetime*) –
The date when the request was submitted.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type L{botocore.paginate.Paginator}

Returns A paginator object.

get_waiter (*waiter_name*)

list_domains (***kwargs*)

This operation returns all the domain names registered with Amazon Route 53 for the current AWS account.

Request Syntax

```
response = client.list_domains(
    Marker='string',
    MaxItems=123
)
```

Parameters

- **Marker** (*string*) – For an initial request for a list of domains, omit this element. If the number of domains that are associated with the current AWS account is greater than the value that you specified for `MaxItems`, you can use `Marker` to return additional domains. Get the value of `NextPageMarker` from the previous response, and submit another request that includes the value of `NextPageMarker` in the `Marker` element.

Type: String

Default: None

Constraints: The marker must match the value specified in the previous request.

Required: No

- **MaxItems** (*integer*) – Number of domains to be returned.

Type: Integer

Default: 20

Constraints: A numeral between 1 and 100.

Required: No

Return type dict

Returns

Response Syntax

```
{
  'Domains': [
    {
      'DomainName': 'string',
      'AutoRenew': True|False,
      'TransferLock': True|False,
      'Expiry': datetime(2015, 1, 1)
    },
  ],
  'NextPageMarker': 'string'
}
```

Response Structure

- (*dict*) –

The ListDomains response includes the following elements.

- **Domains** (*list*) –

A summary of domains.

Type: Complex type containing a list of domain summaries.

Children: AutoRenew, DomainName, Expiry, TransferLock

* (*dict*) –

- **DomainName** (*string*) –

The name of a domain.

Type: String

- **AutoRenew** (*boolean*) –

Indicates whether the domain is automatically renewed upon expiration.

Type: Boolean

Valid values: True|False

- **TransferLock** (*boolean*) –

Indicates whether a domain is locked from unauthorized transfer to another party.

Type: Boolean

Valid values: True|False

- **Expiry** (*datetime*) –

Expiration date of the domain in Coordinated Universal Time (UTC).

Type: Long

– **NextPageMarker** (*string*) –

If there are more domains than you specified for `MaxItems` in the request, submit another request and include the value of `NextPageMarker` in the value of `Marker`.

Type: String

Parent: Operations

list_operations (***kwargs*)

This operation returns the operation IDs of operations that are not yet complete.

Request Syntax

```
response = client.list_operations(
    Marker='string',
    MaxItems=123
)
```

Parameters

- **Marker** (*string*) – For an initial request for a list of operations, omit this element. If the number of operations that are not yet complete is greater than the value that you specified for `MaxItems`, you can use `Marker` to return additional operations. Get the value of `NextPageMarker` from the previous response, and submit another request that includes the value of `NextPageMarker` in the `Marker` element.

Type: String

Default: None

Required: No

- **MaxItems** (*integer*) – Number of domains to be returned.

Type: Integer

Default: 20

Constraints: A value between 1 and 100.

Required: No

Return type dict

Returns

Response Syntax

```
{
  'Operations': [
    {
      'OperationId': 'string',
      'Status': 'SUBMITTED'|'IN_PROGRESS'|'ERROR'|'SUCCESSFUL'|'FAILED',
      'Type': 'REGISTER_DOMAIN'|'DELETE_DOMAIN'|'TRANSFER_IN_DOMAIN'|'UPDATE',
      'SubmittedDate': datetime(2015, 1, 1)
    },
  ],
  'NextPageMarker': 'string'
}
```

Response Structure

- (*dict*) –

The ListOperations response includes the following elements.

- **Operations** (*list*) –

Lists summaries of the operations.

Type: Complex type containing a list of operation summaries

Children: `OperationId`, `Status`, `SubmittedDate`, `Type`

- * (*dict*) –

OperationSummary includes the following elements.

- **OperationId** (*string*) –

Identifier returned to track the requested action.

Type: String

- **Status** (*string*) –

The current status of the requested operation in the system.

Type: String

- **Type** (*string*) –

Type of the action requested.

Type: String

Valid values: REGISTER_DOMAIN |
DELETE_DOMAIN | TRANSFER_IN_DOMAIN |
UPDATE_DOMAIN_CONTACT | UPDATE_NAMESERVER |
CHANGE_PRIVACY_PROTECTION | DOMAIN_LOCK

- **SubmittedDate** (*datetime*) –

The date when the request was submitted.

- **NextPageMarker** (*string*) –

If there are more operations than you specified for `MaxItems` in the request, submit another request and include the value of `NextPageMarker` in the value of `Marker`.

Type: String

Parent: Operations

list_tags_for_domain (***kwargs*)

This operation returns all of the tags that are associated with the specified domain.

All tag operations are eventually consistent; subsequent operations may not immediately represent all issued operations.

Request Syntax

```
response = client.list_tags_for_domain(  
    DomainName='string'  
)
```

Parameters `DomainName` (*string*) – [REQUIRED]

The domain for which you want to get a list of tags.

Return type dict

Returns

Response Syntax

```
{
  'TagList': [
    {
      'Key': 'string',
      'Value': 'string'
    },
  ]
}
```

Response Structure

- (*dict*) –

The ListTagsForDomain response includes the following elements.

- **TagList** (*list*) –

A list of the tags that are associated with the specified domain.

Type: A complex type containing a list of tags

Each tag includes the following elements.

- * **Key** The key (name) of a tag. Type: String
- * **Value** The value of a tag. Type: String
- * (*dict*) –

Each tag includes the following elements.

- **Key** (*string*) –

The key (name) of a tag.

Type: String

Default: None

Valid values: A-Z, a-z, 0-9, space, ”./=+-@”

Constraints: Each key can be 1-128 characters long.

Required: Yes

- **Value** (*string*) –

The value of a tag.

Type: String

Default: None

Valid values: A-Z, a-z, 0-9, space, ”./=+-@”

Constraints: Each value can be 0-256 characters long.

Required: Yes

register_domain (**kwargs)

This operation registers a domain. Domains are registered by the AWS registrar partner, Gandi. For some top-level domains (TLDs), this operation requires extra parameters.

When you register a domain, Amazon Route 53 does the following:

- Creates a Amazon Route 53 hosted zone that has the same name as the domain. Amazon Route 53 assigns four name servers to your hosted zone and automatically updates your domain registration with the names of these name servers.

- Enables autorenew, so your domain registration will renew automatically each year. We'll notify you in advance of the renewal date so you can choose whether to renew the registration.
- Optionally enables privacy protection, so WHOIS queries return contact information for our registrar partner, Gandi, instead of the information you entered for registrant, admin, and tech contacts.
- If registration is successful, returns an operation ID that you can use to track the progress and completion of the action. If the request is not completed successfully, the domain registrant is notified by email.
- Charges your AWS account an amount based on the top-level domain. For more information, see [Amazon Route 53 Pricing](#).

Request Syntax

```
response = client.register_domain(
    DomainName='string',
    IdnLangCode='string',
    DurationInYears=123,
    AutoRenew=True|False,
    AdminContact={
        'FirstName': 'string',
        'LastName': 'string',
        'ContactType': 'PERSON'|'COMPANY'|'ASSOCIATION'|'PUBLIC_BODY'|'RESELLER',
        'OrganizationName': 'string',
        'AddressLine1': 'string',
        'AddressLine2': 'string',
        'City': 'string',
        'State': 'string',
        'CountryCode': 'AD'|'AE'|'AF'|'AG'|'AI'|'AL'|'AM'|'AN'|'AO'|'AQ'|'AR'|'AS'|'AT'|'AU',
        'ZipCode': 'string',
        'PhoneNumber': 'string',
        'Email': 'string',
        'Fax': 'string',
        'ExtraParams': [
            {
                'Name': 'DUNS_NUMBER'|'BRAND_NUMBER'|'BIRTH_DEPARTMENT'|'BIRTH_DATE_IN_YYYY',
                'Value': 'string'
            },
        ],
    },
    RegistrantContact={
        'FirstName': 'string',
        'LastName': 'string',
        'ContactType': 'PERSON'|'COMPANY'|'ASSOCIATION'|'PUBLIC_BODY'|'RESELLER',
        'OrganizationName': 'string',
        'AddressLine1': 'string',
        'AddressLine2': 'string',
        'City': 'string',
        'State': 'string',
        'CountryCode': 'AD'|'AE'|'AF'|'AG'|'AI'|'AL'|'AM'|'AN'|'AO'|'AQ'|'AR'|'AS'|'AT'|'AU',
        'ZipCode': 'string',
        'PhoneNumber': 'string',
        'Email': 'string',
        'Fax': 'string',
        'ExtraParams': [
            {
                'Name': 'DUNS_NUMBER'|'BRAND_NUMBER'|'BIRTH_DEPARTMENT'|'BIRTH_DATE_IN_YYYY',
                'Value': 'string'
            },
        ],
    },
)
```



```

    },
    TechContact={
        'FirstName': 'string',
        'LastName': 'string',
        'ContactType': 'PERSON' | 'COMPANY' | 'ASSOCIATION' | 'PUBLIC_BODY' | 'RESELLER',
        'OrganizationName': 'string',
        'AddressLine1': 'string',
        'AddressLine2': 'string',
        'City': 'string',
        'State': 'string',
        'CountryCode': 'AD' | 'AE' | 'AF' | 'AG' | 'AI' | 'AL' | 'AM' | 'AN' | 'AO' | 'AQ' | 'AR' | 'AS' | 'AT' | 'AU',
        'ZipCode': 'string',
        'PhoneNumber': 'string',
        'Email': 'string',
        'Fax': 'string',
        'ExtraParams': [
            {
                'Name': 'DUNS_NUMBER' | 'BRAND_NUMBER' | 'BIRTH_DEPARTMENT' | 'BIRTH_DATE_IN_YYYY',
                'Value': 'string'
            }
        ]
    },
    PrivacyProtectAdminContact=True|False,
    PrivacyProtectRegistrantContact=True|False,
    PrivacyProtectTechContact=True|False
)

```

Parameters

- **DomainName** (*string*) – [REQUIRED]

The name of a domain.

Type: String

Default: None

Constraints: The domain name can contain only the letters a through z, the numbers 0 through 9, and hyphen (-). Internationalized Domain Names are not supported.

Required: Yes

- **IdnLangCode** (*string*) – Reserved for future use.
- **DurationInYears** (*integer*) – [REQUIRED]

The number of years the domain will be registered. Domains are registered for a minimum of one year. The maximum period depends on the top-level domain.

Type: Integer

Default: 1

Valid values: Integer from 1 to 10

Required: Yes

- **AutoRenew** (*boolean*) – Indicates whether the domain will be automatically renewed (`true`) or not (`false`). Autorenewal only takes effect after the account is charged.

Type: Boolean

Valid values: `true` | `false`

Default: true

Required: No

- **AdminContact** (*dict*) – [REQUIRED]

Provides detailed contact information.

Type: Complex

Children: `FirstName` , `MiddleName` , `LastName` , `ContactType` , `OrganizationName` , `AddressLine1` , `AddressLine2` , `City` , `State` , `CountryCode` , `ZipCode` , `PhoneNumber` , `Email` , `Fax` , `ExtraParams`

Required: Yes

- **FirstName** (*string*) –

First name of contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: `RegistrantContact` , `AdminContact` , `TechContact`

Required: Yes

- **LastName** (*string*) –

Last name of contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: `RegistrantContact` , `AdminContact` , `TechContact`

Required: Yes

- **ContactType** (*string*) –

Indicates whether the contact is a person, company, association, or public organization. If you choose an option other than `PERSON` , you must enter an organization name, and you can't enable privacy protection for the contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Valid values: `PERSON` | `COMPANY` | `ASSOCIATION` | `PUBLIC_BODY`

Parents: `RegistrantContact` , `AdminContact` , `TechContact`

Required: Yes

- **OrganizationName** (*string*) –

Name of the organization for contact types other than `PERSON` .

Type: String

Default: None

Constraints: Maximum 255 characters. Contact type must not be PERSON

.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **AddressLine1** (*string*) –

First line of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **AddressLine2** (*string*) –

Second line of contact's address, if any.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **City** (*string*) –

The city of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **State** (*string*) –

The state or province of the contact's city.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **CountryCode** (*string*) –

Code for the country of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **ZipCode** (*string*) –

The zip or postal code of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **PhoneNumber** (*string*) –

The phone number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format “[+country dialing code].[number including any area code]”. For example, a US phone number might appear as “+1.1234567890”.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **Email** (*string*) –

Email address of the contact.

Type: String

Default: None

Constraints: Maximum 254 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **Fax** (*string*) –

Fax number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format “[+country dialing code].[number including any area code]”. For example, a US phone number might appear as “+1.1234567890”.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **ExtraParams** (*list*) –

A list of name-value pairs for parameters required by certain top-level domains.

Type: Complex

Default: None

Parents: RegistrantContact , AdminContact , TechContact

Children: Name , Value

Required: No

* (*dict*) –

ExtraParam includes the following elements.

· **Name** (*string*) – **[REQUIRED]**

Name of the additional parameter required by the top-level domain.

Type: String

Default: None

Valid values: DUNS_NUMBER | BRAND_NUMBER | BIRTH_DEPARTMENT | BIRTH_DATE_IN_YYYY_MM_DD | BIRTH_COUNTRY | BIRTH_CITY | DOCUMENT_NUMBER | AU_ID_NUMBER | AU_ID_TYPE | CA_LEGAL_TYPE | ES_IDENTIFICATION | ES_IDENTIFICATION_TYPE | ES_LEGAL_FORM | FI_BUSINESS_NUMBER | FI_ID_NUMBER | IT_PIN | RU_PASSPORT_DATA | SE_ID_NUMBER | SG_ID_NUMBER | VAT_NUMBER

Parent: ExtraParams

Required: Yes

· **Value** (*string*) – **[REQUIRED]**

Values corresponding to the additional parameter names required by some top-level domains.

Type: String

Default: None

Constraints: Maximum 2048 characters.

Parent: ExtraParams

Required: Yes

• **RegistrantContact** (*dict*) – **[REQUIRED]**

Provides detailed contact information.

Type: Complex

Children: FirstName , MiddleName , LastName , ContactType , OrganizationName , AddressLine1 , AddressLine2 , City , State , CountryCode , ZipCode , PhoneNumber , Email , Fax , ExtraParams

Required: Yes

– **FirstName** (*string*) –

First name of contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **LastName** (*string*) –

Last name of contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **ContactType** (*string*) –

Indicates whether the contact is a person, company, association, or public organization. If you choose an option other than PERSON , you must enter an organization name, and you can't enable privacy protection for the contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Valid values: PERSON | COMPANY | ASSOCIATION | PUBLIC_BODY

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **OrganizationName** (*string*) –

Name of the organization for contact types other than PERSON .

Type: String

Default: None

Constraints: Maximum 255 characters. Contact type must not be PERSON .

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **AddressLine1** (*string*) –

First line of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **AddressLine2** (*string*) –

Second line of contact's address, if any.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **City** (*string*) –

The city of the contact’s address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **State** (*string*) –

The state or province of the contact’s city.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **CountryCode** (*string*) –

Code for the country of the contact’s address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **ZipCode** (*string*) –

The zip or postal code of the contact’s address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **PhoneNumber** (*string*) –

The phone number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format “+[country dialing code].[number including any area code]”. For example, a US phone number might appear as “+1.1234567890”.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **Email** (*string*) –

Email address of the contact.

Type: String

Default: None

Constraints: Maximum 254 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **Fax** (*string*) –

Fax number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format “[+country dialing code].[number including any area code]”. For example, a US phone number might appear as “+1.1234567890” .

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **ExtraParams** (*list*) –

A list of name-value pairs for parameters required by certain top-level domains.

Type: Complex

Default: None

Parents: RegistrantContact , AdminContact , TechContact

Children: Name , Value

Required: No

* (*dict*) –

ExtraParam includes the following elements.

· **Name** (*string*) – **[REQUIRED]**

Name of the additional parameter required by the top-level domain.

Type: String

Default: None

Valid values: DUNS_NUMBER |
BRAND_NUMBER | BIRTH_DEPARTMENT |
BIRTH_DATE_IN_YYYY_MM_DD | BIRTH_COUNTRY |
BIRTH_CITY | DOCUMENT_NUMBER |
AU_ID_NUMBER | AU_ID_TYPE | CA_LEGAL_TYPE |
ES_IDENTIFICATION | ES_IDENTIFICATION_TYPE |
ES_LEGAL_FORM | FI_BUSINESS_NUMBER |
FI_ID_NUMBER | IT_PIN | RU_PASSPORT_DATA |
SE_ID_NUMBER | SG_ID_NUMBER | VAT_NUMBER

Parent: ExtraParams

Required: Yes

• **Value** (*string*) – **[REQUIRED]**

Values corresponding to the additional parameter names required by some top-level domains.

Type: String

Default: None

Constraints: Maximum 2048 characters.

Parent: ExtraParams

Required: Yes

• **TechContact** (*dict*) – **[REQUIRED]**

Provides detailed contact information.

Type: Complex

Children: `FirstName` , `MiddleName` , `LastName` , `ContactType` , `OrganizationName` , `AddressLine1` , `AddressLine2` , `City` , `State` , `CountryCode` , `ZipCode` , `PhoneNumber` , `Email` , `Fax` , `ExtraParams`

Required: Yes

– **FirstName** (*string*) –

First name of contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: `RegistrantContact` , `AdminContact` , `TechContact`

Required: Yes

– **LastName** (*string*) –

Last name of contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: `RegistrantContact` , `AdminContact` , `TechContact`

Required: Yes

– **ContactType** (*string*) –

Indicates whether the contact is a person, company, association, or public organization. If you choose an option other than `PERSON` , you must enter an organization name, and you can't enable privacy protection for the contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Valid values: `PERSON` | `COMPANY` | `ASSOCIATION` | `PUBLIC_BODY`

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **OrganizationName** (*string*) –

Name of the organization for contact types other than PERSON .

Type: String

Default: None

Constraints: Maximum 255 characters. Contact type must not be PERSON .

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **AddressLine1** (*string*) –

First line of the contact’s address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **AddressLine2** (*string*) –

Second line of contact’s address, if any.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **City** (*string*) –

The city of the contact’s address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **State** (*string*) –

The state or province of the contact’s city.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **CountryCode** (*string*) –

Code for the country of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **ZipCode** (*string*) –

The zip or postal code of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **PhoneNumber** (*string*) –

The phone number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format “[+country dialing code].[number including any area code]”. For example, a US phone number might appear as “+1.1234567890”.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **Email** (*string*) –

Email address of the contact.

Type: String

Default: None

Constraints: Maximum 254 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **Fax** (*string*) –

Fax number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format “[+country dialing code].[number including any area code]”. For example, a US phone number might appear as “+1.1234567890”.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **ExtraParams** (*list*) –

A list of name-value pairs for parameters required by certain top-level domains.

Type: Complex

Default: None

Parents: RegistrantContact, AdminContact, TechContact

Children: Name, Value

Required: No

* (*dict*) –

ExtraParam includes the following elements.

· **Name** (*string*) – **[REQUIRED]**

Name of the additional parameter required by the top-level domain.

Type: String

Default: None

Valid values: DUNS_NUMBER |
BRAND_NUMBER | BIRTH_DEPARTMENT |
BIRTH_DATE_IN_YYYY_MM_DD | BIRTH_COUNTRY
| BIRTH_CITY | DOCUMENT_NUMBER |
AU_ID_NUMBER | AU_ID_TYPE | CA_LEGAL_TYPE
| ES_IDENTIFICATION | ES_IDENTIFICATION_TYPE
| ES_LEGAL_FORM | FI_BUSINESS_NUMBER |
FI_ID_NUMBER | IT_PIN | RU_PASSPORT_DATA |
SE_ID_NUMBER | SG_ID_NUMBER | VAT_NUMBER

Parent: ExtraParams

Required: Yes

· **Value** (*string*) – **[REQUIRED]**

Values corresponding to the additional parameter names required by some top-level domains.

Type: String

Default: None

Constraints: Maximum 2048 characters.

Parent: ExtraParams

Required: Yes

- **PrivacyProtectAdminContact** (*boolean*) – Whether you want to conceal contact information from WHOIS queries. If you specify true, WHOIS (“who is”) queries will return contact information for our registrar partner, Gandi, instead of the contact information that you enter.

Type: Boolean

Default: true

Valid values: true | false

Required: No

- **PrivacyProtectRegistrantContact** (*boolean*) – Whether you want to conceal contact information from WHOIS queries. If you specify true, WHOIS (“who is”) queries will return contact information for our registrar partner, Gandi, instead of the contact information that you enter.

Type: Boolean

Default: true

Valid values: true | false

Required: No

- **PrivacyProtectTechContact** (*boolean*) – Whether you want to conceal contact information from WHOIS queries. If you specify true, WHOIS (“who is”) queries will return contact information for our registrar partner, Gandi, instead of the contact information that you enter.

Type: Boolean

Default: true

Valid values: true | false

Required: No

Return type dict

Returns

Response Syntax

```
{
    'OperationId': 'string'
}
```

Response Structure

- (*dict*) –

The RegisterDomain response includes the following element.

- **OperationId** (*string*) –

Identifier for tracking the progress of the request. To use this ID to query the operation status, use GetOperationDetail.

Type: String

Default: None

Constraints: Maximum 255 characters.

retrieve_domain_auth_code (***kwargs*)

This operation returns the AuthCode for the domain. To transfer a domain to another registrar, you provide this value to the new registrar.

Request Syntax

```
response = client.retrieve_domain_auth_code(
    DomainName='string'
)
```

Parameters **DomainName** (*string*) – [REQUIRED]

The name of a domain.

Type: String

Default: None

Constraints: The domain name can contain only the letters a through z, the numbers 0 through 9, and hyphen (-). Internationalized Domain Names are not supported.

Required: Yes

Return type dict

Returns

Response Syntax

```
{
    'AuthCode': 'string'
}
```

Response Structure

- (dict) –

The RetrieveDomainAuthCode response includes the following element.

- **AuthCode** (*string*) –

The authorization code for the domain.

Type: String

transfer_domain (**kwargs)

This operation transfers a domain from another registrar to Amazon Route 53. When the transfer is complete, the domain is registered with the AWS registrar partner, Gandi.

For transfer requirements, a detailed procedure, and information about viewing the status of a domain transfer, see [Transferring Registration for a Domain to Amazon Route 53](#) in the Amazon Route 53 Developer Guide.

If the registrar for your domain is also the DNS service provider for the domain, we highly recommend that you consider transferring your DNS service to Amazon Route 53 or to another DNS service provider before you transfer your registration. Some registrars provide free DNS service when you purchase a domain registration. When you transfer the registration, the previous registrar will not renew your domain registration and could end your DNS service at any time.

Note: Caution! If the registrar for your domain is also the DNS service provider for the domain and you don't transfer DNS service to another provider, your website, email, and the web applications associated with the domain might become unavailable.

If the transfer is successful, this method returns an operation ID that you can use to track the progress and completion of the action. If the transfer doesn't complete successfully, the domain registrant will be notified by email.

Request Syntax

```
response = client.transfer_domain(
    DomainName='string',
    IdnLangCode='string',
    DurationInYears=123,
    Nameservers=[
        {
            'Name': 'string',
            'GlueIps': [
                'string',
            ]
        },
    ],
)
```

```

],
AuthCode='string',
AutoRenew=True|False,
AdminContact={
    'FirstName': 'string',
    'LastName': 'string',
    'ContactType': 'PERSON'|'COMPANY'|'ASSOCIATION'|'PUBLIC_BODY'|'RESELLER',
    'OrganizationName': 'string',
    'AddressLine1': 'string',
    'AddressLine2': 'string',
    'City': 'string',
    'State': 'string',
    'CountryCode': 'AD'|'AE'|'AF'|'AG'|'AI'|'AL'|'AM'|'AN'|'AO'|'AQ'|'AR'|'AS'|'AT'|'AU',
    'ZipCode': 'string',
    'PhoneNumber': 'string',
    'Email': 'string',
    'Fax': 'string',
    'ExtraParams': [
        {
            'Name': 'DUNS_NUMBER'|'BRAND_NUMBER'|'BIRTH_DEPARTMENT'|'BIRTH_DATE_IN_YYYY',
            'Value': 'string'
        }
    ]
},
RegistrantContact={
    'FirstName': 'string',
    'LastName': 'string',
    'ContactType': 'PERSON'|'COMPANY'|'ASSOCIATION'|'PUBLIC_BODY'|'RESELLER',
    'OrganizationName': 'string',
    'AddressLine1': 'string',
    'AddressLine2': 'string',
    'City': 'string',
    'State': 'string',
    'CountryCode': 'AD'|'AE'|'AF'|'AG'|'AI'|'AL'|'AM'|'AN'|'AO'|'AQ'|'AR'|'AS'|'AT'|'AU',
    'ZipCode': 'string',
    'PhoneNumber': 'string',
    'Email': 'string',
    'Fax': 'string',
    'ExtraParams': [
        {
            'Name': 'DUNS_NUMBER'|'BRAND_NUMBER'|'BIRTH_DEPARTMENT'|'BIRTH_DATE_IN_YYYY',
            'Value': 'string'
        }
    ]
},
TechContact={
    'FirstName': 'string',
    'LastName': 'string',
    'ContactType': 'PERSON'|'COMPANY'|'ASSOCIATION'|'PUBLIC_BODY'|'RESELLER',
    'OrganizationName': 'string',
    'AddressLine1': 'string',
    'AddressLine2': 'string',
    'City': 'string',
    'State': 'string',
    'CountryCode': 'AD'|'AE'|'AF'|'AG'|'AI'|'AL'|'AM'|'AN'|'AO'|'AQ'|'AR'|'AS'|'AT'|'AU',
    'ZipCode': 'string',
    'PhoneNumber': 'string',
    'Email': 'string',

```

```
        'Fax': 'string',
        'ExtraParams': [
            {
                'Name': 'DUNS_NUMBER' | 'BRAND_NUMBER' | 'BIRTH_DEPARTMENT' | 'BIRTH_DATE_IN_YYYY',
                'Value': 'string'
            },
        ],
    },
    PrivacyProtectAdminContact=True|False,
    PrivacyProtectRegistrantContact=True|False,
    PrivacyProtectTechContact=True|False
)
```

Parameters

- **DomainName** (*string*) – **[REQUIRED]**

The name of a domain.

Type: String

Default: None

Constraints: The domain name can contain only the letters a through z, the numbers 0 through 9, and hyphen (-). Internationalized Domain Names are not supported.

Required: Yes

- **IdnLangCode** (*string*) – Reserved for future use.

- **DurationInYears** (*integer*) – **[REQUIRED]**

The number of years the domain will be registered. Domains are registered for a minimum of one year. The maximum period depends on the top-level domain.

Type: Integer

Default: 1

Valid values: Integer from 1 to 10

Required: Yes

- **Nameservers** (*list*) – Contains details for the host and glue IP addresses.

Type: Complex

Children: GlueIps, Name

Required: No

– (*dict*) –

Nameserver includes the following elements.

- * **Name** (*string*) – **[REQUIRED]**

The fully qualified host name of the name server.

Type: String

Constraint: Maximum 255 characters

Parent: Nameservers

- * **GlueIps** (*list*) –

Glue IP address of a name server entry. Glue IP addresses are required only when the name of the name server is a subdomain of the domain. For example, if your domain is example.com and the name

server for the domain is ns.example.com, you need to specify the IP address for ns.example.com.

Type: List of IP addresses.

Constraints: The list can contain only one IPv4 and one IPv6 address.

Parent: Nameservers

· *(string)* –

- **AuthCode** (*string*) – The authorization code for the domain. You get this value from the current registrar.

Type: String

Required: Yes

- **AutoRenew** (*boolean*) – Indicates whether the domain will be automatically renewed (true) or not (false). Autorenewal only takes effect after the account is charged.

Type: Boolean

Valid values: true | false

Default: true

Required: No

- **AdminContact** (*dict*) – **[REQUIRED]**

Provides detailed contact information.

Type: Complex

Children: `FirstName` , `MiddleName` , `LastName` , `ContactType` , `OrganizationName` , `AddressLine1` , `AddressLine2` , `City` , `State` , `CountryCode` , `ZipCode` , `PhoneNumber` , `Email` , `Fax` , `ExtraParams`

Required: Yes

- **FirstName** (*string*) –

First name of contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: `RegistrantContact` , `AdminContact` , `TechContact`

Required: Yes

- **LastName** (*string*) –

Last name of contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: `RegistrantContact` , `AdminContact` , `TechContact`

Required: Yes

– **ContactType** (*string*) –

Indicates whether the contact is a person, company, association, or public organization. If you choose an option other than `PERSON`, you must enter an organization name, and you can't enable privacy protection for the contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Valid values: `PERSON` | `COMPANY` | `ASSOCIATION` | `PUBLIC_BODY`

Parents: `RegistrantContact`, `AdminContact`, `TechContact`

Required: Yes

– **OrganizationName** (*string*) –

Name of the organization for contact types other than `PERSON`.

Type: String

Default: None

Constraints: Maximum 255 characters. Contact type must not be `PERSON`.

Parents: `RegistrantContact`, `AdminContact`, `TechContact`

Required: No

– **AddressLine1** (*string*) –

First line of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: `RegistrantContact`, `AdminContact`, `TechContact`

Required: Yes

– **AddressLine2** (*string*) –

Second line of contact's address, if any.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: `RegistrantContact`, `AdminContact`, `TechContact`

Required: No

– **City** (*string*) –

The city of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: `RegistrantContact`, `AdminContact`, `TechContact`

Required: Yes

– **State** (*string*) –

The state or province of the contact's city.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **CountryCode** (*string*) –

Code for the country of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **ZipCode** (*string*) –

The zip or postal code of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **PhoneNumber** (*string*) –

The phone number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format “[+country dialing code].[number including any area code]”. For example, a US phone number might appear as “+1.1234567890”.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **Email** (*string*) –

Email address of the contact.

Type: String

Default: None

Constraints: Maximum 254 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **Fax** (*string*) –

Fax number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format “[+country dialing code].[number including any area code]”. For example, a US phone number might appear as “+1.1234567890”.

Parents: RegistrantContact, AdminContact, TechContact

Required: No

– **ExtraParams** (*list*) –

A list of name-value pairs for parameters required by certain top-level domains.

Type: Complex

Default: None

Parents: RegistrantContact, AdminContact, TechContact

Children: Name, Value

Required: No

* (*dict*) –

ExtraParam includes the following elements.

· **Name** (*string*) – **[REQUIRED]**

Name of the additional parameter required by the top-level domain.

Type: String

Default: None

Valid values: DUNS_NUMBER |
BRAND_NUMBER | BIRTH_DEPARTMENT |
BIRTH_DATE_IN_YYYY_MM_DD | BIRTH_COUNTRY
| BIRTH_CITY | DOCUMENT_NUMBER |
AU_ID_NUMBER | AU_ID_TYPE | CA_LEGAL_TYPE
| ES_IDENTIFICATION | ES_IDENTIFICATION_TYPE
| ES_LEGAL_FORM | FI_BUSINESS_NUMBER |
FI_ID_NUMBER | IT_PIN | RU_PASSPORT_DATA |
SE_ID_NUMBER | SG_ID_NUMBER | VAT_NUMBER

Parent: ExtraParams

Required: Yes

· **Value** (*string*) – **[REQUIRED]**

Values corresponding to the additional parameter names required by some top-level domains.

Type: String

Default: None

Constraints: Maximum 2048 characters.

Parent: ExtraParams

Required: Yes

- **RegistrantContact** (*dict*) – **[REQUIRED]**

Provides detailed contact information.

Type: Complex

Children: `FirstName` , `MiddleName` , `LastName` , `ContactType` , `OrganizationName` , `AddressLine1` , `AddressLine2` , `City` , `State` , `CountryCode` , `ZipCode` , `PhoneNumber` , `Email` , `Fax` , `ExtraParams`

Required: Yes

- **FirstName** (*string*) –

First name of contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: `RegistrantContact` , `AdminContact` , `TechContact`

Required: Yes

- **LastName** (*string*) –

Last name of contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: `RegistrantContact` , `AdminContact` , `TechContact`

Required: Yes

- **ContactType** (*string*) –

Indicates whether the contact is a person, company, association, or public organization. If you choose an option other than `PERSON` , you must enter an organization name, and you can't enable privacy protection for the contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Valid values: `PERSON` | `COMPANY` | `ASSOCIATION` | `PUBLIC_BODY`

Parents: `RegistrantContact` , `AdminContact` , `TechContact`

Required: Yes

- **OrganizationName** (*string*) –

Name of the organization for contact types other than `PERSON` .

Type: String

Default: None

Constraints: Maximum 255 characters. Contact type must not be `PERSON` .

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **AddressLine1** (*string*) –

First line of the contact’s address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **AddressLine2** (*string*) –

Second line of contact’s address, if any.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **City** (*string*) –

The city of the contact’s address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **State** (*string*) –

The state or province of the contact’s city.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **CountryCode** (*string*) –

Code for the country of the contact’s address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **ZipCode** (*string*) –

The zip or postal code of the contact’s address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **PhoneNumber** (*string*) –

The phone number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format “[+country dialing code].[number including any area code]”. For example, a US phone number might appear as “+1.1234567890” .

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **Email** (*string*) –

Email address of the contact.

Type: String

Default: None

Constraints: Maximum 254 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **Fax** (*string*) –

Fax number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format “[+country dialing code].[number including any area code]”. For example, a US phone number might appear as “+1.1234567890” .

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **ExtraParams** (*list*) –

A list of name-value pairs for parameters required by certain top-level domains.

Type: Complex

Default: None

Parents: RegistrantContact , AdminContact , TechContact

Children: Name , Value

Required: No

* (*dict*) –

ExtraParam includes the following elements.

- **Name** (*string*) – **[REQUIRED]**

Name of the additional parameter required by the top-level domain.

Type: String

Default: None

Valid values: DUNS_NUMBER |
BRAND_NUMBER | BIRTH_DEPARTMENT |
BIRTH_DATE_IN_YYYY_MM_DD | BIRTH_COUNTRY |
BIRTH_CITY | DOCUMENT_NUMBER |
AU_ID_NUMBER | AU_ID_TYPE | CA_LEGAL_TYPE |
ES_IDENTIFICATION | ES_IDENTIFICATION_TYPE |
ES_LEGAL_FORM | FI_BUSINESS_NUMBER |
FI_ID_NUMBER | IT_PIN | RU_PASSPORT_DATA |
SE_ID_NUMBER | SG_ID_NUMBER | VAT_NUMBER

Parent: ExtraParams

Required: Yes

- **Value** (*string*) – **[REQUIRED]**

Values corresponding to the additional parameter names required by some top-level domains.

Type: String

Default: None

Constraints: Maximum 2048 characters.

Parent: ExtraParams

Required: Yes

- **TechContact** (*dict*) – **[REQUIRED]**

Provides detailed contact information.

Type: Complex

Children: FirstName , MiddleName , LastName , ContactType ,
OrganizationName , AddressLine1 , AddressLine2 , City ,
State , CountryCode , ZipCode , PhoneNumber , Email , Fax ,
ExtraParams

Required: Yes

- **FirstName** (*string*) –

First name of contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

- **LastName** (*string*) –
Last name of contact.
Type: String
Default: None
Constraints: Maximum 255 characters.
Parents: RegistrantContact , AdminContact , TechContact
Required: Yes
- **ContactType** (*string*) –
Indicates whether the contact is a person, company, association, or public organization. If you choose an option other than PERSON , you must enter an organization name, and you can't enable privacy protection for the contact.
Type: String
Default: None
Constraints: Maximum 255 characters.
Valid values: PERSON | COMPANY | ASSOCIATION | PUBLIC_BODY
Parents: RegistrantContact , AdminContact , TechContact
Required: Yes
- **OrganizationName** (*string*) –
Name of the organization for contact types other than PERSON .
Type: String
Default: None
Constraints: Maximum 255 characters. Contact type must not be PERSON .
Parents: RegistrantContact , AdminContact , TechContact
Required: No
- **AddressLine1** (*string*) –
First line of the contact's address.
Type: String
Default: None
Constraints: Maximum 255 characters.
Parents: RegistrantContact , AdminContact , TechContact
Required: Yes
- **AddressLine2** (*string*) –
Second line of contact's address, if any.
Type: String
Default: None
Constraints: Maximum 255 characters.
Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **City** (*string*) –

The city of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **State** (*string*) –

The state or province of the contact's city.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **CountryCode** (*string*) –

Code for the country of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **ZipCode** (*string*) –

The zip or postal code of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **PhoneNumber** (*string*) –

The phone number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format “[+country dialing code].[number including any area code]”. For example, a US phone number might appear as “+1.1234567890”.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **Email** (*string*) –

Email address of the contact.

Type: String

Default: None

Constraints: Maximum 254 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **Fax** (*string*) –

Fax number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format “[+country dialing code].[number including any area code]”. For example, a US phone number might appear as “+1.1234567890”.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **ExtraParams** (*list*) –

A list of name-value pairs for parameters required by certain top-level domains.

Type: Complex

Default: None

Parents: RegistrantContact , AdminContact , TechContact

Children: Name , Value

Required: No

* (*dict*) –

ExtraParam includes the following elements.

· **Name** (*string*) – **[REQUIRED]**

Name of the additional parameter required by the top-level domain.

Type: String

Default: None

Valid	values:	DUNS_NUMBER	
BRAND_NUMBER		BIRTH_DEPARTMENT	
BIRTH_DATE_IN_YYYY_MM_DD		BIRTH_COUNTRY	
BIRTH_CITY		DOCUMENT_NUMBER	
AU_ID_NUMBER		AU_ID_TYPE	
CA_LEGAL_TYPE		ES_IDENTIFICATION	
ES_IDENTIFICATION_TYPE		ES_LEGAL_FORM	
FI_BUSINESS_NUMBER		FI_ID_NUMBER	
IT_PIN		RU_PASSPORT_DATA	
SE_ID_NUMBER		SG_ID_NUMBER	
VAT_NUMBER			

Parent: ExtraParams

Required: Yes

• **Value** (*string*) – **[REQUIRED]**

Values corresponding to the additional parameter names required by some top-level domains.

Type: String

Default: None

Constraints: Maximum 2048 characters.

Parent: ExtraParams

Required: Yes

- **PrivacyProtectAdminContact** (*boolean*) – Whether you want to conceal contact information from WHOIS queries. If you specify true, WHOIS (“who is”) queries will return contact information for our registrar partner, Gandi, instead of the contact information that you enter.

Type: Boolean

Default: true

Valid values: true | false

Required: No

- **PrivacyProtectRegistrantContact** (*boolean*) – Whether you want to conceal contact information from WHOIS queries. If you specify true, WHOIS (“who is”) queries will return contact information for our registrar partner, Gandi, instead of the contact information that you enter.

Type: Boolean

Default: true

Valid values: true | false

Required: No

- **PrivacyProtectTechContact** (*boolean*) – Whether you want to conceal contact information from WHOIS queries. If you specify true, WHOIS (“who is”) queries will return contact information for our registrar partner, Gandi, instead of the contact information that you enter.

Type: Boolean

Default: true

Valid values: true | false

Required: No

Return type dict

Returns

Response Syntax

```
{
    'OperationId': 'string'
}
```

Response Structure

- (*dict*) –

The TranserDomain response includes the following element.

– **OperationId** (*string*) –

Identifier for tracking the progress of the request. To use this ID to query the operation status, use `GetOperationDetail`.

Type: String

Default: None

Constraints: Maximum 255 characters.

update_domain_contact (***kwargs*)

This operation updates the contact information for a particular domain. Information for at least one contact (registrant, administrator, or technical) must be supplied for update.

If the update is successful, this method returns an operation ID that you can use to track the progress and completion of the action. If the request is not completed successfully, the domain registrant will be notified by email.

Request Syntax

```
response = client.update_domain_contact(
    DomainName='string',
    AdminContact={
        'FirstName': 'string',
        'LastName': 'string',
        'ContactType': 'PERSON'|'COMPANY'|'ASSOCIATION'|'PUBLIC_BODY'|'RESELLER',
        'OrganizationName': 'string',
        'AddressLine1': 'string',
        'AddressLine2': 'string',
        'City': 'string',
        'State': 'string',
        'CountryCode': 'AD'|'AE'|'AF'|'AG'|'AI'|'AL'|'AM'|'AN'|'AO'|'AQ'|'AR'|'AS'|'AT'|'AU',
        'ZipCode': 'string',
        'PhoneNumber': 'string',
        'Email': 'string',
        'Fax': 'string',
        'ExtraParams': [
            {
                'Name': 'DUNS_NUMBER'|'BRAND_NUMBER'|'BIRTH_DEPARTMENT'|'BIRTH_DATE_IN_YYYY',
                'Value': 'string'
            },
        ],
    },
    RegistrantContact={
        'FirstName': 'string',
        'LastName': 'string',
        'ContactType': 'PERSON'|'COMPANY'|'ASSOCIATION'|'PUBLIC_BODY'|'RESELLER',
        'OrganizationName': 'string',
        'AddressLine1': 'string',
        'AddressLine2': 'string',
        'City': 'string',
        'State': 'string',
        'CountryCode': 'AD'|'AE'|'AF'|'AG'|'AI'|'AL'|'AM'|'AN'|'AO'|'AQ'|'AR'|'AS'|'AT'|'AU',
        'ZipCode': 'string',
        'PhoneNumber': 'string',
        'Email': 'string',
        'Fax': 'string',
        'ExtraParams': [
            {
```

```
        'Name': 'DUNS_NUMBER' | 'BRAND_NUMBER' | 'BIRTH_DEPARTMENT' | 'BIRTH_DATE_IN_YYYY',
        'Value': 'string'
    },
]
},
TechContact={
    'FirstName': 'string',
    'LastName': 'string',
    'ContactType': 'PERSON' | 'COMPANY' | 'ASSOCIATION' | 'PUBLIC_BODY' | 'RESELLER',
    'OrganizationName': 'string',
    'AddressLine1': 'string',
    'AddressLine2': 'string',
    'City': 'string',
    'State': 'string',
    'CountryCode': 'AD' | 'AE' | 'AF' | 'AG' | 'AI' | 'AL' | 'AM' | 'AN' | 'AO' | 'AQ' | 'AR' | 'AS' | 'AT' | 'AU',
    'ZipCode': 'string',
    'PhoneNumber': 'string',
    'Email': 'string',
    'Fax': 'string',
    'ExtraParams': [
        {
            'Name': 'DUNS_NUMBER' | 'BRAND_NUMBER' | 'BIRTH_DEPARTMENT' | 'BIRTH_DATE_IN_YYYY',
            'Value': 'string'
        },
    ]
}
)
```

Parameters

- **DomainName** (*string*) – **[REQUIRED]**

The name of a domain.

Type: String

Default: None

Constraints: The domain name can contain only the letters a through z, the numbers 0 through 9, and hyphen (-). Internationalized Domain Names are not supported.

Required: Yes

- **AdminContact** (*dict*) – Provides detailed contact information.

Type: Complex

Children: `FirstName` , `MiddleName` , `LastName` , `ContactType` , `OrganizationName` , `AddressLine1` , `AddressLine2` , `City` , `State` , `CountryCode` , `ZipCode` , `PhoneNumber` , `Email` , `Fax` , `ExtraParams`

Required: Yes

- **FirstName** (*string*) –

First name of contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **LastName** (*string*) –

Last name of contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **ContactType** (*string*) –

Indicates whether the contact is a person, company, association, or public organization. If you choose an option other than PERSON , you must enter an organization name, and you can't enable privacy protection for the contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Valid values: PERSON | COMPANY | ASSOCIATION | PUBLIC_BODY

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **OrganizationName** (*string*) –

Name of the organization for contact types other than PERSON .

Type: String

Default: None

Constraints: Maximum 255 characters. Contact type must not be PERSON .

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **AddressLine1** (*string*) –

First line of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **AddressLine2** (*string*) –

Second line of contact's address, if any.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **City** (*string*) –

The city of the contact’s address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **State** (*string*) –

The state or province of the contact’s city.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **CountryCode** (*string*) –

Code for the country of the contact’s address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **ZipCode** (*string*) –

The zip or postal code of the contact’s address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **PhoneNumber** (*string*) –

The phone number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format “+[country dialing code].[number including any area code]”. For example, a US phone number might appear as “+1.1234567890”.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **Email** (*string*) –

Email address of the contact.

Type: String

Default: None

Constraints: Maximum 254 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **Fax** (*string*) –

Fax number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format “[+country dialing code].[number including any area code]”. For example, a US phone number might appear as “+1.1234567890”.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **ExtraParams** (*list*) –

A list of name-value pairs for parameters required by certain top-level domains.

Type: Complex

Default: None

Parents: RegistrantContact , AdminContact , TechContact

Children: Name , Value

Required: No

* (*dict*) –

ExtraParam includes the following elements.

· **Name** (*string*) – **[REQUIRED]**

Name of the additional parameter required by the top-level domain.

Type: String

Default: None

Valid	values:	DUNS_NUMBER	
BRAND_NUMBER		BIRTH_DEPARTMENT	
BIRTH_DATE_IN_YYYY_MM_DD		BIRTH_COUNTRY	
BIRTH_CITY		DOCUMENT_NUMBER	
AU_ID_NUMBER		AU_ID_TYPE	
CA_LEGAL_TYPE		ES_IDENTIFICATION	
ES_IDENTIFICATION_TYPE		ES_LEGAL_FORM	
FI_BUSINESS_NUMBER		FI_ID_NUMBER	
IT_PIN		RU_PASSPORT_DATA	
SE_ID_NUMBER		SG_ID_NUMBER	
VAT_NUMBER			

Parent: ExtraParams

Required: Yes

• **Value** (*string*) – [REQUIRED]

Values corresponding to the additional parameter names required by some top-level domains.

Type: String

Default: None

Constraints: Maximum 2048 characters.

Parent: ExtraParams

Required: Yes

• **RegistrantContact** (*dict*) – Provides detailed contact information.

Type: Complex

Children: `FirstName` , `MiddleName` , `LastName` , `ContactType` , `OrganizationName` , `AddressLine1` , `AddressLine2` , `City` , `State` , `CountryCode` , `ZipCode` , `PhoneNumber` , `Email` , `Fax` , `ExtraParams`

Required: Yes

– **FirstName** (*string*) –

First name of contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: `RegistrantContact` , `AdminContact` , `TechContact`

Required: Yes

– **LastName** (*string*) –

Last name of contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: `RegistrantContact` , `AdminContact` , `TechContact`

Required: Yes

– **ContactType** (*string*) –

Indicates whether the contact is a person, company, association, or public organization. If you choose an option other than `PERSON` , you must enter an organization name, and you can't enable privacy protection for the contact.

Type: String

Default: None

Constraints: Maximum 255 characters.

Valid values: `PERSON` | `COMPANY` | `ASSOCIATION` | `PUBLIC_BODY`

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **OrganizationName** (*string*) –

Name of the organization for contact types other than PERSON .

Type: String

Default: None

Constraints: Maximum 255 characters. Contact type must not be PERSON .

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **AddressLine1** (*string*) –

First line of the contact’s address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **AddressLine2** (*string*) –

Second line of contact’s address, if any.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **City** (*string*) –

The city of the contact’s address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **State** (*string*) –

The state or province of the contact’s city.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **CountryCode** (*string*) –

Code for the country of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **ZipCode** (*string*) –

The zip or postal code of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **PhoneNumber** (*string*) –

The phone number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format “[+country dialing code].[number including any area code]”. For example, a US phone number might appear as “+1.1234567890”.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **Email** (*string*) –

Email address of the contact.

Type: String

Default: None

Constraints: Maximum 254 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **Fax** (*string*) –

Fax number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format “[+country dialing code].[number including any area code]”. For example, a US phone number might appear as “+1.1234567890”.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **ExtraParams** (*list*) –

A list of name-value pairs for parameters required by certain top-level domains.

Type: Complex

Default: None

Parents: RegistrantContact , AdminContact , TechContact

Children: Name , Value

Required: No

* (*dict*) –

ExtraParam includes the following elements.

· **Name** (*string*) – **[REQUIRED]**

Name of the additional parameter required by the top-level domain.

Type: String

Default: None

Valid values: DUNS_NUMBER |
 BRAND_NUMBER | BIRTH_DEPARTMENT |
 BIRTH_DATE_IN_YYYY_MM_DD | BIRTH_COUNTRY
 | BIRTH_CITY | DOCUMENT_NUMBER |
 AU_ID_NUMBER | AU_ID_TYPE | CA_LEGAL_TYPE
 | ES_IDENTIFICATION | ES_IDENTIFICATION_TYPE
 | ES_LEGAL_FORM | FI_BUSINESS_NUMBER |
 FI_ID_NUMBER | IT_PIN | RU_PASSPORT_DATA |
 SE_ID_NUMBER | SG_ID_NUMBER | VAT_NUMBER

Parent: ExtraParams

Required: Yes

· **Value** (*string*) – **[REQUIRED]**

Values corresponding to the additional parameter names required by some top-level domains.

Type: String

Default: None

Constraints: Maximum 2048 characters.

Parent: ExtraParams

Required: Yes

• **TechContact** (*dict*) – Provides detailed contact information.

Type: Complex

Children: FirstName , MiddleName , LastName , ContactType
 , OrganizationName , AddressLine1 , AddressLine2 , City ,
 State , CountryCode , ZipCode , PhoneNumber , Email , Fax ,
 ExtraParams

Required: Yes

- **FirstName** (*string*) –
First name of contact.
Type: String
Default: None
Constraints: Maximum 255 characters.
Parents: RegistrantContact , AdminContact , TechContact
Required: Yes
- **LastName** (*string*) –
Last name of contact.
Type: String
Default: None
Constraints: Maximum 255 characters.
Parents: RegistrantContact , AdminContact , TechContact
Required: Yes
- **ContactType** (*string*) –
Indicates whether the contact is a person, company, association, or public organization. If you choose an option other than PERSON , you must enter an organization name, and you can't enable privacy protection for the contact.
Type: String
Default: None
Constraints: Maximum 255 characters.
Valid values: PERSON | COMPANY | ASSOCIATION | PUBLIC_BODY
Parents: RegistrantContact , AdminContact , TechContact
Required: Yes
- **OrganizationName** (*string*) –
Name of the organization for contact types other than PERSON .
Type: String
Default: None
Constraints: Maximum 255 characters. Contact type must not be PERSON .
Parents: RegistrantContact , AdminContact , TechContact
Required: No
- **AddressLine1** (*string*) –
First line of the contact's address.
Type: String
Default: None
Constraints: Maximum 255 characters.
Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **AddressLine2** (*string*) –

Second line of contact's address, if any.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **City** (*string*) –

The city of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **State** (*string*) –

The state or province of the contact's city.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **CountryCode** (*string*) –

Code for the country of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **ZipCode** (*string*) –

The zip or postal code of the contact's address.

Type: String

Default: None

Constraints: Maximum 255 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **PhoneNumber** (*string*) –

The phone number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format “[country dialing code].[number including any area code]”. For example, a US phone number might appear as “+1.1234567890”.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **Email** (*string*) –

Email address of the contact.

Type: String

Default: None

Constraints: Maximum 254 characters.

Parents: RegistrantContact , AdminContact , TechContact

Required: Yes

– **Fax** (*string*) –

Fax number of the contact.

Type: String

Default: None

Constraints: Phone number must be specified in the format “[country dialing code].[number including any area code]”. For example, a US phone number might appear as “+1.1234567890”.

Parents: RegistrantContact , AdminContact , TechContact

Required: No

– **ExtraParams** (*list*) –

A list of name-value pairs for parameters required by certain top-level domains.

Type: Complex

Default: None

Parents: RegistrantContact , AdminContact , TechContact

Children: Name , Value

Required: No

* (*dict*) –

ExtraParam includes the following elements.

· **Name** (*string*) – **[REQUIRED]**

Name of the additional parameter required by the top-level domain.

Type: String

Default: None

Valid values: DUNS_NUMBER |
 BRAND_NUMBER | BIRTH_DEPARTMENT |
 BIRTH_DATE_IN_YYYY_MM_DD | BIRTH_COUNTRY
 | BIRTH_CITY | DOCUMENT_NUMBER |
 AU_ID_NUMBER | AU_ID_TYPE | CA_LEGAL_TYPE
 | ES_IDENTIFICATION | ES_IDENTIFICATION_TYPE
 | ES_LEGAL_FORM | FI_BUSINESS_NUMBER |
 FI_ID_NUMBER | IT_PIN | RU_PASSPORT_DATA |
 SE_ID_NUMBER | SG_ID_NUMBER | VAT_NUMBER

Parent: ExtraParams

Required: Yes

• **Value** (*string*) – **[REQUIRED]**

Values corresponding to the additional parameter names required by some top-level domains.

Type: String

Default: None

Constraints: Maximum 2048 characters.

Parent: ExtraParams

Required: Yes

Return type dict

Returns

Response Syntax

```
{
    'OperationId': 'string'
}
```

Response Structure

- (*dict*) –

The UpdateDomainContact response includes the following element.

- **OperationId** (*string*) –

Identifier for tracking the progress of the request. To use this ID to query the operation status, use GetOperationDetail.

Type: String

Default: None

Constraints: Maximum 255 characters.

update_domain_contact_privacy (***kwargs*)

This operation updates the specified domain contact's privacy setting. When the privacy option is enabled, personal information such as postal or email address is hidden from the results of a public WHOIS query. The privacy services are provided by the AWS registrar, Gandi. For more information, see the [Gandi privacy features](#).

This operation only affects the privacy of the specified contact type (registrant, administrator, or tech). Successful acceptance returns an operation ID that you can use with GetOperationDetail to track the progress and completion of the action. If the request is not completed successfully, the domain registrant will be notified by email.

Request Syntax

```
response = client.update_domain_contact_privacy(  
    DomainName='string',  
    AdminPrivacy=True|False,  
    RegistrantPrivacy=True|False,  
    TechPrivacy=True|False  
)
```

Parameters

- **DomainName** (*string*) – [REQUIRED]

The name of a domain.

Type: String

Default: None

Constraints: The domain name can contain only the letters a through z, the numbers 0 through 9, and hyphen (-). Internationalized Domain Names are not supported.

Required: Yes

- **AdminPrivacy** (*boolean*) – Whether you want to conceal contact information from WHOIS queries. If you specify true, WHOIS (“who is”) queries will return contact information for our registrar partner, Gandi, instead of the contact information that you enter.

Type: Boolean

Default: None

Valid values: true|false

Required: No

- **RegistrantPrivacy** (*boolean*) – Whether you want to conceal contact information from WHOIS queries. If you specify true, WHOIS (“who is”) queries will return contact information for our registrar partner, Gandi, instead of the contact information that you enter.

Type: Boolean

Default: None

Valid values: true|false

Required: No

- **TechPrivacy** (*boolean*) – Whether you want to conceal contact information from WHOIS queries. If you specify true, WHOIS (“who is”) queries will return contact information for our registrar partner, Gandi, instead of the contact information that you enter.

Type: Boolean

Default: None

Valid values: true|false

Required: No

Return type dict

Returns

Response Syntax

```
{
    'OperationId': 'string'
}
```

Response Structure

- (*dict*) –

The UpdateDomainContactPrivacy response includes the following element.

- **OperationId** (*string*) –

Identifier for tracking the progress of the request. To use this ID to query the operation status, use GetOperationDetail.

Type: String

Default: None

Constraints: Maximum 255 characters.

update_domain_nameservers (***kwargs*)

This operation replaces the current set of name servers for the domain with the specified set of name servers. If you use Amazon Route 53 as your DNS service, specify the four name servers in the delegation set for the hosted zone for the domain.

If successful, this operation returns an operation ID that you can use to track the progress and completion of the action. If the request is not completed successfully, the domain registrant will be notified by email.

Request Syntax

```
response = client.update_domain_nameservers(
    DomainName='string',
    FIAuthKey='string',
    Nameservers=[
        {
            'Name': 'string',
            'GlueIps': [
                'string',
            ]
        },
    ],
)
```

Parameters

- **DomainName** (*string*) – [REQUIRED]

The name of a domain.

Type: String

Default: None

Constraints: The domain name can contain only the letters a through z, the numbers 0 through 9, and hyphen (-). Internationalized Domain Names are not supported.

Required: Yes

- **FIAuthKey** (*string*) – The authorization key for .fi domains
- **Nameservers** (*list*) – [REQUIRED]

A list of new name servers for the domain.

Type: Complex

Children: Name , GlueIps

Required: Yes

– (dict) –

Nameserver includes the following elements.

* **Name** (string) – [REQUIRED]

The fully qualified host name of the name server.

Type: String

Constraint: Maximum 255 characters

Parent: Nameservers

* **GlueIps** (list) –

Glue IP address of a name server entry. Glue IP addresses are required only when the name of the name server is a subdomain of the domain. For example, if your domain is example.com and the name server for the domain is ns.example.com, you need to specify the IP address for ns.example.com.

Type: List of IP addresses.

Constraints: The list can contain only one IPv4 and one IPv6 address.

Parent: Nameservers

· (string) –

Return type dict

Returns

Response Syntax

```
{
    'OperationId': 'string'
}
```

Response Structure

• (dict) –

The UpdateDomainNameservers response includes the following element.

– **OperationId** (string) –

Identifier for tracking the progress of the request. To use this ID to query the operation status, use GetOperationDetail.

Type: String

Default: None

Constraints: Maximum 255 characters.

update_tags_for_domain (**kwargs)

This operation adds or updates tags for a specified domain.

All tag operations are eventually consistent; subsequent operations may not immediately represent all issued operations.

Request Syntax

```

response = client.update_tags_for_domain(
    DomainName='string',
    TagsToUpdate=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)

```

Parameters

- **DomainName** (*string*) – [REQUIRED]

The domain for which you want to add or update tags.

The name of a domain.

Type: String

Default: None

Constraints: The domain name can contain only the letters a through z, the numbers 0 through 9, and hyphen (-). Hyphens are allowed only when they are surrounded by letters, numbers, or other hyphens. You cannot specify a hyphen at the beginning or end of a label. To specify an Internationalized Domain Name, you must convert the name to Punycode.

Required: Yes

- **TagsToUpdate** (*list*) – A list of the tag keys and values that you want to add or update. If you specify a key that already exists, the corresponding value will be replaced.

Type: A complex type containing a list of tags

Default: None

Required: No

,

Each tag includes the following elements:

- **Key** The key (name) of a tag. Type: String Default: None Valid values: Unicode characters including alphanumeric, space, and ".:/=+-@" Constraints: Each key can be 1-128 characters long. Required: Yes
- **Value** The value of a tag. Type: String Default: None Valid values: Unicode characters including alphanumeric, space, and ".:/=+-@" Constraints: Each value can be 0-256 characters long. Required: Yes
- (*dict*) –

Each tag includes the following elements.

- * **Key** (*string*) –

The key (name) of a tag.

Type: String

Default: None

Valid values: A-Z, a-z, 0-9, space, ".:/=+-@"

Constraints: Each key can be 1-128 characters long.

Required: Yes

* **Value** (*string*) –

The value of a tag.

Type: String

Default: None

Valid values: A-Z, a-z, 0-9, space, ”./=+-@”

Constraints: Each value can be 0-256 characters long.

Required: Yes

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –

Paginators

The available paginators are:

- `Route53Domains.Paginator.ListDomains`
- `Route53Domains.Paginator.ListOperations`

class `Route53Domains.Paginator.ListDomains`

```
paginator = client.get_paginator('list_domains')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Route53Domains.Client.list_domains()`.

Request Syntax

```
response_iterator = paginator.paginate(
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Domains': [
    {
      'DomainName': 'string',
      'AutoRenew': True|False,
      'TransferLock': True|False,
      'Expiry': datetime(2015, 1, 1)
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

The `ListDomains` response includes the following elements.

- **Domains** (*list*) –

A summary of domains.

Type: Complex type containing a list of domain summaries.

Children: `AutoRenew`, `DomainName`, `Expiry`, `TransferLock`

- * (*dict*) –

- **DomainName** (*string*) –

The name of a domain.

Type: String

- **AutoRenew** (*boolean*) –

Indicates whether the domain is automatically renewed upon expiration.

Type: Boolean

Valid values: `True`|`False`

- **TransferLock** (*boolean*) –

Indicates whether a domain is locked from unauthorized transfer to another party.

Type: Boolean

Valid values: `True`|`False`

- **Expiry** (*datetime*) –

Expiration date of the domain in Coordinated Universal Time (UTC).

Type: Long

- **NextToken** (*string*) –

A token to resume pagination.

class Route53Domains.Paginator.**ListOperations**

```
paginator = client.get_paginator('list_operations')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Route53Domains.Client.list_operations()`.

Request Syntax

```
response_iterator = paginator.paginate(
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
    'Operations': [
        {
            'OperationId': 'string',
            'Status': 'SUBMITTED'|'IN_PROGRESS'|'ERROR'|'SUCCESSFUL'|'FAILED',
            'Type': 'REGISTER_DOMAIN'|'DELETE_DOMAIN'|'TRANSFER_IN_DOMAIN'|'UPDATE',
            'SubmittedDate': datetime(2015, 1, 1)
        },
    ],
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

The `ListOperations` response includes the following elements.

- **Operations** (*list*) –

Lists summaries of the operations.

Type: Complex type containing a list of operation summaries

Children: `OperationId`, `Status`, `SubmittedDate`, `Type`
 * (*dict*) –

`OperationSummary` includes the following elements.

- **OperationId** (*string*) –

Identifier returned to track the requested action.

Type: `String`

- **Status** (*string*) –

The current status of the requested operation in the system.

Type: `String`

- **Type** (*string*) –

Type of the action requested.

Type: `String`

Valid values: REGISTER_DOMAIN |
 DELETE_DOMAIN | TRANSFER_IN_DOMAIN |
 UPDATE_DOMAIN_CONTACT | UPDATE_NAMESERVER |
 CHANGE_PRIVACY_PROTECTION | DOMAIN_LOCK

- **SubmittedDate** (*datetime*) –

The date when the request was submitted.

- **NextToken** (*string*) –

A token to resume pagination.

S3

Table of Contents

- S3
 - Client
 - Paginators
 - Waiters
 - Service Resource
 - Bucket
 - BucketAcl
 - BucketCors
 - BucketLifecycle
 - BucketLogging
 - BucketNotification
 - BucketPolicy
 - BucketRequestPayment
 - BucketTagging
 - BucketVersioning
 - BucketWebsite
 - MultipartUpload
 - MultipartUploadPart
 - Object
 - ObjectAcl
 - ObjectSummary
 - ObjectVersion

Client

class S3.**Client**

A low-level client representing Amazon Simple Storage Service (S3):

```
import boto3
```

```
client = boto3.client('s3')
```

These are the available methods:

- `abort_multipart_upload()`
- `can_paginate()`
- `complete_multipart_upload()`
- `copy_object()`
- `create_bucket()`
- `create_multipart_upload()`
- `delete_bucket()`
- `delete_bucket_cors()`
- `delete_bucket_lifecycle()`
- `delete_bucket_policy()`
- `delete_bucket_replication()`
- `delete_bucket_tagging()`
- `delete_bucket_website()`
- `delete_object()`
- `delete_objects()`
- `download_file()`
- `generate_presigned_post()`
- `generate_presigned_url()`
- `get_bucket_acl()`
- `get_bucket_cors()`
- `get_bucket_lifecycle()`
- `get_bucket_lifecycle_configuration()`
- `get_bucket_location()`
- `get_bucket_logging()`
- `get_bucket_notification()`
- `get_bucket_notification_configuration()`
- `get_bucket_policy()`
- `get_bucket_replication()`
- `get_bucket_request_payment()`
- `get_bucket_tagging()`
- `get_bucket_versioning()`
- `get_bucket_website()`
- `get_object()`
- `get_object_acl()`
- `get_object_torrent()`
- `get_paginator()`
- `get_waiter()`
- `head_bucket()`
- `head_object()`
- `list_buckets()`
- `list_multipart_uploads()`
- `list_object_versions()`
- `list_objects()`
- `list_parts()`

- `put_bucket_acl()`
- `put_bucket_cors()`
- `put_bucket_lifecycle()`
- `put_bucket_lifecycle_configuration()`
- `put_bucket_logging()`
- `put_bucket_notification()`
- `put_bucket_notification_configuration()`
- `put_bucket_policy()`
- `put_bucket_replication()`
- `put_bucket_request_payment()`
- `put_bucket_tagging()`
- `put_bucket_versioning()`
- `put_bucket_website()`
- `put_object()`
- `put_object_acl()`
- `restore_object()`
- `upload_file()`
- `upload_part()`
- `upload_part_copy()`

abort_multipart_upload (***kwargs*)

Aborts a multipart upload.

To verify that all parts have been removed, so you don't get charged for the part storage, you should call the List Parts operation and ensure the parts list is empty.

Request Syntax

```
response = client.abort_multipart_upload(
    Bucket='string',
    Key='string',
    UploadId='string',
    RequestPayer='requester'
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **Key** (*string*) – [REQUIRED]
- **UploadId** (*string*) – [REQUIRED]
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
    'RequestCharged': 'requester'
}
```

Response Structure

- (*dict*) –
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

complete_multipart_upload (***kwargs*)

Completes a multipart upload by assembling previously uploaded parts.

Request Syntax

```
response = client.complete_multipart_upload(
    Bucket='string',
    Key='string',
    MultipartUpload={
        'Parts': [
            {
                'ETag': 'string',
                'PartNumber': 123
            },
        ]
    },
    UploadId='string',
    RequestPayer='requester'
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **Key** (*string*) – [REQUIRED]
- **MultipartUpload** (*dict*) –
 - **Parts** (*list*) –
 - * (*dict*) –
 - **ETag** (*string*) – Entity tag returned when the part was uploaded.
 - **PartNumber** (*integer*) – Part number that identifies the part. This is a positive integer between 1 and 10,000.
- **UploadId** (*string*) – [REQUIRED]
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns**Response Syntax**

```
{
    'Location': 'string',
    'Bucket': 'string',
    'Key': 'string',
    'Expiration': 'string',
    'ETag': 'string',
    'ServerSideEncryption': 'AES256'|'aws:kms',
```

```

    'VersionId': 'string',
    'SSEKMSKeyId': 'string',
    'RequestCharged': 'requester'
}

```

Response Structure

- (*dict*) –
 - **Location** (*string*) –
 - **Bucket** (*string*) –
 - **Key** (*string*) –
 - **Expiration** (*string*) – If the object expiration is configured, this will contain the expiration date (expiry-date) and rule ID (rule-id). The value of rule-id is URL encoded.
 - **ETag** (*string*) – Entity tag of the object.
 - **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
 - **VersionId** (*string*) – Version of the object.
 - **SSEKMSKeyId** (*string*) – If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

copy_object (**kwargs)

Creates a copy of an object that is already stored in Amazon S3.

Request Syntax

```

response = client.copy_object(
    ACL='private'|'public-read'|'public-read-write'|'authenticated-read'|'bucket-owner-read',
    Bucket='string',
    CacheControl='string',
    ContentDisposition='string',
    ContentEncoding='string',
    ContentLanguage='string',
    ContentType='string',
    CopySource='string',
    CopySourceIfMatch='string',
    CopySourceIfModifiedSince=datetime(2015, 1, 1),
    CopySourceIfNoneMatch='string',
    CopySourceIfUnmodifiedSince=datetime(2015, 1, 1),
    Expires=datetime(2015, 1, 1),
    GrantFullControl='string',
    GrantRead='string',
    GrantReadACP='string',
    GrantWriteACP='string',
    Key='string',
    Metadata={
        'string': 'string'
    },
    MetadataDirective='COPY'|'REPLACE',
    ServerSideEncryption='AES256'|'aws:kms',
    StorageClass='STANDARD'|'REDUCED_REDUNDANCY'|'STANDARD_IA',
    WebsiteRedirectLocation='string',
    SSECustomerAlgorithm='string',
    SSECustomerKey='string',
    SSEKMSKeyId='string',

```

```
CopySourceSSECustomerAlgorithm='string',
CopySourceSSECustomerKey='string',
CopySourceSSECustomerKeyMD5='string',
RequestPayer='requester'
)
```

Parameters

- **ACL** (*string*) – The canned ACL to apply to the object.
- **Bucket** (*string*) – [REQUIRED]
- **CacheControl** (*string*) – Specifies caching behavior along the request/reply chain.
- **ContentDisposition** (*string*) – Specifies presentational information for the object.
- **ContentEncoding** (*string*) – Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.
- **ContentLanguage** (*string*) – The language the content is in.
- **ContentType** (*string*) – A standard MIME type describing the format of the object data.
- **CopySource** (*string*) – [REQUIRED] The name of the source bucket and key name of the source object, separated by a slash (/). Must be URL-encoded.
- **CopySourceIfMatch** (*string*) – Copies the object if its entity tag (ETag) matches the specified tag.
- **CopySourceIfModifiedSince** (*datetime*) – Copies the object if it has been modified since the specified time.
- **CopySourceIfNoneMatch** (*string*) – Copies the object if its entity tag (ETag) is different than the specified ETag.
- **CopySourceIfUnmodifiedSince** (*datetime*) – Copies the object if it hasn't been modified since the specified time.
- **Expires** (*datetime*) – The date and time at which the object is no longer cacheable.
- **GrantFullControl** (*string*) – Gives the grantee READ, READ_ACP, and WRITE_ACP permissions on the object.
- **GrantRead** (*string*) – Allows grantee to read the object data and its metadata.
- **GrantReadACP** (*string*) – Allows grantee to read the object ACL.
- **GrantWriteACP** (*string*) – Allows grantee to write the ACL for the applicable object.
- **Key** (*string*) – [REQUIRED]
- **Metadata** (*dict*) – A map of metadata to store with the object in S3.
 - (*string*) –
 - * (*string*) –
- **MetadataDirective** (*string*) – Specifies whether the metadata is copied from the source object or replaced with metadata provided in the request.
- **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
- **StorageClass** (*string*) – The type of storage to use for the object. Defaults to 'STANDARD'.
- **WebsiteRedirectLocation** (*string*) – If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object

and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the `x-amz-server-side-encryption-customer-algorithm` header.

- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **SSEKMSKeyId** (*string*) – Specifies the AWS KMS key ID to use for object encryption. All GET and PUT requests for an object protected by AWS KMS will fail if not made via SSL or using SigV4. Documentation on configuring any of the officially supported AWS SDKs and CLI can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingAWSSDK.html#specify-signature-version>
- **CopySourceSSECustomerAlgorithm** (*string*) – Specifies the algorithm to use when decrypting the source object (e.g., AES256).
- **CopySourceSSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use to decrypt the source object. The encryption key provided in this header must be one that was used when the source object was created.
- **CopySourceSSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error.
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
  'CopyObjectResult': {
    'ETag': 'string',
    'LastModified': datetime(2015, 1, 1)
  },
  'Expiration': 'string',
  'CopySourceVersionId': 'string',
  'ServerSideEncryption': 'AES256'|'aws:kms',
  'SSECustomerAlgorithm': 'string',
  'SSECustomerKeyMD5': 'string',
  'SSEKMSKeyId': 'string',
  'RequestCharged': 'requester'
}
```

Response Structure

- (*dict*) –
 - **CopyObjectResult** (*dict*) –
 - * **ETag** (*string*) –
 - * **LastModified** (*datetime*) –
 - **Expiration** (*string*) – If the object expiration is configured, the response includes this header.
 - **CopySourceVersionId** (*string*) –

- **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
- **SSECustomerAlgorithm** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.
- **SSECustomerKeyMD5** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.
- **SSEKMSKeyId** (*string*) – If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.
- **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

```
create_bucket (**kwargs)
```

Creates a new bucket.

Request Syntax

```
response = client.create_bucket(
    ACL='private'|'public-read'|'public-read-write'|'authenticated-read',
    Bucket='string',
    CreateBucketConfiguration={
        'LocationConstraint': 'EU'|'eu-west-1'|'us-west-1'|'us-west-2'|'ap-southeast-1'|'ap
    },
    GrantFullControl='string',
    GrantRead='string',
    GrantReadACP='string',
    GrantWrite='string',
    GrantWriteACP='string'
)
```

Parameters

- **ACL** (*string*) – The canned ACL to apply to the bucket.
- **Bucket** (*string*) – [REQUIRED]
- **CreateBucketConfiguration** (*dict*) –
 - **LocationConstraint** (*string*) – Specifies the region where the bucket will be created. If you don't specify a region, the bucket will be created in US Standard.
- **GrantFullControl** (*string*) – Allows grantee the read, write, read ACP, and write ACP permissions on the bucket.
- **GrantRead** (*string*) – Allows grantee to list the objects in the bucket.
- **GrantReadACP** (*string*) – Allows grantee to read the bucket ACL.
- **GrantWrite** (*string*) – Allows grantee to create, overwrite, and delete any object in the bucket.
- **GrantWriteACP** (*string*) – Allows grantee to write the ACL for the applicable bucket.

Return type dict

Returns

Response Syntax

```
{
  'Location': 'string'
}
```


Response Structure

- (*dict*) –
 - **Location** (*string*) –

create_multipart_upload (**kwargs)

Initiates a multipart upload and returns an upload ID.

Note: After you initiate multipart upload and upload one or more parts, you must either complete or abort multipart upload in order to stop getting charged for storage of the uploaded parts. Only after you either complete or abort multipart upload, Amazon S3 frees up the parts storage and stops charging you for the parts storage.

Request Syntax

```
response = client.create_multipart_upload(
    ACL='private'|'public-read'|'public-read-write'|'authenticated-read'|'bucket-owner-read',
    Bucket='string',
    CacheControl='string',
    ContentDisposition='string',
    ContentEncoding='string',
    ContentLanguage='string',
    ContentType='string',
    Expires=datetime(2015, 1, 1),
    GrantFullControl='string',
    GrantRead='string',
    GrantReadACP='string',
    GrantWriteACP='string',
    Key='string',
    Metadata={
        'string': 'string'
    },
    ServerSideEncryption='AES256'|'aws:kms',
    StorageClass='STANDARD'|'REDUCED_REDUNDANCY'|'STANDARD_IA',
    WebsiteRedirectLocation='string',
    SSECustomerAlgorithm='string',
    SSECustomerKey='string',
    SSEKMSKeyId='string',
    RequestPayer='requester'
)
```

Parameters

- **ACL** (*string*) – The canned ACL to apply to the object.
- **Bucket** (*string*) – **[REQUIRED]**
- **CacheControl** (*string*) – Specifies caching behavior along the request/reply chain.
- **ContentDisposition** (*string*) – Specifies presentational information for the object.
- **ContentEncoding** (*string*) – Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.
- **ContentLanguage** (*string*) – The language the content is in.
- **ContentType** (*string*) – A standard MIME type describing the format of the object data.
- **Expires** (*datetime*) – The date and time at which the object is no longer cacheable.
- **GrantFullControl** (*string*) – Gives the grantee READ, READ_ACP, and WRITE_ACP permissions on the object.
- **GrantRead** (*string*) – Allows grantee to read the object data and its metadata.

- **GrantReadACP** (*string*) – Allows grantee to read the object ACL.
- **GrantWriteACP** (*string*) – Allows grantee to write the ACL for the applicable object.
- **Key** (*string*) – **[REQUIRED]**
- **Metadata** (*dict*) – A map of metadata to store with the object in S3.
 - (*string*) –
 - * (*string*) –
- **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
- **StorageClass** (*string*) – The type of storage to use for the object. Defaults to 'STANDARD'.
- **WebsiteRedirectLocation** (*string*) – If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.
- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **SSEKMSKeyId** (*string*) – Specifies the AWS KMS key ID to use for object encryption. All GET and PUT requests for an object protected by AWS KMS will fail if not made via SSL or using SigV4. Documentation on configuring any of the officially supported AWS SDKs and CLI can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingAWSSDK.html#specify-signature-version>
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
    'Bucket': 'string',
    'Key': 'string',
    'UploadId': 'string',
    'ServerSideEncryption': 'AES256'|'aws:kms',
    'SSECustomerAlgorithm': 'string',
    'SSECustomerKeyMD5': 'string',
    'SSEKMSKeyId': 'string',
    'RequestCharged': 'requester'
}
```

Response Structure

- (*dict*) –

- **Bucket** (*string*) – Name of the bucket to which the multipart upload was initiated.
- **Key** (*string*) – Object key for which the multipart upload was initiated.
- **UploadId** (*string*) – ID for the initiated multipart upload.
- **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
- **SSECustomerAlgorithm** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.
- **SSECustomerKeyMD5** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.
- **SSEKMSKeyId** (*string*) – If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.
- **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

delete_bucket (***kwargs*)

Deletes the bucket. All objects (including all object versions and Delete Markers) in the bucket must be deleted before the bucket itself can be deleted.

Request Syntax

```
response = client.delete_bucket(
    Bucket='string'
)
```

Parameters **Bucket** (*string*) – [REQUIRED]

Returns None

delete_bucket_cors (***kwargs*)

Deletes the cors configuration information set for the bucket.

Request Syntax

```
response = client.delete_bucket_cors(
    Bucket='string'
)
```

Parameters **Bucket** (*string*) – [REQUIRED]

Returns None

delete_bucket_lifecycle (***kwargs*)

Deletes the lifecycle configuration from the bucket.

Request Syntax

```
response = client.delete_bucket_lifecycle(
    Bucket='string'
)
```

Parameters **Bucket** (*string*) – [REQUIRED]

Returns None

delete_bucket_policy (***kwargs*)

Deletes the policy from the bucket.

Request Syntax

```
response = client.delete_bucket_policy(
    Bucket='string'
)
```

Parameters **Bucket** (*string*) – [REQUIRED]**Returns** None**delete_bucket_replication** (**kwargs)**Request Syntax**

```
response = client.delete_bucket_replication(
    Bucket='string'
)
```

Parameters **Bucket** (*string*) – [REQUIRED]**Returns** None**delete_bucket_tagging** (**kwargs)

Deletes the tags from the bucket.

Request Syntax

```
response = client.delete_bucket_tagging(
    Bucket='string'
)
```

Parameters **Bucket** (*string*) – [REQUIRED]**Returns** None**delete_bucket_website** (**kwargs)

This operation removes the website configuration from the bucket.

Request Syntax

```
response = client.delete_bucket_website(
    Bucket='string'
)
```

Parameters **Bucket** (*string*) – [REQUIRED]**Returns** None**delete_object** (**kwargs)

Removes the null version (if there is one) of an object and inserts a delete marker, which becomes the latest version of the object. If there isn't a null version, Amazon S3 does not remove any objects.

Request Syntax

```
response = client.delete_object(
    Bucket='string',
    Key='string',
    MFA='string',
    VersionId='string',
    RequestPayer='requester'
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]

- **Key** (*string*) – [REQUIRED]
- **MFA** (*string*) – The concatenation of the authentication device’s serial number, a space, and the value that is displayed on your authentication device.
- **VersionId** (*string*) – VersionId used to reference a specific version of the object.
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
    'DeleteMarker': True|False,
    'VersionId': 'string',
    'RequestCharged': 'requester'
}
```

Response Structure

- (*dict*) –
 - **DeleteMarker** (*boolean*) – Specifies whether the versioned object that was permanently deleted was (true) or was not (false) a delete marker.
 - **VersionId** (*string*) – Returns the version ID of the delete marker created as a result of the DELETE operation.
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

`delete_objects` (**kwargs)

This operation enables you to delete multiple objects from a bucket using a single HTTP request. You may specify up to 1000 keys.

Request Syntax

```
response = client.delete_objects(
    Bucket='string',
    Delete={
        'Objects': [
            {
                'Key': 'string',
                'VersionId': 'string'
            },
        ],
        'Quiet': True|False
    },
    MFA='string',
    RequestPayer='requester'
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **Delete** (*dict*) – [REQUIRED]
 - **Objects** (*list*) – [REQUIRED]
 - * (*dict*) –
 - **Key** (*string*) – [REQUIRED] Key name of the object to delete.

- **VersionId** (*string*) – VersionId for the specific version of the object to delete.
- **Quiet** (*boolean*) – Element to enable quiet mode for the request. When you add this element, you must set its value to true.
- **MFA** (*string*) – The concatenation of the authentication device’s serial number, a space, and the value that is displayed on your authentication device.
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
  'Deleted': [
    {
      'Key': 'string',
      'VersionId': 'string',
      'DeleteMarker': True|False,
      'DeleteMarkerVersionId': 'string'
    },
  ],
  'RequestCharged': 'requester',
  'Errors': [
    {
      'Key': 'string',
      'VersionId': 'string',
      'Code': 'string',
      'Message': 'string'
    },
  ],
}
```

Response Structure

- (*dict*) –
 - **Deleted** (*list*) –
 - * (*dict*) –
 - **Key** (*string*) –
 - **VersionId** (*string*) –
 - **DeleteMarker** (*boolean*) –
 - **DeleteMarkerVersionId** (*string*) –
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.
 - **Errors** (*list*) –
 - * (*dict*) –
 - **Key** (*string*) –
 - **VersionId** (*string*) –
 - **Code** (*string*) –
 - **Message** (*string*) –

download_file (*Bucket, Key, Filename, ExtraArgs=None, Callback=None, Config=None*)

Download an S3 object to a file.

Usage:

```
import boto3
s3 = boto3.resource('s3')
s3.meta.client.download_file('mybucket', 'hello.txt', '/tmp/hello.txt')
```

Similar behavior as `S3Transfer`'s `download_file()` method, except that parameters are capitalized. Detailed examples can be found at [S3Transfer's Usage](#).

generate_presigned_post (*Bucket, Key, Fields=None, Conditions=None, ExpiresIn=3600*)

Builds the url and the form fields used for a presigned s3 post

Parameters

- **Bucket** (*string*) – The name of the bucket to presign the post to. Note that bucket related conditions should not be included in the `conditions` parameter.
- **Key** (*string*) – Key name, optionally add `${filename}` to the end to attach the submitted filename. Note that key related conditions and fields are filled out for you and should not be included in the `fields` or `conditions` parameter.
- **Fields** (*dict*) – A dictionary of prefilled form fields to build on top of. Elements that may be included are `acl`, `Cache-Control`, `Content-Type`, `Content-Disposition`, `Content-Encoding`, `Expires`, `success_action_redirect`, `redirect`, `success_action_status`, and `x-amz-meta-`.

Note that if a particular element is included in the `fields` dictionary it will not be automatically added to the `conditions` list. You must specify a condition for the element as well.

- **Conditions** (*list*) – A list of conditions to include in the policy. Each element can be either a list or a structure. For example:

```
[ {"acl": "public-read"}, ["content-length-range", 2, 5], ["starts-with", "$success_action_redirect", ""] ]
```

Conditions that are included may pertain to `acl`, `content-length-range`, `Cache-Control`, `Content-Type`, `Content-Disposition`, `Content-Encoding`, `Expires`, `success_action_redirect`, `redirect`, `success_action_status`, and/or `x-amz-meta-`.

Note that if you include a condition, you must specify the a valid value in the `fields` dictionary as well. A value will not be added automatically to the `fields` dictionary based on the conditions.

- **ExpiresIn** (*int*) – The number of seconds the presigned post is valid for.

Return type dict

Returns

A dictionary with two elements: `url` and `fields`. `url` is the url to post to. `fields` is a dictionary filled with the form fields and respective values to use when submitting the post. For example:

```
{'url': 'https://mybucket.s3.amazonaws.com',
 'fields': {'acl': 'public-read', 'key': 'mykey', 'signature': 'mysignature',
            'policy': 'mybase64 encoded policy'}}
```

generate_presigned_url (*ClientMethod, Params=None, ExpiresIn=3600, HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default,

the http method is whatever is used in the method's model.

Returns The presigned url

get_bucket_acl (**kwargs)

Gets the access control policy for the bucket.

Request Syntax

```
response = client.get_bucket_acl(
    Bucket='string'
)
```

Parameters **Bucket** (*string*) – [REQUIRED]

Return type dict

Returns

Response Syntax

```
{
  'Owner': {
    'DisplayName': 'string',
    'ID': 'string'
  },
  'Grants': [
    {
      'Grantee': {
        'DisplayName': 'string',
        'EmailAddress': 'string',
        'ID': 'string',
        'Type': 'CanonicalUser'|'AmazonCustomerByEmail'|'Group',
        'URI': 'string'
      },
      'Permission': 'FULL_CONTROL'|'WRITE'|'WRITE_ACP'|'READ'|'READ_ACP'
    },
    ...
  ]
}
```

Response Structure

- (*dict*) –
 - **Owner** (*dict*) –
 - * **DisplayName** (*string*) –
 - * **ID** (*string*) –
 - **Grants** (*list*) – A list of grants.
 - * (*dict*) –
 - **Grantee** (*dict*) –
 - **DisplayName** (*string*) – Screen name of the grantee.
 - **EmailAddress** (*string*) – Email address of the grantee.
 - **ID** (*string*) – The canonical user ID of the grantee.
 - **Type** (*string*) – Type of grantee
 - **URI** (*string*) – URI of the grantee group.
 - **Permission** (*string*) – Specifies the permission given to the grantee.

get_bucket_cors (**kwargs)

Returns the cors configuration for the bucket.

Request Syntax


```
response = client.get_bucket_cors(
    Bucket='string'
)
```

Parameters **Bucket** (*string*) – [REQUIRED]

Return type dict

Returns

Response Syntax

```
{
  'CORSRules': [
    {
      'AllowedHeaders': [
        'string',
      ],
      'AllowedMethods': [
        'string',
      ],
      'AllowedOrigins': [
        'string',
      ],
      'ExposeHeaders': [
        'string',
      ],
      'MaxAgeSeconds': 123
    },
  ]
}
```

Response Structure

- (*dict*) –
 - **CORSRules** (*list*) –
 - * (*dict*) –
 - **AllowedHeaders** (*list*) – Specifies which headers are allowed in a pre-flight OPTIONS request.
 - (*string*) –
 - **AllowedMethods** (*list*) – Identifies HTTP methods that the domain/origin specified in the rule is allowed to execute.
 - (*string*) –
 - **AllowedOrigins** (*list*) – One or more origins you want customers to be able to access the bucket from.
 - (*string*) –
 - **ExposeHeaders** (*list*) – One or more headers in the response that you want customers to be able to access from their applications (for example, from a JavaScript XMLHttpRequest object).
 - (*string*) –
 - **MaxAgeSeconds** (*integer*) – The time in seconds that your browser is to cache the preflight response for the specified resource.

get_bucket_lifecycle (***kwargs*)

Deprecated, see the GetBucketLifecycleConfiguration operation.

Request Syntax

```
response = client.get_bucket_lifecycle(  
    Bucket='string'  
)
```

Parameters **Bucket** (*string*) – [REQUIRED]

Return type dict

Returns

Response Syntax

```
{  
    'Rules': [  
        {  
            'Expiration': {  
                'Date': datetime(2015, 1, 1),  
                'Days': 123  
            },  
            'ID': 'string',  
            'Prefix': 'string',  
            'Status': 'Enabled'|'Disabled',  
            'Transition': {  
                'Date': datetime(2015, 1, 1),  
                'Days': 123,  
                'StorageClass': 'GLACIER'|'STANDARD_IA'  
            },  
            'NoncurrentVersionTransition': {  
                'NoncurrentDays': 123,  
                'StorageClass': 'GLACIER'|'STANDARD_IA'  
            },  
            'NoncurrentVersionExpiration': {  
                'NoncurrentDays': 123  
            }  
        },  
    ],  
}
```

Response Structure

- (*dict*) –
 - **Rules** (*list*) –
 - * (*dict*) –
 - **Expiration** (*dict*) –
 - **Date** (*datetime*) – Indicates at what date the object is to be moved or deleted. Should be in GMT ISO 8601 Format.
 - **Days** (*integer*) – Indicates the lifetime, in days, of the objects that are subject to the rule. The value must be a non-zero positive integer.
 - **ID** (*string*) – Unique identifier for the rule. The value cannot be longer than 255 characters.
 - **Prefix** (*string*) – Prefix identifying one or more objects to which the rule applies.
 - **Status** (*string*) – If ‘Enabled’, the rule is currently being applied. If ‘Disabled’, the rule is not currently being applied.
 - **Transition** (*dict*) –
 - **Date** (*datetime*) – Indicates at what date the object is to be moved or deleted. Should be in GMT ISO 8601 Format.
 - **Days** (*integer*) – Indicates the lifetime, in days, of the objects

that are subject to the rule. The value must be a non-zero positive integer.

- **StorageClass** (*string*) – The class of storage used to store the object.
- **NoncurrentVersionTransition** (*dict*) – Container for the transition rule that describes when noncurrent objects transition to the STANDARD_IA or GLACIER storage class. If your bucket is versioning-enabled (or versioning is suspended), you can set this action to request that Amazon S3 transition noncurrent object versions to the STANDARD_IA or GLACIER storage class at a specific period in the object's lifetime.
- **NoncurrentDays** (*integer*) – Specifies the number of days an object is noncurrent before Amazon S3 can perform the associated action. For information about the noncurrent days calculations, see *How Amazon S3 Calculates When an Object Became Noncurrent* in the Amazon Simple Storage Service Developer Guide.
- **StorageClass** (*string*) – The class of storage used to store the object.
- **NoncurrentVersionExpiration** (*dict*) – Specifies when noncurrent object versions expire. Upon expiration, Amazon S3 permanently deletes the noncurrent object versions. You set this lifecycle configuration action on a bucket that has versioning enabled (or suspended) to request that Amazon S3 delete noncurrent object versions at a specific period in the object's lifetime.
- **NoncurrentDays** (*integer*) – Specifies the number of days an object is noncurrent before Amazon S3 can perform the associated action. For information about the noncurrent days calculations, see *How Amazon S3 Calculates When an Object Became Noncurrent* in the Amazon Simple Storage Service Developer Guide.

get_bucket_lifecycle_configuration (**kwargs)

Returns the lifecycle configuration information set on the bucket.

Request Syntax

```
response = client.get_bucket_lifecycle_configuration(
    Bucket='string'
)
```

Parameters **Bucket** (*string*) – [REQUIRED]

Return type dict

Returns

Response Syntax

```
{
    'Rules': [
        {
            'Expiration': {
                'Date': datetime(2015, 1, 1),
                'Days': 123
            },
```

```
'ID': 'string',
'Prefix': 'string',
'Status': 'Enabled'|'Disabled',
'Transitions': [
    {
        'Date': datetime(2015, 1, 1),
        'Days': 123,
        'StorageClass': 'GLACIER'|'STANDARD_IA'
    },
],
'NoncurrentVersionTransitions': [
    {
        'NoncurrentDays': 123,
        'StorageClass': 'GLACIER'|'STANDARD_IA'
    },
],
'NoncurrentVersionExpiration': {
    'NoncurrentDays': 123
}
},
]
```

Response Structure

- *(dict)* –
 - **Rules** (*list*) –
 - * *(dict)* –
 - **Expiration** (*dict*) –
 - **Date** (*datetime*) – Indicates at what date the object is to be moved or deleted. Should be in GMT ISO 8601 Format.
 - **Days** (*integer*) – Indicates the lifetime, in days, of the objects that are subject to the rule. The value must be a non-zero positive integer.
 - **ID** (*string*) – Unique identifier for the rule. The value cannot be longer than 255 characters.
 - **Prefix** (*string*) – Prefix identifying one or more objects to which the rule applies.
 - **Status** (*string*) – If ‘Enabled’, the rule is currently being applied. If ‘Disabled’, the rule is not currently being applied.
 - **Transitions** (*list*) –
 - *(dict)* –
 - **Date** (*datetime*) – Indicates at what date the object is to be moved or deleted. Should be in GMT ISO 8601 Format.
 - **Days** (*integer*) – Indicates the lifetime, in days, of the objects that are subject to the rule. The value must be a non-zero positive integer.
 - **StorageClass** (*string*) – The class of storage used to store the object.
 - **NoncurrentVersionTransitions** (*list*) –
 - *(dict)* – Container for the transition rule that describes when noncurrent objects transition to the STANDARD_IA or GLACIER storage class. If your bucket is versioning-enabled (or versioning is suspended), you can set this action to request that Amazon S3 transition noncurrent object versions to the STANDARD_IA or GLACIER storage class at a specific

period in the object's lifetime.

- **NoncurrentDays** (*integer*) – Specifies the number of days an object is noncurrent before Amazon S3 can perform the associated action. For information about the noncurrent days calculations, see *How Amazon S3 Calculates When an Object Became Noncurrent* in the Amazon Simple Storage Service Developer Guide.
- **StorageClass** (*string*) – The class of storage used to store the object.
- **NoncurrentVersionExpiration** (*dict*) – Specifies when non-current object versions expire. Upon expiration, Amazon S3 permanently deletes the noncurrent object versions. You set this lifecycle configuration action on a bucket that has versioning enabled (or suspended) to request that Amazon S3 delete noncurrent object versions at a specific period in the object's lifetime.
- **NoncurrentDays** (*integer*) – Specifies the number of days an object is noncurrent before Amazon S3 can perform the associated action. For information about the noncurrent days calculations, see *How Amazon S3 Calculates When an Object Became Noncurrent* in the Amazon Simple Storage Service Developer Guide.

get_bucket_location (***kwargs*)

Returns the region the bucket resides in.

Request Syntax

```
response = client.get_bucket_location(
    Bucket='string'
)
```

Parameters **Bucket** (*string*) – [REQUIRED]

Return type dict

Returns

Response Syntax

```
{
    'LocationConstraint': 'EU'|'eu-west-1'|'us-west-1'|'us-west-2'|'ap-southeast-1'
}
```

Response Structure

- (*dict*) –
 - **LocationConstraint** (*string*) –

get_bucket_logging (***kwargs*)

Returns the logging status of a bucket and the permissions users have to view and modify that status. To use GET, you must be the bucket owner.

Request Syntax

```
response = client.get_bucket_logging(
    Bucket='string'
)
```

Parameters **Bucket** (*string*) – [REQUIRED]

Return type dict

Returns

Response Syntax

```
{
    'LoggingEnabled': {
        'TargetBucket': 'string',
        'TargetGrants': [
            {
                'Grantee': {
                    'DisplayName': 'string',
                    'EmailAddress': 'string',
                    'ID': 'string',
                    'Type': 'CanonicalUser'|'AmazonCustomerByEmail'|'Group',
                    'URI': 'string'
                },
                'Permission': 'FULL_CONTROL'|'READ'|'WRITE'
            },
        ],
        'TargetPrefix': 'string'
    }
}
```

Response Structure

- (dict) –
 - **LoggingEnabled** (dict) –
 - * **TargetBucket** (string) – Specifies the bucket where you want Amazon S3 to store server access logs. You can have your logs delivered to any bucket that you own, including the same bucket that is being logged. You can also configure multiple buckets to deliver their logs to the same target bucket. In this case you should choose a different TargetPrefix for each source bucket so that the delivered log files can be distinguished by key.
 - * **TargetGrants** (list) –
 - (dict) –
 - **Grantee** (dict) –
 - **DisplayName** (string) – Screen name of the grantee.
 - **EmailAddress** (string) – Email address of the grantee.
 - **ID** (string) – The canonical user ID of the grantee.
 - **Type** (string) – Type of grantee
 - **URI** (string) – URI of the grantee group.
 - **Permission** (string) – Logging permissions assigned to the Grantee for the bucket.
 - * **TargetPrefix** (string) – This element lets you specify a prefix for the keys that the log files will be stored under.

get_bucket_notification (**kwargs)

Deprecated, see the GetBucketNotificationConfiguration operation.

Request Syntax

```
response = client.get_bucket_notification(
    Bucket='string'
)
```

Parameters **Bucket** (*string*) – [REQUIRED] Name of the bucket to get the notification configuration for.

Return type dict

Returns

Response Syntax

```
{
  'TopicConfiguration': {
    'Id': 'string',
    'Events': [
      's3:ReducedRedundancyLostObject'|'s3:ObjectCreated:*'|'s3:ObjectCreate
    ],
    'Event': 's3:ReducedRedundancyLostObject'|'s3:ObjectCreated:*'|'s3:ObjectC
    'Topic': 'string'
  },
  'QueueConfiguration': {
    'Id': 'string',
    'Event': 's3:ReducedRedundancyLostObject'|'s3:ObjectCreated:*'|'s3:ObjectC
    'Events': [
      's3:ReducedRedundancyLostObject'|'s3:ObjectCreated:*'|'s3:ObjectCreate
    ],
    'Queue': 'string'
  },
  'CloudFunctionConfiguration': {
    'Id': 'string',
    'Event': 's3:ReducedRedundancyLostObject'|'s3:ObjectCreated:*'|'s3:ObjectC
    'Events': [
      's3:ReducedRedundancyLostObject'|'s3:ObjectCreated:*'|'s3:ObjectCreate
    ],
    'CloudFunction': 'string',
    'InvocationRole': 'string'
  }
}
```

Response Structure

- (*dict*) –
 - **TopicConfiguration** (*dict*) –
 - * **Id** (*string*) – Optional unique identifier for configurations in a notification configuration. If you don’t provide one, Amazon S3 will assign an ID.
 - * **Events** (*list*) –
 - (*string*) – Bucket event for which to send notifications.
 - * **Event** (*string*) – Bucket event for which to send notifications.
 - * **Topic** (*string*) – Amazon SNS topic to which Amazon S3 will publish a message to report the specified events for the bucket.
 - **QueueConfiguration** (*dict*) –
 - * **Id** (*string*) – Optional unique identifier for configurations in a notification configuration. If you don’t provide one, Amazon S3 will assign an ID.
 - * **Event** (*string*) – Bucket event for which to send notifications.
 - * **Events** (*list*) –
 - (*string*) – Bucket event for which to send notifications.
 - * **Queue** (*string*) –
 - **CloudFunctionConfiguration** (*dict*) –
 - * **Id** (*string*) – Optional unique identifier for configurations in a notification configuration. If you don’t provide one, Amazon S3 will

- assign an ID.
- * **Event** (*string*) – Bucket event for which to send notifications.
- * **Events** (*list*) –
 - (*string*) – Bucket event for which to send notifications.
- * **CloudFunction** (*string*) –
- * **InvocationRole** (*string*) –

get_bucket_notification_configuration (**kwargs)

Returns the notification configuration of a bucket.

Request Syntax

```
response = client.get_bucket_notification_configuration(
    Bucket='string'
)
```

Parameters **Bucket** (*string*) – [REQUIRED] Name of the bucket to get the notification configuration for.

Return type dict

Returns

Response Syntax

```
{
    'TopicConfigurations': [
        {
            'Id': 'string',
            'TopicArn': 'string',
            'Events': [
                's3:ReducedRedundancyLostObject'|'s3:ObjectCreated:*'|'s3:ObjectCr
            ],
            'Filter': {
                'Key': {
                    'FilterRules': [
                        {
                            'Name': 'prefix'|'suffix',
                            'Value': 'string'
                        },
                    ],
                },
            },
        },
    ],
    'QueueConfigurations': [
        {
            'Id': 'string',
            'QueueArn': 'string',
            'Events': [
                's3:ReducedRedundancyLostObject'|'s3:ObjectCreated:*'|'s3:ObjectCr
            ],
            'Filter': {
                'Key': {
                    'FilterRules': [
                        {
                            'Name': 'prefix'|'suffix',
                            'Value': 'string'
                        },
                    ],
                },
            },
        },
    ],
}
```



```

    }
  },
],
'LambdaFunctionConfigurations': [
  {
    'Id': 'string',
    'LambdaFunctionArn': 'string',
    'Events': [
      's3:ReducedRedundancyLostObject'|'s3:ObjectCreated:*'|'s3:ObjectCr
    ],
    'Filter': {
      'Key': {
        'FilterRules': [
          {
            'Name': 'prefix'|'suffix',
            'Value': 'string'
          },
        ],
      },
    },
  },
]
}

```

Response Structure

- (*dict*) – Container for specifying the notification configuration of the bucket. If this element is empty, notifications are turned off on the bucket.
 - **TopicConfigurations** (*list*) –
 - * (*dict*) – Container for specifying the configuration when you want Amazon S3 to publish events to an Amazon Simple Notification Service (Amazon SNS) topic.
 - **Id** (*string*) – Optional unique identifier for configurations in a notification configuration. If you don't provide one, Amazon S3 will assign an ID.
 - **TopicArn** (*string*) – Amazon SNS topic ARN to which Amazon S3 will publish a message when it detects events of specified type.
 - **Events** (*list*) –
 - (*string*) – Bucket event for which to send notifications.
 - **Filter** (*dict*) – Container for object key name filtering rules. For information about key name filtering, go to [Configuring Event Notifications](#) in the Amazon Simple Storage Service Developer Guide.
 - **Key** (*dict*) – Container for object key name prefix and suffix filtering rules.
 - **FilterRules** (*list*) – A list of containers for key value pair that defines the criteria for the filter rule.
 - (*dict*) – Container for key value pair that defines the criteria for the filter rule.
 - **Name** (*string*) – Object key name prefix or suffix identifying one or more objects to which the filtering rule applies. Maximum prefix length can be up to 1,024 characters. Overlapping prefixes and suffixes are not supported. For more information, go to [Configuring Event Notifications](#) in the Amazon Simple Storage Service Developer Guide.

- **Value** (*string*) –
- **QueueConfigurations** (*list*) –
 - * (*dict*) – Container for specifying an configuration when you want Amazon S3 to publish events to an Amazon Simple Queue Service (Amazon SQS) queue.
 - **Id** (*string*) – Optional unique identifier for configurations in a notification configuration. If you don't provide one, Amazon S3 will assign an ID.
 - **QueueArn** (*string*) – Amazon SQS queue ARN to which Amazon S3 will publish a message when it detects events of specified type.
 - **Events** (*list*) –
 - (*string*) – Bucket event for which to send notifications.
 - **Filter** (*dict*) – Container for object key name filtering rules. For information about key name filtering, go to [Configuring Event Notifications](#) in the Amazon Simple Storage Service Developer Guide.
 - **Key** (*dict*) – Container for object key name prefix and suffix filtering rules.
 - **FilterRules** (*list*) – A list of containers for key value pair that defines the criteria for the filter rule.
 - (*dict*) – Container for key value pair that defines the criteria for the filter rule.
 - **Name** (*string*) – Object key name prefix or suffix identifying one or more objects to which the filtering rule applies. Maximum prefix length can be up to 1,024 characters. Overlapping prefixes and suffixes are not supported. For more information, go to [Configuring Event Notifications](#) in the Amazon Simple Storage Service Developer Guide.
 - **Value** (*string*) –
- **LambdaFunctionConfigurations** (*list*) –
 - * (*dict*) – Container for specifying the AWS Lambda notification configuration.
 - **Id** (*string*) – Optional unique identifier for configurations in a notification configuration. If you don't provide one, Amazon S3 will assign an ID.
 - **LambdaFunctionArn** (*string*) – Lambda cloud function ARN that Amazon S3 can invoke when it detects events of the specified type.
 - **Events** (*list*) –
 - (*string*) – Bucket event for which to send notifications.
 - **Filter** (*dict*) – Container for object key name filtering rules. For information about key name filtering, go to [Configuring Event Notifications](#) in the Amazon Simple Storage Service Developer Guide.
 - **Key** (*dict*) – Container for object key name prefix and suffix filtering rules.
 - **FilterRules** (*list*) – A list of containers for key value pair that defines the criteria for the filter rule.
 - (*dict*) – Container for key value pair that defines the criteria for the filter rule.
 - **Name** (*string*) – Object key name prefix or suffix identifying one or more objects to which the filtering rule applies. Maximum prefix length can be up to 1,024 characters. Overlapping

prefixes and suffixes are not supported. For more information, go to [Configuring Event Notifications](#) in the Amazon Simple Storage Service Developer Guide.

- **Value** (*string*) –

get_bucket_policy (***kwargs*)

Returns the policy of a specified bucket.

Request Syntax

```
response = client.get_bucket_policy(
    Bucket='string'
)
```

Parameters **Bucket** (*string*) – **[REQUIRED]**

Return type dict

Returns

Response Syntax

```
{
    'Policy': 'string'
}
```

Response Structure

- (*dict*) –
 - **Policy** (*string*) – The bucket policy as a JSON document.

get_bucket_replication (***kwargs*)

Request Syntax

```
response = client.get_bucket_replication(
    Bucket='string'
)
```

Parameters **Bucket** (*string*) – **[REQUIRED]**

Return type dict

Returns

Response Syntax

```
{
    'ReplicationConfiguration': {
        'Role': 'string',
        'Rules': [
            {
                'ID': 'string',
                'Prefix': 'string',
                'Status': 'Enabled'|'Disabled',
                'Destination': {
                    'Bucket': 'string',
                    'StorageClass': 'STANDARD'|'REDUCED_REDUNDANCY'|'STANDARD_IA'
                }
            },
        ]
    }
}
```

Response Structure

- (*dict*) –
 - **ReplicationConfiguration** (*dict*) – Container for replication rules. You can add as many as 1,000 rules. Total replication configuration size can be up to 2 MB.
 - * **Role** (*string*) – Amazon Resource Name (ARN) of an IAM role for Amazon S3 to assume when replicating the objects.
 - * **Rules** (*list*) – Container for information about a particular replication rule. Replication configuration must have at least one rule and can contain up to 1,000 rules.
 - (*dict*) –
 - **ID** (*string*) – Unique identifier for the rule. The value cannot be longer than 255 characters.
 - **Prefix** (*string*) – Object keyname prefix identifying one or more objects to which the rule applies. Maximum prefix length can be up to 1,024 characters. Overlapping prefixes are not supported.
 - **Status** (*string*) – The rule is ignored if status is not Enabled.
 - **Destination** (*dict*) –
 - **Bucket** (*string*) – Amazon resource name (ARN) of the bucket where you want Amazon S3 to store replicas of the object identified by the rule.
 - **StorageClass** (*string*) – The class of storage used to store the object.

get_bucket_request_payment (***kwargs*)
Returns the request payment configuration of a bucket.

Request Syntax

```
response = client.get_bucket_request_payment(  
    Bucket='string'  
)
```

Parameters **Bucket** (*string*) – [REQUIRED]

Return type dict

Returns

Response Syntax

```
{  
    'Payer': 'Requester'|'BucketOwner'  
}
```

Response Structure

- (*dict*) –
 - **Payer** (*string*) – Specifies who pays for the download and request fees.

get_bucket_tagging (***kwargs*)
Returns the tag set associated with the bucket.

Request Syntax

```
response = client.get_bucket_tagging(  
    Bucket='string'  
)
```

Parameters **Bucket** (*string*) – [REQUIRED]

Return type dict**Returns****Response Syntax**

```
{
    'TagSet': [
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
}
```

Response Structure

- (*dict*) –
 - **TagSet** (*list*) –
 - * (*dict*) –
 - **Key** (*string*) – Name of the tag.
 - **Value** (*string*) – Value of the tag.

get_bucket_versioning (***kwargs*)

Returns the versioning state of a bucket.

Request Syntax

```
response = client.get_bucket_versioning(
    Bucket='string'
)
```

Parameters **Bucket** (*string*) – [REQUIRED]**Return type** dict**Returns****Response Syntax**

```
{
    'Status': 'Enabled'|'Suspended',
    'MFADelete': 'Enabled'|'Disabled'
}
```

Response Structure

- (*dict*) –
 - **Status** (*string*) – The versioning state of the bucket.
 - **MFADelete** (*string*) – Specifies whether MFA delete is enabled in the bucket versioning configuration. This element is only returned if the bucket has been configured with MFA delete. If the bucket has never been so configured, this element is not returned.

get_bucket_website (***kwargs*)

Returns the website configuration for a bucket.

Request Syntax

```
response = client.get_bucket_website(
    Bucket='string'
)
```

Parameters **Bucket** (*string*) – [REQUIRED]

Return type dict

Returns

Response Syntax

```
{
  'RedirectAllRequestsTo': {
    'HostName': 'string',
    'Protocol': 'http'|'https'
  },
  'IndexDocument': {
    'Suffix': 'string'
  },
  'ErrorDocument': {
    'Key': 'string'
  },
  'RoutingRules': [
    {
      'Condition': {
        'HttpErrorCodeReturnedEquals': 'string',
        'KeyPrefixEquals': 'string'
      },
      'Redirect': {
        'HostName': 'string',
        'HttpRedirectCode': 'string',
        'Protocol': 'http'|'https',
        'ReplaceKeyPrefixWith': 'string',
        'ReplaceKeyWith': 'string'
      }
    }
  ]
}
```

Response Structure

- (dict) –
 - **RedirectAllRequestsTo** (dict) –
 - * **HostName** (string) – Name of the host where requests will be redirected.
 - * **Protocol** (string) – Protocol to use (http, https) when redirecting requests. The default is the protocol that is used in the original request.
 - **IndexDocument** (dict) –
 - * **Suffix** (string) – A suffix that is appended to a request that is for a directory on the website endpoint (e.g. if the suffix is index.html and you make a request to samplebucket/images/ the data that is returned will be for the object with the key name images/index.html) The suffix must not be empty and must not include a slash character.
 - **ErrorDocument** (dict) –
 - * **Key** (string) – The object key name to use when a 4XX class error occurs.
 - **RoutingRules** (list) –
 - * (dict) –
 - **Condition** (dict) – A container for describing a condition that must be met for the specified redirect to apply. For example,
 1. If request is for pages in the /docs folder, redirect to the /documents folder.
 2. If request results in HTTP error 4xx, redirect request to another host where you might process the

error.

- **HttpErrorCodeReturnedEquals** (*string*) – The HTTP error code when the redirect is applied. In the event of an error, if the error code equals this value, then the specified redirect is applied. Required when parent element Condition is specified and sibling KeyPrefixEquals is not specified. If both are specified, then both must be true for the redirect to be applied.
- **KeyPrefixEquals** (*string*) – The object key name prefix when the redirect is applied. For example, to redirect requests for ExamplePage.html, the key prefix will be ExamplePage.html. To redirect request for all pages with the prefix docs/, the key prefix will be /docs, which identifies all objects in the docs/ folder. Required when the parent element Condition is specified and sibling HttpErrorCodeReturnedEquals is not specified. If both conditions are specified, both must be true for the redirect to be applied.
- **Redirect** (*dict*) – Container for redirect information. You can redirect requests to another host, to another page, or with another protocol. In the event of an error, you can specify a different error code to return.
- **HostName** (*string*) – The host name to use in the redirect request.
- **HttpRedirectCode** (*string*) – The HTTP redirect code to use on the response. Not required if one of the siblings is present.
- **Protocol** (*string*) – Protocol to use (http, https) when redirecting requests. The default is the protocol that is used in the original request.
- **ReplaceKeyPrefixWith** (*string*) – The object key prefix to use in the redirect request. For example, to redirect requests for all pages with prefix docs/ (objects in the docs/ folder) to documents/, you can set a condition block with KeyPrefixEquals set to docs/ and in the Redirect set ReplaceKeyPrefixWith to /documents. Not required if one of the siblings is present. Can be present only if ReplaceKeyWith is not provided.
- **ReplaceKeyWith** (*string*) – The specific object key to use in the redirect request. For example, redirect request to error.html. Not required if one of the sibling is present. Can be present only if ReplaceKeyPrefixWith is not provided.

get_object (***kwargs*)

Retrieves objects from Amazon S3.

Request Syntax

```
response = client.get_object(
    Bucket='string',
    IfMatch='string',
    IfModifiedSince=datetime(2015, 1, 1),
    IfNoneMatch='string',
    IfUnmodifiedSince=datetime(2015, 1, 1),
    Key='string',
    Range='string',
    ResponseCacheControl='string',
    ResponseContentDisposition='string',
```

```
ResponseContentEncoding='string',
ResponseContentLanguage='string',
ResponseContentType='string',
ResponseExpires=datetime(2015, 1, 1),
VersionId='string',
SSECustomerAlgorithm='string',
SSECustomerKey='string',
RequestPayer='requester'
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **IfMatch** (*string*) – Return the object only if its entity tag (ETag) is the same as the one specified, otherwise return a 412 (precondition failed).
- **IfModifiedSince** (*datetime*) – Return the object only if it has been modified since the specified time, otherwise return a 304 (not modified).
- **IfNoneMatch** (*string*) – Return the object only if its entity tag (ETag) is different from the one specified, otherwise return a 304 (not modified).
- **IfUnmodifiedSince** (*datetime*) – Return the object only if it has not been modified since the specified time, otherwise return a 412 (precondition failed).
- **Key** (*string*) – [REQUIRED]
- **Range** (*string*) – Downloads the specified range bytes of an object. For more information about the HTTP Range header, go to <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35>.
- **ResponseCacheControl** (*string*) – Sets the Cache-Control header of the response.
- **ResponseContentDisposition** (*string*) – Sets the Content-Disposition header of the response.
- **ResponseContentEncoding** (*string*) – Sets the Content-Encoding header of the response.
- **ResponseContentLanguage** (*string*) – Sets the Content-Language header of the response.
- **ResponseContentType** (*string*) – Sets the Content-Type header of the response.
- **ResponseExpires** (*datetime*) – Sets the Expires header of the response.
- **VersionId** (*string*) – VersionId used to reference a specific version of the object.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.
- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required.
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
  'Body': b'bytes',
  'DeleteMarker': True|False,
  'AcceptRanges': 'string',
  'Expiration': 'string',
  'Restore': 'string',
  'LastModified': datetime(2015, 1, 1),
  'ContentLength': 123,
  'ETag': 'string',
  'MissingMeta': 123,
  'VersionId': 'string',
  'CacheControl': 'string',
  'ContentDisposition': 'string',
  'ContentEncoding': 'string',
  'ContentLanguage': 'string',
  'ContentRange': 'string',
  'ContentType': 'string',
  'Expires': datetime(2015, 1, 1),
  'WebsiteRedirectLocation': 'string',
  'ServerSideEncryption': 'AES256'|'aws:kms',
  'Metadata': {
    'string': 'string'
  },
  'SSECustomerAlgorithm': 'string',
  'SSECustomerKeyMD5': 'string',
  'SSEKMSKeyId': 'string',
  'StorageClass': 'STANDARD'|'REDUCED_REDUNDANCY'|'STANDARD_IA',
  'RequestCharged': 'requester',
  'ReplicationStatus': 'COMPLETE'|'PENDING'|'FAILED'|'REPLICA'
}
```

Response Structure

- (*dict*) –
 - **Body** (*bytes*) – Object data.
 - **DeleteMarker** (*boolean*) – Specifies whether the object retrieved was (true) or was not (false) a Delete Marker. If false, this response header does not appear in the response.
 - **AcceptRanges** (*string*) –
 - **Expiration** (*string*) – If the object expiration is configured (see PUT Bucket lifecycle), the response includes this header. It includes the expiry-date and rule-id key value pairs providing object expiration information. The value of the rule-id is URL encoded.
 - **Restore** (*string*) – Provides information about object restoration operation and expiration time of the restored object copy.
 - **LastModified** (*datetime*) – Last modified date of the object
 - **ContentLength** (*integer*) – Size of the body in bytes.
 - **ETag** (*string*) – An ETag is an opaque identifier assigned by a web server to a specific version of a resource found at a URL
 - **MissingMeta** (*integer*) – This is set to the number of metadata entries not returned in x-amz-meta headers. This can happen if you create metadata using an API like SOAP that supports more flexible metadata than the REST API. For example, using SOAP, you can create metadata whose values are not legal HTTP headers.
 - **VersionId** (*string*) – Version of the object.

- **CacheControl** (*string*) – Specifies caching behavior along the request/reply chain.
- **ContentDisposition** (*string*) – Specifies presentational information for the object.
- **ContentEncoding** (*string*) – Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.
- **ContentLanguage** (*string*) – The language the content is in.
- **ContentRange** (*string*) – The portion of the object returned in the response.
- **ContentType** (*string*) – A standard MIME type describing the format of the object data.
- **Expires** (*datetime*) – The date and time at which the object is no longer cacheable.
- **WebsiteRedirectLocation** (*string*) – If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.
- **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
- **Metadata** (*dict*) – A map of metadata to store with the object in S3.
 - * (*string*) –
 - (*string*) –
- **SSECustomerAlgorithm** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.
- **SSECustomerKeyMD5** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.
- **SSEKMSKeyId** (*string*) – If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.
- **StorageClass** (*string*) –
- **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.
- **ReplicationStatus** (*string*) –

get_object_acl (***kwargs*)

Returns the access control list (ACL) of an object.

Request Syntax

```
response = client.get_object_acl(  
    Bucket='string',  
    Key='string',  
    VersionId='string',  
    RequestPayer='requester'  
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **Key** (*string*) – [REQUIRED]
- **VersionId** (*string*) – VersionId used to reference a specific version of the object.
- **RequestPayer** (*string*) – Confirms that the requester knows that

she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
  'Owner': {
    'DisplayName': 'string',
    'ID': 'string'
  },
  'Grants': [
    {
      'Grantee': {
        'DisplayName': 'string',
        'EmailAddress': 'string',
        'ID': 'string',
        'Type': 'CanonicalUser'|'AmazonCustomerByEmail'|'Group',
        'URI': 'string'
      },
      'Permission': 'FULL_CONTROL'|'WRITE'|'WRITE_ACP'|'READ'|'READ_ACP'
    },
    ...
  ],
  'RequestCharged': 'requester'
}
```

Response Structure

- (dict) –
 - **Owner** (dict) –
 - * **DisplayName** (string) –
 - * **ID** (string) –
 - **Grants** (list) – A list of grants.
 - * (dict) –
 - **Grantee** (dict) –
 - **DisplayName** (string) – Screen name of the grantee.
 - **EmailAddress** (string) – Email address of the grantee.
 - **ID** (string) – The canonical user ID of the grantee.
 - **Type** (string) – Type of grantee
 - **URI** (string) – URI of the grantee group.
 - **Permission** (string) – Specifies the permission given to the grantee.
 - **RequestCharged** (string) – If present, indicates that the requester was successfully charged for the request.

get_object_torrent (**kwargs)

Return torrent files from a bucket.

Request Syntax

```
response = client.get_object_torrent(
    Bucket='string',
    Key='string',
    RequestPayer='requester'
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **Key** (*string*) – [REQUIRED]
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict**Returns****Response Syntax**

```
{
    'Body': b'bytes',
    'RequestCharged': 'requester'
}
```

Response Structure

- (*dict*) –
 - **Body** (*bytes*) –
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type L{botocore.paginate.Paginator}**Returns** A paginator object.**get_waiter** (*waiter_name*)**head_bucket** (***kwargs*)

This operation is useful to determine if a bucket exists and you have permission to access it.

Request Syntax

```
response = client.head_bucket(
    Bucket='string'
)
```

Parameters **Bucket** (*string*) – [REQUIRED]**Returns** None**head_object** (***kwargs*)

The HEAD operation retrieves metadata from an object without returning the object itself. This operation is useful if you're only interested in an object's metadata. To use HEAD, you must have READ access to the object.

Request Syntax

```

response = client.head_object(
    Bucket='string',
    IfMatch='string',
    IfModifiedSince=datetime(2015, 1, 1),
    IfNoneMatch='string',
    IfUnmodifiedSince=datetime(2015, 1, 1),
    Key='string',
    Range='string',
    VersionId='string',
    SSECustomerAlgorithm='string',
    SSECustomerKey='string',
    RequestPayer='requester'
)

```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **IfMatch** (*string*) – Return the object only if its entity tag (ETag) is the same as the one specified, otherwise return a 412 (precondition failed).
- **IfModifiedSince** (*datetime*) – Return the object only if it has been modified since the specified time, otherwise return a 304 (not modified).
- **IfNoneMatch** (*string*) – Return the object only if its entity tag (ETag) is different from the one specified, otherwise return a 304 (not modified).
- **IfUnmodifiedSince** (*datetime*) – Return the object only if it has not been modified since the specified time, otherwise return a 412 (precondition failed).
- **Key** (*string*) – [REQUIRED]
- **Range** (*string*) – Downloads the specified range bytes of an object. For more information about the HTTP Range header, go to <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35>.
- **VersionId** (*string*) – VersionId used to reference a specific version of the object.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.
- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```

{
    'DeleteMarker': True|False,
    'AcceptRanges': 'string',
    'Expiration': 'string',

```

```
'Restore': 'string',
'LastModified': datetime(2015, 1, 1),
'ContentLength': 123,
'ETag': 'string',
'MissingMeta': 123,
'VersionId': 'string',
'CacheControl': 'string',
'ContentDisposition': 'string',
'ContentEncoding': 'string',
'ContentLanguage': 'string',
'ContentType': 'string',
'Expires': datetime(2015, 1, 1),
'WebsiteRedirectLocation': 'string',
'ServerSideEncryption': 'AES256'|'aws:kms',
'Metadata': {
    'string': 'string'
},
'SSECustomerAlgorithm': 'string',
'SSECustomerKeyMD5': 'string',
'SSEKMSKeyId': 'string',
'StorageClass': 'STANDARD'|'REDUCED_REDUNDANCY'|'STANDARD_IA',
'RequestCharged': 'requester',
'ReplicationStatus': 'COMPLETE'|'PENDING'|'FAILED'|'REPLICA'
}
```

Response Structure

- (*dict*) –
 - **DeleteMarker** (*boolean*) – Specifies whether the object retrieved was (true) or was not (false) a Delete Marker. If false, this response header does not appear in the response.
 - **AcceptRanges** (*string*) –
 - **Expiration** (*string*) – If the object expiration is configured (see PUT Bucket lifecycle), the response includes this header. It includes the expiry-date and rule-id key value pairs providing object expiration information. The value of the rule-id is URL encoded.
 - **Restore** (*string*) – Provides information about object restoration operation and expiration time of the restored object copy.
 - **LastModified** (*datetime*) – Last modified date of the object
 - **ContentLength** (*integer*) – Size of the body in bytes.
 - **ETag** (*string*) – An ETag is an opaque identifier assigned by a web server to a specific version of a resource found at a URL
 - **MissingMeta** (*integer*) – This is set to the number of metadata entries not returned in x-amz-meta headers. This can happen if you create metadata using an API like SOAP that supports more flexible metadata than the REST API. For example, using SOAP, you can create metadata whose values are not legal HTTP headers.
 - **VersionId** (*string*) – Version of the object.
 - **CacheControl** (*string*) – Specifies caching behavior along the request/reply chain.
 - **ContentDisposition** (*string*) – Specifies presentational information for the object.
 - **ContentEncoding** (*string*) – Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.
 - **ContentLanguage** (*string*) – The language the content is in.
 - **ContentType** (*string*) – A standard MIME type describing the format of

the object data.

- **Expires** (*datetime*) – The date and time at which the object is no longer cacheable.
- **WebsiteRedirectLocation** (*string*) – If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.
- **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
- **Metadata** (*dict*) – A map of metadata to store with the object in S3.
 - * (*string*) –
 - (*string*) –
- **SSECustomerAlgorithm** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.
- **SSECustomerKeyMD5** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.
- **SSEKMSKeyId** (*string*) – If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.
- **StorageClass** (*string*) –
- **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.
- **ReplicationStatus** (*string*) –

list_buckets()

Returns a list of all buckets owned by the authenticated sender of the request.

Request Syntax

```
response = client.list_buckets()
```

Return type dict

Returns

Response Syntax

```
{
  'Buckets': [
    {
      'Name': 'string',
      'CreationDate': datetime(2015, 1, 1)
    },
  ],
  'Owner': {
    'DisplayName': 'string',
    'ID': 'string'
  }
}
```

Response Structure

- (*dict*) –
 - **Buckets** (*list*) –
 - * (*dict*) –
 - **Name** (*string*) – The name of the bucket.

- **CreationDate** (*datetime*) – Date the bucket was created.
- **Owner** (*dict*) –
 - * **DisplayName** (*string*) –
 - * **ID** (*string*) –

list_multipart_uploads (**kwargs)

This operation lists in-progress multipart uploads.

Request Syntax

```
response = client.list_multipart_uploads(  
    Bucket='string',  
    Delimiter='string',  
    EncodingType='url',  
    KeyMarker='string',  
    MaxUploads=123,  
    Prefix='string',  
    UploadIdMarker='string'  
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **Delimiter** (*string*) – Character you use to group keys.
- **EncodingType** (*string*) – Requests Amazon S3 to encode the object keys in the response and specifies the encoding method to use. An object key may contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.
- **KeyMarker** (*string*) – Together with upload-id-marker, this parameter specifies the multipart upload after which listing should begin.
- **MaxUploads** (*integer*) – Sets the maximum number of multipart uploads, from 1 to 1,000, to return in the response body. 1,000 is the maximum number of uploads that can be returned in a response.
- **Prefix** (*string*) – Lists in-progress uploads only for those keys that begin with the specified prefix.
- **UploadIdMarker** (*string*) – Together with key-marker, specifies the multipart upload after which listing should begin. If key-marker is not specified, the upload-id-marker parameter is ignored.

Return type dict

Returns

Response Syntax

```
{  
    'Bucket': 'string',  
    'KeyMarker': 'string',  
    'UploadIdMarker': 'string',  
    'NextKeyMarker': 'string',  
    'Prefix': 'string',  
    'Delimiter': 'string',  
    'NextUploadIdMarker': 'string',  
    'MaxUploads': 123,  
    'IsTruncated': True|False,  
    'Uploads': [  
        {  
            'UploadId': 'string',
```



```

        'Key': 'string',
        'Initiated': datetime(2015, 1, 1),
        'StorageClass': 'STANDARD'|'REDUCED_REDUNDANCY'|'STANDARD_IA',
        'Owner': {
            'DisplayName': 'string',
            'ID': 'string'
        },
        'Initiator': {
            'ID': 'string',
            'DisplayName': 'string'
        }
    },
    ],
    'CommonPrefixes': [
        {
            'Prefix': 'string'
        },
    ],
    ],
    'EncodingType': 'url'
}

```

Response Structure

- *(dict)* –
 - **Bucket** (*string*) – Name of the bucket to which the multipart upload was initiated.
 - **KeyMarker** (*string*) – The key at or after which the listing began.
 - **UploadIdMarker** (*string*) – Upload ID after which listing began.
 - **NextKeyMarker** (*string*) – When a list is truncated, this element specifies the value that should be used for the key-marker request parameter in a subsequent request.
 - **Prefix** (*string*) – When a prefix is provided in the request, this field contains the specified prefix. The result contains only keys starting with the specified prefix.
 - **Delimiter** (*string*) –
 - **NextUploadIdMarker** (*string*) – When a list is truncated, this element specifies the value that should be used for the upload-id-marker request parameter in a subsequent request.
 - **MaxUploads** (*integer*) – Maximum number of multipart uploads that could have been included in the response.
 - **IsTruncated** (*boolean*) – Indicates whether the returned list of multipart uploads is truncated. A value of true indicates that the list was truncated. The list can be truncated if the number of multipart uploads exceeds the limit allowed or specified by max uploads.
 - **Uploads** (*list*) –
 - * *(dict)* –
 - **UploadId** (*string*) – Upload ID that identifies the multipart upload.
 - **Key** (*string*) – Key of the object for which the multipart upload was initiated.
 - **Initiated** (*datetime*) – Date and time at which the multipart upload was initiated.
 - **StorageClass** (*string*) – The class of storage used to store the object.
 - **Owner** (*dict*) –
 - **DisplayName** (*string*) –

- **ID** (*string*) –
- **Initiator** (*dict*) – Identifies who initiated the multipart upload.
- **ID** (*string*) – If the principal is an AWS account, it provides the Canonical User ID. If the principal is an IAM User, it provides a user ARN value.
- **DisplayName** (*string*) – Name of the Principal.
- **CommonPrefixes** (*list*) –
 - * (*dict*) –
 - **Prefix** (*string*) –
- **EncodingType** (*string*) – Encoding type used by Amazon S3 to encode object keys in the response.

list_object_versions (***kwargs*)

Returns metadata about all of the versions of objects in a bucket.

Request Syntax

```
response = client.list_object_versions(  
    Bucket='string',  
    Delimiter='string',  
    EncodingType='url',  
    KeyMarker='string',  
    MaxKeys=123,  
    Prefix='string',  
    VersionIdMarker='string'  
)
```

Parameters

- **Bucket** (*string*) – **[REQUIRED]**
- **Delimiter** (*string*) – A delimiter is a character you use to group keys.
- **EncodingType** (*string*) – Requests Amazon S3 to encode the object keys in the response and specifies the encoding method to use. An object key may contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.
- **KeyMarker** (*string*) – Specifies the key to start with when listing objects in a bucket.
- **MaxKeys** (*integer*) – Sets the maximum number of keys returned in the response. The response might contain fewer keys but will never contain more.
- **Prefix** (*string*) – Limits the response to keys that begin with the specified prefix.
- **VersionIdMarker** (*string*) – Specifies the object version you want to start listing from.

Return type dict

Returns

Response Syntax

```
{  
    'IsTruncated': True|False,  
    'KeyMarker': 'string',  
    'VersionIdMarker': 'string',  
    'NextKeyMarker': 'string',  
    'NextVersionIdMarker': 'string',  
    'Versions': [  
        {  
            'ETag': 'string',
```

```

        'Size': 123,
        'StorageClass': 'STANDARD',
        'Key': 'string',
        'VersionId': 'string',
        'IsLatest': True|False,
        'LastModified': datetime(2015, 1, 1),
        'Owner': {
            'DisplayName': 'string',
            'ID': 'string'
        }
    },
    'DeleteMarkers': [
        {
            'Owner': {
                'DisplayName': 'string',
                'ID': 'string'
            },
            'Key': 'string',
            'VersionId': 'string',
            'IsLatest': True|False,
            'LastModified': datetime(2015, 1, 1)
        },
        ...
    ],
    'Name': 'string',
    'Prefix': 'string',
    'Delimiter': 'string',
    'MaxKeys': 123,
    'CommonPrefixes': [
        {
            'Prefix': 'string'
        },
        ...
    ],
    'EncodingType': 'url'
}

```

Response Structure

- *(dict)* –
 - **IsTruncated** (*boolean*) – A flag that indicates whether or not Amazon S3 returned all of the results that satisfied the search criteria. If your results were truncated, you can make a follow-up paginated request using the `NextKeyMarker` and `NextVersionIdMarker` response parameters as a starting place in another request to return the rest of the results.
 - **KeyMarker** (*string*) – Marks the last Key returned in a truncated response.
 - **VersionIdMarker** (*string*) –
 - **NextKeyMarker** (*string*) – Use this value for the key marker request parameter in a subsequent request.
 - **NextVersionIdMarker** (*string*) – Use this value for the next version id marker parameter in a subsequent request.
 - **Versions** (*list*) –
 - * *(dict)* –
 - **ETag** (*string*) –
 - **Size** (*integer*) – Size in bytes of the object.
 - **StorageClass** (*string*) – The class of storage used to store the object.
 - **Key** (*string*) – The object key.
 - **VersionId** (*string*) – Version ID of an object.

- **IsLatest** (*boolean*) – Specifies whether the object is (true) or is not (false) the latest version of an object.
- **LastModified** (*datetime*) – Date and time the object was last modified.
- **Owner** (*dict*) –
- **DisplayName** (*string*) –
- **ID** (*string*) –
- **DeleteMarkers** (*list*) –
 - * (*dict*) –
 - **Owner** (*dict*) –
 - **DisplayName** (*string*) –
 - **ID** (*string*) –
 - **Key** (*string*) – The object key.
 - **VersionId** (*string*) – Version ID of an object.
 - **IsLatest** (*boolean*) – Specifies whether the object is (true) or is not (false) the latest version of an object.
 - **LastModified** (*datetime*) – Date and time the object was last modified.
- **Name** (*string*) –
- **Prefix** (*string*) –
- **Delimiter** (*string*) –
- **MaxKeys** (*integer*) –
- **CommonPrefixes** (*list*) –
 - * (*dict*) –
 - **Prefix** (*string*) –
- **EncodingType** (*string*) – Encoding type used by Amazon S3 to encode object keys in the response.

list_objects (**kwargs)

Returns some or all (up to 1000) of the objects in a bucket. You can use the request parameters as selection criteria to return a subset of the objects in a bucket.

Request Syntax

```
response = client.list_objects(  
    Bucket='string',  
    Delimiter='string',  
    EncodingType='url',  
    Marker='string',  
    MaxKeys=123,  
    Prefix='string'  
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **Delimiter** (*string*) – A delimiter is a character you use to group keys.
- **EncodingType** (*string*) – Requests Amazon S3 to encode the object keys in the response and specifies the encoding method to use. An object key may contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.
- **Marker** (*string*) – Specifies the key to start with when listing objects in a bucket.
- **MaxKeys** (*integer*) – Sets the maximum number of keys returned in the response. The response might contain fewer keys but will never contain more.
- **Prefix** (*string*) – Limits the response to keys that begin with the specified prefix.

Return type dict

Returns

Response Syntax

```
{
    'IsTruncated': True|False,
    'Marker': 'string',
    'NextMarker': 'string',
    'Contents': [
        {
            'Key': 'string',
            'LastModified': datetime(2015, 1, 1),
            'ETag': 'string',
            'Size': 123,
            'StorageClass': 'STANDARD'|'REDUCED_REDUNDANCY'|'GLACIER',
            'Owner': {
                'DisplayName': 'string',
                'ID': 'string'
            }
        },
    ],
    'Name': 'string',
    'Prefix': 'string',
    'Delimiter': 'string',
    'MaxKeys': 123,
    'CommonPrefixes': [
        {
            'Prefix': 'string'
        },
    ],
    'EncodingType': 'url'
}
```

Response Structure

- (dict) –
 - **IsTruncated** (boolean) – A flag that indicates whether or not Amazon S3 returned all of the results that satisfied the search criteria.
 - **Marker** (string) –
 - **NextMarker** (string) – When response is truncated (the IsTruncated element value in the response is true), you can use the key name in this field as marker in the subsequent request to get next set of objects. Amazon S3 lists objects in alphabetical order Note: This element is returned only if you have delimiter request parameter specified. If response does not include the NextMaker and it is truncated, you can use the value of the last Key in the response as the marker in the subsequent request to get the next set of object keys.
 - **Contents** (list) –
 - * (dict) –
 - **Key** (string) –
 - **LastModified** (datetime) –
 - **ETag** (string) –
 - **Size** (integer) –
 - **StorageClass** (string) – The class of storage used to store the object.
 - **Owner** (dict) –
 - **DisplayName** (string) –

- **ID** (*string*) –
- **Name** (*string*) –
- **Prefix** (*string*) –
- **Delimiter** (*string*) –
- **MaxKeys** (*integer*) –
- **CommonPrefixes** (*list*) –
 - * (*dict*) –
 - **Prefix** (*string*) –
- **EncodingType** (*string*) – Encoding type used by Amazon S3 to encode object keys in the response.

list_parts (***kwargs*)

Lists the parts that have been uploaded for a specific multipart upload.

Request Syntax

```
response = client.list_parts(  
    Bucket='string',  
    Key='string',  
    MaxParts=123,  
    PartNumberMarker=123,  
    UploadId='string',  
    RequestPayer='requester'  
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **Key** (*string*) – [REQUIRED]
- **MaxParts** (*integer*) – Sets the maximum number of parts to return.
- **PartNumberMarker** (*integer*) – Specifies the part after which listing should begin. Only parts with higher part numbers will be listed.
- **UploadId** (*string*) – [REQUIRED] Upload ID identifying the multipart upload whose parts are being listed.
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{  
    'Bucket': 'string',  
    'Key': 'string',  
    'UploadId': 'string',  
    'PartNumberMarker': 123,  
    'NextPartNumberMarker': 123,  
    'MaxParts': 123,  
    'IsTruncated': True|False,  
    'Parts': [  
        {  
            'PartNumber': 123,  
            'LastModified': datetime(2015, 1, 1),  
            'ETag': 'string',  
            'Size': 123
```

```

    },
  ],
  'Initiator': {
    'ID': 'string',
    'DisplayName': 'string'
  },
  'Owner': {
    'DisplayName': 'string',
    'ID': 'string'
  },
  'StorageClass': 'STANDARD' | 'REDUCED_REDUNDANCY' | 'STANDARD_IA',
  'RequestCharged': 'requester'
}

```

Response Structure

- *(dict)* –
 - **Bucket** (*string*) – Name of the bucket to which the multipart upload was initiated.
 - **Key** (*string*) – Object key for which the multipart upload was initiated.
 - **UploadId** (*string*) – Upload ID identifying the multipart upload whose parts are being listed.
 - **PartNumberMarker** (*integer*) – Part number after which listing begins.
 - **NextPartNumberMarker** (*integer*) – When a list is truncated, this element specifies the last part in the list, as well as the value to use for the part-number-marker request parameter in a subsequent request.
 - **MaxParts** (*integer*) – Maximum number of parts that were allowed in the response.
 - **IsTruncated** (*boolean*) – Indicates whether the returned list of parts is truncated.
 - **Parts** (*list*) –
 - * *(dict)* –
 - **PartNumber** (*integer*) – Part number identifying the part. This is a positive integer between 1 and 10,000.
 - **LastModified** (*datetime*) – Date and time at which the part was uploaded.
 - **ETag** (*string*) – Entity tag returned when the part was uploaded.
 - **Size** (*integer*) – Size of the uploaded part data.
 - **Initiator** (*dict*) – Identifies who initiated the multipart upload.
 - * **ID** (*string*) – If the principal is an AWS account, it provides the Canonical User ID. If the principal is an IAM User, it provides a user ARN value.
 - * **DisplayName** (*string*) – Name of the Principal.
 - **Owner** (*dict*) –
 - * **DisplayName** (*string*) –
 - * **ID** (*string*) –
 - **StorageClass** (*string*) – The class of storage used to store the object.
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

put_bucket_acl (***kwargs*)

Sets the permissions on a bucket using access control lists (ACL).

Request Syntax

```
response = client.put_bucket_acl(
    ACL='private'|'public-read'|'public-read-write'|'authenticated-read',
    AccessControlPolicy={
        'Grants': [
            {
                'Grantee': {
                    'DisplayName': 'string',
                    'EmailAddress': 'string',
                    'ID': 'string',
                    'Type': 'CanonicalUser'|'AmazonCustomerByEmail'|'Group',
                    'URI': 'string'
                },
                'Permission': 'FULL_CONTROL'|'WRITE'|'WRITE_ACP'|'READ'|'READ_ACP'
            },
        ],
        'Owner': {
            'DisplayName': 'string',
            'ID': 'string'
        }
    },
    Bucket='string',
    GrantFullControl='string',
    GrantRead='string',
    GrantReadACP='string',
    GrantWrite='string',
    GrantWriteACP='string'
)
```

Parameters

- **ACL** (*string*) – The canned ACL to apply to the bucket.
- **AccessControlPolicy** (*dict*) –
 - **Grants** (*list*) – A list of grants.
 - * (*dict*) –
 - **Grantee** (*dict*) –
 - **DisplayName** (*string*) – Screen name of the grantee.
 - **EmailAddress** (*string*) – Email address of the grantee.
 - **ID** (*string*) – The canonical user ID of the grantee.
 - **Type** (*string*) – [REQUIRED] Type of grantee
 - **URI** (*string*) – URI of the grantee group.
 - **Permission** (*string*) – Specifies the permission given to the grantee.
 - **Owner** (*dict*) –
 - * **DisplayName** (*string*) –
 - * **ID** (*string*) –
- **Bucket** (*string*) – [REQUIRED]
- **GrantFullControl** (*string*) – Allows grantee the read, write, read ACP, and write ACP permissions on the bucket.
- **GrantRead** (*string*) – Allows grantee to list the objects in the bucket.
- **GrantReadACP** (*string*) – Allows grantee to read the bucket ACL.
- **GrantWrite** (*string*) – Allows grantee to create, overwrite, and delete any object in the bucket.
- **GrantWriteACP** (*string*) – Allows grantee to write the ACL for the applicable bucket.

Returns None

put_bucket_cors (***kwargs*)

Sets the cors configuration for a bucket.

Request Syntax

```
response = client.put_bucket_cors(
    Bucket='string',
    CORSConfiguration={
        'CORSRules': [
            {
                'AllowedHeaders': [
                    'string',
                ],
                'AllowedMethods': [
                    'string',
                ],
                'AllowedOrigins': [
                    'string',
                ],
                'ExposeHeaders': [
                    'string',
                ],
                'MaxAgeSeconds': 123
            },
        ]
    },
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **CORSConfiguration** (*dict*) – [REQUIRED]
 - **CORSRules** (*list*) – [REQUIRED]
 - * (*dict*) –
 - **AllowedHeaders** (*list*) – Specifies which headers are allowed in a pre-flight OPTIONS request.
 - (*string*) –
 - **AllowedMethods** (*list*) – [REQUIRED] Identifies HTTP methods that the domain/origin specified in the rule is allowed to execute.
 - (*string*) –
 - **AllowedOrigins** (*list*) – [REQUIRED] One or more origins you want customers to be able to access the bucket from.
 - (*string*) –
 - **ExposeHeaders** (*list*) – One or more headers in the response that you want customers to be able to access from their applications (for example, from a JavaScript XMLHttpRequest object).
 - (*string*) –
 - **MaxAgeSeconds** (*integer*) – The time in seconds that your browser is to cache the preflight response for the specified resource.

Returns None

put_bucket_lifecycle (***kwargs*)

Deprecated, see the PutBucketLifecycleConfiguration operation.

Request Syntax

```
response = client.put_bucket_lifecycle(
    Bucket='string',
    LifecycleConfiguration={
        'Rules': [
            {
                'Expiration': {
                    'Date': datetime(2015, 1, 1),
                    'Days': 123
                },
                'ID': 'string',
                'Prefix': 'string',
                'Status': 'Enabled'|'Disabled',
                'Transition': {
                    'Date': datetime(2015, 1, 1),
                    'Days': 123,
                    'StorageClass': 'GLACIER'|'STANDARD_IA'
                },
                'NoncurrentVersionTransition': {
                    'NoncurrentDays': 123,
                    'StorageClass': 'GLACIER'|'STANDARD_IA'
                },
                'NoncurrentVersionExpiration': {
                    'NoncurrentDays': 123
                }
            }
        ]
    }
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **LifecycleConfiguration** (*dict*) –
 - **Rules** (*list*) – [REQUIRED]
 - * (*dict*) –
 - **Expiration** (*dict*) –
 - **Date** (*datetime*) – Indicates at what date the object is to be moved or deleted. Should be in GMT ISO 8601 Format.
 - **Days** (*integer*) – Indicates the lifetime, in days, of the objects that are subject to the rule. The value must be a non-zero positive integer.
 - **ID** (*string*) – Unique identifier for the rule. The value cannot be longer than 255 characters.
 - **Prefix** (*string*) – [REQUIRED] Prefix identifying one or more objects to which the rule applies.
 - **Status** (*string*) – [REQUIRED] If 'Enabled', the rule is currently being applied. If 'Disabled', the rule is not currently being applied.
 - **Transition** (*dict*) –
 - **Date** (*datetime*) – Indicates at what date the object is to be moved or deleted. Should be in GMT ISO 8601 Format.
 - **Days** (*integer*) – Indicates the lifetime, in days, of the objects that are subject to the rule. The value must be a non-zero positive integer.
 - **StorageClass** (*string*) – The class of storage used to store the object.

- **NoncurrentVersionTransition** (*dict*) – Container for the transition rule that describes when noncurrent objects transition to the STANDARD_IA or GLACIER storage class. If your bucket is versioning-enabled (or versioning is suspended), you can set this action to request that Amazon S3 transition noncurrent object versions to the STANDARD_IA or GLACIER storage class at a specific period in the object's lifetime.
- **NoncurrentDays** (*integer*) – Specifies the number of days an object is noncurrent before Amazon S3 can perform the associated action. For information about the noncurrent days calculations, see [How Amazon S3 Calculates When an Object Became Noncurrent](#) in the Amazon Simple Storage Service Developer Guide.
- **StorageClass** (*string*) – The class of storage used to store the object.
- **NoncurrentVersionExpiration** (*dict*) – Specifies when noncurrent object versions expire. Upon expiration, Amazon S3 permanently deletes the noncurrent object versions. You set this lifecycle configuration action on a bucket that has versioning enabled (or suspended) to request that Amazon S3 delete noncurrent object versions at a specific period in the object's lifetime.
- **NoncurrentDays** (*integer*) – Specifies the number of days an object is noncurrent before Amazon S3 can perform the associated action. For information about the noncurrent days calculations, see [How Amazon S3 Calculates When an Object Became Noncurrent](#) in the Amazon Simple Storage Service Developer Guide.

Returns None

put_bucket_lifecycle_configuration (**kwargs)

Sets lifecycle configuration for your bucket. If a lifecycle configuration exists, it replaces it.

Request Syntax

```
response = client.put_bucket_lifecycle_configuration(
    Bucket='string',
    LifecycleConfiguration={
        'Rules': [
            {
                'Expiration': {
                    'Date': datetime(2015, 1, 1),
                    'Days': 123
                },
                'ID': 'string',
                'Prefix': 'string',
                'Status': 'Enabled'|'Disabled',
                'Transitions': [
                    {
                        'Date': datetime(2015, 1, 1),
                        'Days': 123,
                        'StorageClass': 'GLACIER'|'STANDARD_IA'
                    },
                ],
                'NoncurrentVersionTransitions': [
```

```
        {
            'NoncurrentDays': 123,
            'StorageClass': 'GLACIER' | 'STANDARD_IA'
        },
    ],
    'NoncurrentVersionExpiration': {
        'NoncurrentDays': 123
    }
},
]
}
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **LifecycleConfiguration** (*dict*) –
 - **Rules** (*list*) – [REQUIRED]
 - * (*dict*) –
 - **Expiration** (*dict*) –
 - **Date** (*datetime*) – Indicates at what date the object is to be moved or deleted. Should be in GMT ISO 8601 Format.
 - **Days** (*integer*) – Indicates the lifetime, in days, of the objects that are subject to the rule. The value must be a non-zero positive integer.
 - **ID** (*string*) – Unique identifier for the rule. The value cannot be longer than 255 characters.
 - **Prefix** (*string*) – [REQUIRED] Prefix identifying one or more objects to which the rule applies.
 - **Status** (*string*) – [REQUIRED] If 'Enabled', the rule is currently being applied. If 'Disabled', the rule is not currently being applied.
 - **Transitions** (*list*) –
 - (*dict*) –
 - **Date** (*datetime*) – Indicates at what date the object is to be moved or deleted. Should be in GMT ISO 8601 Format.
 - **Days** (*integer*) – Indicates the lifetime, in days, of the objects that are subject to the rule. The value must be a non-zero positive integer.
 - **StorageClass** (*string*) – The class of storage used to store the object.
 - **NoncurrentVersionTransitions** (*list*) –
 - (*dict*) – Container for the transition rule that describes when noncurrent objects transition to the STANDARD_IA or GLACIER storage class. If your bucket is versioning-enabled (or versioning is suspended), you can set this action to request that Amazon S3 transition noncurrent object versions to the STANDARD_IA or GLACIER storage class at a specific period in the object's lifetime.
 - **NoncurrentDays** (*integer*) – Specifies the number of days an object is noncurrent before Amazon S3 can perform the associated action. For information about the noncurrent days calculations, see [How Amazon S3 Calculates When an Object Became Noncurrent in the Amazon Simple Storage Service Developer Guide](#).
 - **StorageClass** (*string*) – The class of storage used to store the

object.

- **NoncurrentVersionExpiration** (*dict*) – Specifies when non-current object versions expire. Upon expiration, Amazon S3 permanently deletes the noncurrent object versions. You set this lifecycle configuration action on a bucket that has versioning enabled (or suspended) to request that Amazon S3 delete noncurrent object versions at a specific period in the object's lifetime.
- **NoncurrentDays** (*integer*) – Specifies the number of days an object is noncurrent before Amazon S3 can perform the associated action. For information about the noncurrent days calculations, see [How Amazon S3 Calculates When an Object Became Noncurrent in the Amazon Simple Storage Service Developer Guide](#).

Returns None

`put_bucket_logging` (***kwargs*)

Set the logging parameters for a bucket and to specify permissions for who can view and modify the logging parameters. To set the logging status of a bucket, you must be the bucket owner.

Request Syntax

```
response = client.put_bucket_logging(
    Bucket='string',
    BucketLoggingStatus={
        'LoggingEnabled': {
            'TargetBucket': 'string',
            'TargetGrants': [
                {
                    'Grantee': {
                        'DisplayName': 'string',
                        'EmailAddress': 'string',
                        'ID': 'string',
                        'Type': 'CanonicalUser'|'AmazonCustomerByEmail'|'Group',
                        'URI': 'string'
                    },
                    'Permission': 'FULL_CONTROL'|'READ'|'WRITE'
                },
            ],
            'TargetPrefix': 'string'
        }
    },
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **BucketLoggingStatus** (*dict*) – [REQUIRED]
 - **LoggingEnabled** (*dict*) –
 - * **TargetBucket** (*string*) – Specifies the bucket where you want Amazon S3 to store server access logs. You can have your logs delivered to any bucket that you own, including the same bucket that is being logged. You can also configure multiple buckets to deliver their logs to the same target bucket. In this case you should choose a different **TargetPrefix** for each source bucket so that the delivered log files can be distinguished by key.
 - * **TargetGrants** (*list*) –

- *(dict)* –
- **Grantee** (*dict*) –
- **DisplayName** (*string*) – Screen name of the grantee.
- **EmailAddress** (*string*) – Email address of the grantee.
- **ID** (*string*) – The canonical user ID of the grantee.
- **Type** (*string*) – **[REQUIRED]** Type of grantee
- **URI** (*string*) – URI of the grantee group.
- **Permission** (*string*) – Logging permissions assigned to the Grantee for the bucket.
- * **TargetPrefix** (*string*) – This element lets you specify a prefix for the keys that the log files will be stored under.

Returns None

put_bucket_notification (**kwargs)

Deprecated, see the PutBucketNotificationConfiguraiton operation.

Request Syntax

```
response = client.put_bucket_notification(
    Bucket='string',
    NotificationConfiguration={
        'TopicConfiguration': {
            'Id': 'string',
            'Events': [
                's3:ReducedRedundancyLostObject' | 's3:ObjectCreated:*' | 's3:ObjectCreated:Put
            ],
            'Event': 's3:ReducedRedundancyLostObject' | 's3:ObjectCreated:*' | 's3:ObjectCreate
            'Topic': 'string'
        },
        'QueueConfiguration': {
            'Id': 'string',
            'Event': 's3:ReducedRedundancyLostObject' | 's3:ObjectCreated:*' | 's3:ObjectCreate
            'Events': [
                's3:ReducedRedundancyLostObject' | 's3:ObjectCreated:*' | 's3:ObjectCreated:Put
            ],
            'Queue': 'string'
        },
        'CloudFunctionConfiguration': {
            'Id': 'string',
            'Event': 's3:ReducedRedundancyLostObject' | 's3:ObjectCreated:*' | 's3:ObjectCreate
            'Events': [
                's3:ReducedRedundancyLostObject' | 's3:ObjectCreated:*' | 's3:ObjectCreated:Put
            ],
            'CloudFunction': 'string',
            'InvocationRole': 'string'
        }
    }
)
```

Parameters

- **Bucket** (*string*) – **[REQUIRED]**
- **NotificationConfiguration** (*dict*) – **[REQUIRED]**
 - **TopicConfiguration** (*dict*) –
 - * **Id** (*string*) – Optional unique identifier for configurations in a notification configuration. If you don’t provide one, Amazon S3 will assign an ID.
 - * **Events** (*list*) –

- (*string*) – Bucket event for which to send notifications.
- * **Event** (*string*) – Bucket event for which to send notifications.
- * **Topic** (*string*) – Amazon SNS topic to which Amazon S3 will publish a message to report the specified events for the bucket.
- **QueueConfiguration** (*dict*) –
 - * **Id** (*string*) – Optional unique identifier for configurations in a notification configuration. If you don’t provide one, Amazon S3 will assign an ID.
 - * **Event** (*string*) – Bucket event for which to send notifications.
 - * **Events** (*list*) –
 - (*string*) – Bucket event for which to send notifications.
 - * **Queue** (*string*) –
- **CloudFunctionConfiguration** (*dict*) –
 - * **Id** (*string*) – Optional unique identifier for configurations in a notification configuration. If you don’t provide one, Amazon S3 will assign an ID.
 - * **Event** (*string*) – Bucket event for which to send notifications.
 - * **Events** (*list*) –
 - (*string*) – Bucket event for which to send notifications.
 - * **CloudFunction** (*string*) –
 - * **InvocationRole** (*string*) –

Returns None

put_bucket_notification_configuration (**kwargs)

Enables notifications of specified events for a bucket.

Request Syntax

```
response = client.put_bucket_notification_configuration(
    Bucket='string',
    NotificationConfiguration={
        'TopicConfigurations': [
            {
                'Id': 'string',
                'TopicArn': 'string',
                'Events': [
                    's3:ReducedRedundancyLostObject'|'s3:ObjectCreated:*'|'s3:ObjectCreated:
                ],
                'Filter': {
                    'Key': {
                        'FilterRules': [
                            {
                                'Name': 'prefix'|'suffix',
                                'Value': 'string'
                            },
                        ],
                    },
                },
            },
        ],
        'QueueConfigurations': [
            {
                'Id': 'string',
                'QueueArn': 'string',
                'Events': [
                    's3:ReducedRedundancyLostObject'|'s3:ObjectCreated:*'|'s3:ObjectCreated:
                ],
            },
        ],
    },
)
```

```
        'Filter': {
            'Key': {
                'FilterRules': [
                    {
                        'Name': 'prefix'|'suffix',
                        'Value': 'string'
                    },
                ]
            }
        },
    ],
    'LambdaFunctionConfigurations': [
        {
            'Id': 'string',
            'LambdaFunctionArn': 'string',
            'Events': [
                's3:ReducedRedundancyLostObject'|'s3:ObjectCreated:*'|'s3:ObjectCreated:
            ],
            'Filter': {
                'Key': {
                    'FilterRules': [
                        {
                            'Name': 'prefix'|'suffix',
                            'Value': 'string'
                        },
                    ]
                }
            }
        },
    ],
}
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **NotificationConfiguration** (*dict*) – [REQUIRED] Container for specifying the notification configuration of the bucket. If this element is empty, notifications are turned off on the bucket.
 - **TopicConfigurations** (*list*) –
 - * (*dict*) – Container for specifying the configuration when you want Amazon S3 to publish events to an Amazon Simple Notification Service (Amazon SNS) topic.
 - **Id** (*string*) – Optional unique identifier for configurations in a notification configuration. If you don't provide one, Amazon S3 will assign an ID.
 - **TopicArn** (*string*) – [REQUIRED] Amazon SNS topic ARN to which Amazon S3 will publish a message when it detects events of specified type.
 - **Events** (*list*) – [REQUIRED]
 - (*string*) – Bucket event for which to send notifications.
 - **Filter** (*dict*) – Container for object key name filtering rules. For information about key name filtering, go to [Configuring Event Notifications](#) in the Amazon Simple Storage Service Developer Guide.
 - **Key** (*dict*) – Container for object key name prefix and suffix

- filtering rules.
- **FilterRules** (*list*) – A list of containers for key value pair that defines the criteria for the filter rule.
- (*dict*) – Container for key value pair that defines the criteria for the filter rule.
- **Name** (*string*) – Object key name prefix or suffix identifying one or more objects to which the filtering rule applies. Maximum prefix length can be up to 1,024 characters. Overlapping prefixes and suffixes are not supported. For more information, go to [Configuring Event Notifications](#) in the Amazon Simple Storage Service Developer Guide.
- **Value** (*string*) –
- **QueueConfigurations** (*list*) –
 - * (*dict*) – Container for specifying an configuration when you want Amazon S3 to publish events to an Amazon Simple Queue Service (Amazon SQS) queue.
 - **Id** (*string*) – Optional unique identifier for configurations in a notification configuration. If you don't provide one, Amazon S3 will assign an ID.
 - **QueueArn** (*string*) – **[REQUIRED]** Amazon SQS queue ARN to which Amazon S3 will publish a message when it detects events of specified type.
 - **Events** (*list*) – **[REQUIRED]**
 - (*string*) – Bucket event for which to send notifications.
 - **Filter** (*dict*) – Container for object key name filtering rules. For information about key name filtering, go to [Configuring Event Notifications](#) in the Amazon Simple Storage Service Developer Guide.
 - **Key** (*dict*) – Container for object key name prefix and suffix filtering rules.
 - **FilterRules** (*list*) – A list of containers for key value pair that defines the criteria for the filter rule.
 - (*dict*) – Container for key value pair that defines the criteria for the filter rule.
 - **Name** (*string*) – Object key name prefix or suffix identifying one or more objects to which the filtering rule applies. Maximum prefix length can be up to 1,024 characters. Overlapping prefixes and suffixes are not supported. For more information, go to [Configuring Event Notifications](#) in the Amazon Simple Storage Service Developer Guide.
 - **Value** (*string*) –
- **LambdaFunctionConfigurations** (*list*) –
 - * (*dict*) – Container for specifying the AWS Lambda notification configuration.
 - **Id** (*string*) – Optional unique identifier for configurations in a notification configuration. If you don't provide one, Amazon S3 will assign an ID.
 - **LambdaFunctionArn** (*string*) – **[REQUIRED]** Lambda cloud function ARN that Amazon S3 can invoke when it detects events of the specified type.
 - **Events** (*list*) – **[REQUIRED]**
 - (*string*) – Bucket event for which to send notifications.
 - **Filter** (*dict*) – Container for object key name filtering rules. For information about key name filtering, go to [Configuring](#)

[Event Notifications](#) in the Amazon Simple Storage Service Developer Guide.

- **Key** (*dict*) – Container for object key name prefix and suffix filtering rules.
- **FilterRules** (*list*) – A list of containers for key value pair that defines the criteria for the filter rule.
- (*dict*) – Container for key value pair that defines the criteria for the filter rule.
- **Name** (*string*) – Object key name prefix or suffix identifying one or more objects to which the filtering rule applies. Maximum prefix length can be up to 1,024 characters. Overlapping prefixes and suffixes are not supported. For more information, go to [Configuring Event Notifications](#) in the Amazon Simple Storage Service Developer Guide.
- **Value** (*string*) –

Returns None

put_bucket_policy (***kwargs*)

Replaces a policy on a bucket. If the bucket already has a policy, the one in this request completely replaces it.

Request Syntax

```
response = client.put_bucket_policy(
    Bucket='string',
    Policy='string'
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **Policy** (*string*) – [REQUIRED] The bucket policy as a JSON document.

Returns None

put_bucket_replication (***kwargs*)

Creates a new replication configuration (or replaces an existing one, if present).

Request Syntax

```
response = client.put_bucket_replication(
    Bucket='string',
    ReplicationConfiguration={
        'Role': 'string',
        'Rules': [
            {
                'ID': 'string',
                'Prefix': 'string',
                'Status': 'Enabled' | 'Disabled',
                'Destination': {
                    'Bucket': 'string',
                    'StorageClass': 'STANDARD' | 'REDUCED_REDUNDANCY' | 'STANDARD_IA'
                }
            },
        ],
    }
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]

- **ReplicationConfiguration** (*dict*) – **[REQUIRED]** Container for replication rules. You can add as many as 1,000 rules. Total replication configuration size can be up to 2 MB.
 - **Role** (*string*) – **[REQUIRED]** Amazon Resource Name (ARN) of an IAM role for Amazon S3 to assume when replicating the objects.
 - **Rules** (*list*) – **[REQUIRED]** Container for information about a particular replication rule. Replication configuration must have at least one rule and can contain up to 1,000 rules.
 - * (*dict*) –
 - **ID** (*string*) – Unique identifier for the rule. The value cannot be longer than 255 characters.
 - **Prefix** (*string*) – **[REQUIRED]** Object keyname prefix identifying one or more objects to which the rule applies. Maximum prefix length can be up to 1,024 characters. Overlapping prefixes are not supported.
 - **Status** (*string*) – **[REQUIRED]** The rule is ignored if status is not Enabled.
 - **Destination** (*dict*) – **[REQUIRED]**
 - **Bucket** (*string*) – **[REQUIRED]** Amazon resource name (ARN) of the bucket where you want Amazon S3 to store replicas of the object identified by the rule.
 - **StorageClass** (*string*) – The class of storage used to store the object.

Returns None

put_bucket_request_payment (***kwargs*)

Sets the request payment configuration for a bucket. By default, the bucket owner pays for downloads from the bucket. This configuration parameter enables the bucket owner (only) to specify that the person requesting the download will be charged for the download. Documentation on requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/RequesterPaysBuckets.html>

Request Syntax

```
response = client.put_bucket_request_payment(
    Bucket='string',
    RequestPaymentConfiguration={
        'Payer': 'Requester' | 'BucketOwner'
    }
)
```

Parameters

- **Bucket** (*string*) – **[REQUIRED]**
- **RequestPaymentConfiguration** (*dict*) – **[REQUIRED]**
 - **Payer** (*string*) – **[REQUIRED]** Specifies who pays for the download and request fees.

Returns None

put_bucket_tagging (***kwargs*)

Sets the tags for a bucket.

Request Syntax

```
response = client.put_bucket_tagging(
    Bucket='string',
    Tagging={
        'TagSet': [
```

```
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
}
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **Tagging** (*dict*) – [REQUIRED]
 - **TagSet** (*list*) – [REQUIRED]
 - * (*dict*) –
 - **Key** (*string*) – [REQUIRED] Name of the tag.
 - **Value** (*string*) – [REQUIRED] Value of the tag.

Returns None

put_bucket_versioning (***kwargs*)

Sets the versioning state of an existing bucket. To set the versioning state, you must be the bucket owner.

Request Syntax

```
response = client.put_bucket_versioning(
    Bucket='string',
    MFA='string',
    VersioningConfiguration={
        'MFADelete': 'Enabled'|'Disabled',
        'Status': 'Enabled'|'Suspended'
    }
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **MFA** (*string*) – The concatenation of the authentication device’s serial number, a space, and the value that is displayed on your authentication device.
- **VersioningConfiguration** (*dict*) – [REQUIRED]
 - **MFADelete** (*string*) – Specifies whether MFA delete is enabled in the bucket versioning configuration. This element is only returned if the bucket has been configured with MFA delete. If the bucket has never been so configured, this element is not returned.
 - **Status** (*string*) – The versioning state of the bucket.

Returns None

put_bucket_website (***kwargs*)

Set the website configuration for a bucket.

Request Syntax

```
response = client.put_bucket_website(
    Bucket='string',
    WebsiteConfiguration={
        'ErrorDocument': {
            'Key': 'string'
        },
        'IndexDocument': {
            'Suffix': 'string'
        },
    },
)
```

```

'RedirectAllRequestsTo': {
    'HostName': 'string',
    'Protocol': 'http'|'https'
},
'RoutingRules': [
    {
        'Condition': {
            'HttpErrorCodeReturnedEquals': 'string',
            'KeyPrefixEquals': 'string'
        },
        'Redirect': {
            'HostName': 'string',
            'HttpRedirectCode': 'string',
            'Protocol': 'http'|'https',
            'ReplaceKeyPrefixWith': 'string',
            'ReplaceKeyWith': 'string'
        }
    }
]
}
)

```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **WebsiteConfiguration** (*dict*) – [REQUIRED]
 - **ErrorDocument** (*dict*) –
 - * **Key** (*string*) – [REQUIRED] The object key name to use when a 4XX class error occurs.
 - **IndexDocument** (*dict*) –
 - * **Suffix** (*string*) – [REQUIRED] A suffix that is appended to a request that is for a directory on the website endpoint (e.g. if the suffix is index.html and you make a request to samplebucket/images/ the data that is returned will be for the object with the key name images/index.html) The suffix must not be empty and must not include a slash character.
 - **RedirectAllRequestsTo** (*dict*) –
 - * **HostName** (*string*) – [REQUIRED] Name of the host where requests will be redirected.
 - * **Protocol** (*string*) – Protocol to use (http, https) when redirecting requests. The default is the protocol that is used in the original request.
 - **RoutingRules** (*list*) –
 - * (*dict*) –
 - **Condition** (*dict*) – A container for describing a condition that must be met for the specified redirect to apply. For example, 1. If request is for pages in the /docs folder, redirect to the /documents folder. 2. If request results in HTTP error 4xx, redirect request to another host where you might process the error.
 - **HttpErrorCodeReturnedEquals** (*string*) – The HTTP error code when the redirect is applied. In the event of an error, if the error code equals this value, then the specified redirect is applied. Required when parent element Condition is specified and sibling KeyPrefixEquals is not specified. If both are specified, then both must be true for the redirect to be applied.

- **KeyPrefixEquals** (*string*) – The object key name prefix when the redirect is applied. For example, to redirect requests for ExamplePage.html, the key prefix will be ExamplePage.html. To redirect request for all pages with the prefix docs/, the key prefix will be /docs, which identifies all objects in the docs/ folder. Required when the parent element Condition is specified and sibling HttpStatusCodeReturnedEquals is not specified. If both conditions are specified, both must be true for the redirect to be applied.
- **Redirect** (*dict*) – **[REQUIRED]** Container for redirect information. You can redirect requests to another host, to another page, or with another protocol. In the event of an error, you can specify a different error code to return.
- **HostName** (*string*) – The host name to use in the redirect request.
- **HttpRedirectCode** (*string*) – The HTTP redirect code to use on the response. Not required if one of the siblings is present.
- **Protocol** (*string*) – Protocol to use (http, https) when redirecting requests. The default is the protocol that is used in the original request.
- **ReplaceKeyPrefixWith** (*string*) – The object key prefix to use in the redirect request. For example, to redirect requests for all pages with prefix docs/ (objects in the docs/ folder) to documents/, you can set a condition block with KeyPrefixEquals set to docs/ and in the Redirect set ReplaceKeyPrefixWith to /documents. Not required if one of the siblings is present. Can be present only if ReplaceKeyWith is not provided.
- **ReplaceKeyWith** (*string*) – The specific object key to use in the redirect request. For example, redirect request to error.html. Not required if one of the sibling is present. Can be present only if ReplaceKeyPrefixWith is not provided.

Returns None

put_object (***kwargs*)
Adds an object to a bucket.

Request Syntax

```
response = client.put_object(  
    ACL='private'|'public-read'|'public-read-write'|'authenticated-read'|'bucket-owner-read',  
    Body=b'bytes',  
    Bucket='string',  
    CacheControl='string',  
    ContentDisposition='string',  
    ContentEncoding='string',  
    ContentLanguage='string',  
    ContentLength=123,  
    ContentMD5='string',  
    ContentType='string',  
    Expires=datetime(2015, 1, 1),  
    GrantFullControl='string',  
    GrantRead='string',  
    GrantReadACP='string',  
    GrantWriteACP='string',  
    Key='string',
```

```

Metadata={
    'string': 'string'
},
ServerSideEncryption='AES256'|'aws:kms',
StorageClass='STANDARD'|'REDUCED_REDUNDANCY'|'STANDARD_IA',
WebsiteRedirectLocation='string',
SSECustomerAlgorithm='string',
SSECustomerKey='string',
SSEKMSKeyId='string',
RequestPayer='requester'
)

```

Parameters

- **ACL** (*string*) – The canned ACL to apply to the object.
- **Body** (*bytes*) – Object data.
- **Bucket** (*string*) – [REQUIRED]
- **CacheControl** (*string*) – Specifies caching behavior along the request/reply chain.
- **ContentDisposition** (*string*) – Specifies presentational information for the object.
- **ContentEncoding** (*string*) – Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.
- **ContentLanguage** (*string*) – The language the content is in.
- **ContentLength** (*integer*) – Size of the body in bytes. This parameter is useful when the size of the body cannot be determined automatically.
- **ContentMD5** (*string*) –
- **ContentType** (*string*) – A standard MIME type describing the format of the object data.
- **Expires** (*datetime*) – The date and time at which the object is no longer cacheable.
- **GrantFullControl** (*string*) – Gives the grantee READ, READ_ACP, and WRITE_ACP permissions on the object.
- **GrantRead** (*string*) – Allows grantee to read the object data and its metadata.
- **GrantReadACP** (*string*) – Allows grantee to read the object ACL.
- **GrantWriteACP** (*string*) – Allows grantee to write the ACL for the applicable object.
- **Key** (*string*) – [REQUIRED]
- **Metadata** (*dict*) – A map of metadata to store with the object in S3.
 - (*string*) –
 - * (*string*) –
- **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
- **StorageClass** (*string*) – The type of storage to use for the object. Defaults to 'STANDARD'.
- **WebsiteRedirectLocation** (*string*) – If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.

- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **SSEKMSKeyId** (*string*) – Specifies the AWS KMS key ID to use for object encryption. All GET and PUT requests for an object protected by AWS KMS will fail if not made via SSL or using SigV4. Documentation on configuring any of the officially supported AWS SDKs and CLI can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingAWSSDK.html#specify-signature-version>
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
    'Expiration': 'string',
    'ETag': 'string',
    'ServerSideEncryption': 'AES256'|'aws:kms',
    'VersionId': 'string',
    'SSECustomerAlgorithm': 'string',
    'SSECustomerKeyMD5': 'string',
    'SSEKMSKeyId': 'string',
    'RequestCharged': 'requester'
}
```

Response Structure

- (*dict*) –
 - **Expiration** (*string*) – If the object expiration is configured, this will contain the expiration date (expiry-date) and rule ID (rule-id). The value of rule-id is URL encoded.
 - **ETag** (*string*) – Entity tag for the uploaded object.
 - **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
 - **VersionId** (*string*) – Version of the object.
 - **SSECustomerAlgorithm** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.
 - **SSECustomerKeyMD5** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.
 - **SSEKMSKeyId** (*string*) – If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

put_object_acl (***kwargs*)

uses the acl subresource to set the access control list (ACL) permissions for an object that already exists

in a bucket

Request Syntax

```
response = client.put_object_acl(
    ACL='private'|'public-read'|'public-read-write'|'authenticated-read'|'bucket-owner-read',
    AccessControlPolicy={
        'Grants': [
            {
                'Grantee': {
                    'DisplayName': 'string',
                    'EmailAddress': 'string',
                    'ID': 'string',
                    'Type': 'CanonicalUser'|'AmazonCustomerByEmail'|'Group',
                    'URI': 'string'
                },
                'Permission': 'FULL_CONTROL'|'WRITE'|'WRITE_ACP'|'READ'|'READ_ACP'
            },
        ],
        'Owner': {
            'DisplayName': 'string',
            'ID': 'string'
        }
    },
    Bucket='string',
    GrantFullControl='string',
    GrantRead='string',
    GrantReadACP='string',
    GrantWrite='string',
    GrantWriteACP='string',
    Key='string',
    RequestPayer='requester'
)
```

Parameters

- **ACL** (*string*) – The canned ACL to apply to the object.
- **AccessControlPolicy** (*dict*) –
 - **Grants** (*list*) – A list of grants.
 - * (*dict*) –
 - **Grantee** (*dict*) –
 - **DisplayName** (*string*) – Screen name of the grantee.
 - **EmailAddress** (*string*) – Email address of the grantee.
 - **ID** (*string*) – The canonical user ID of the grantee.
 - **Type** (*string*) – [REQUIRED] Type of grantee
 - **URI** (*string*) – URI of the grantee group.
 - **Permission** (*string*) – Specifies the permission given to the grantee.
 - **Owner** (*dict*) –
 - * **DisplayName** (*string*) –
 - * **ID** (*string*) –
- **Bucket** (*string*) – [REQUIRED]
- **GrantFullControl** (*string*) – Allows grantee the read, write, read ACP, and write ACP permissions on the bucket.
- **GrantRead** (*string*) – Allows grantee to list the objects in the bucket.
- **GrantReadACP** (*string*) – Allows grantee to read the bucket ACL.
- **GrantWrite** (*string*) – Allows grantee to create, overwrite, and delete any object in the bucket.

- **GrantWriteACP** (*string*) – Allows grantee to write the ACL for the applicable bucket.
- **Key** (*string*) – [REQUIRED]
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
    'RequestCharged': 'requester'
}
```

Response Structure

- (*dict*) –
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

restore_object (***kwargs*)

Restores an archived copy of an object back into Amazon S3

Request Syntax

```
response = client.restore_object(
    Bucket='string',
    Key='string',
    VersionId='string',
    RestoreRequest={
        'Days': 123
    },
    RequestPayer='requester'
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **Key** (*string*) – [REQUIRED]
- **VersionId** (*string*) –
- **RestoreRequest** (*dict*) –
 - **Days** (*integer*) – [REQUIRED] Lifetime of the active copy in days
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
    'RequestCharged': 'requester'
}
```

Response Structure

- *(dict)* –
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

upload_file (*Filename, Bucket, Key, ExtraArgs=None, Callback=None, Config=None*)

Upload a file to an S3 object.

Usage:

```
import boto3
s3 = boto3.resource('s3')
s3.meta.client.upload_file('/tmp/hello.txt', 'mybucket', 'hello.txt')
```

Similar behavior as *S3Transfer*'s `upload_file()` method, except that parameters are capitalized. Detailed examples can be found at *S3Transfer's Usage*.

upload_part (***kwargs*)

Uploads a part in a multipart upload.

Note: After you initiate multipart upload and upload one or more parts, you must either complete or abort multipart upload in order to stop getting charged for storage of the uploaded parts. Only after you either complete or abort multipart upload, Amazon S3 frees up the parts storage and stops charging you for the parts storage.

Request Syntax

```
response = client.upload_part(
    Body=b'bytes',
    Bucket='string',
    ContentLength=123,
    ContentMD5='string',
    Key='string',
    PartNumber=123,
    UploadId='string',
    SSECustomerAlgorithm='string',
    SSECustomerKey='string',
    RequestPayer='requester'
)
```

Parameters

- **Body** (*bytes*) –
- **Bucket** (*string*) – **[REQUIRED]**
- **ContentLength** (*integer*) – Size of the body in bytes. This parameter is useful when the size of the body cannot be determined automatically.
- **ContentMD5** (*string*) –
- **Key** (*string*) – **[REQUIRED]**
- **PartNumber** (*integer*) – **[REQUIRED]** Part number of part being uploaded. This is a positive integer between 1 and 10,000.
- **UploadId** (*string*) – **[REQUIRED]** Upload ID identifying the multipart upload whose part is being uploaded.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key

must be appropriate for use with the algorithm specified in the `x-amz-server-side-encryption-customer-algorithm` header. This must be the same encryption key specified in the initiate multipart upload request.

- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
    'ServerSideEncryption': 'AES256'|'aws:kms',
    'ETag': 'string',
    'SSECustomerAlgorithm': 'string',
    'SSECustomerKeyMD5': 'string',
    'SSEKMSKeyId': 'string',
    'RequestCharged': 'requester'
}
```

Response Structure

- (*dict*) –
 - **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
 - **ETag** (*string*) – Entity tag for the uploaded object.
 - **SSECustomerAlgorithm** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.
 - **SSECustomerKeyMD5** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.
 - **SSEKMSKeyId** (*string*) – If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

upload_part_copy (***kwargs*)

Uploads a part by copying data from an existing object as data source.

Request Syntax

```
response = client.upload_part_copy(
    Bucket='string',
    CopySource='string',
    CopySourceIfMatch='string',
    CopySourceIfModifiedSince=datetime(2015, 1, 1),
    CopySourceIfNoneMatch='string',
```

```

CopySourceIfUnmodifiedSince=datetime(2015, 1, 1),
CopySourceRange='string',
Key='string',
PartNumber=123,
UploadId='string',
SSECustomerAlgorithm='string',
SSECustomerKey='string',
CopySourceSSECustomerAlgorithm='string',
CopySourceSSECustomerKey='string',
CopySourceSSECustomerKeyMD5='string',
RequestPayer='requester'
)

```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **CopySource** (*string*) – [REQUIRED] The name of the source bucket and key name of the source object, separated by a slash (/). Must be URL-encoded.
- **CopySourceIfMatch** (*string*) – Copies the object if its entity tag (ETag) matches the specified tag.
- **CopySourceIfModifiedSince** (*datetime*) – Copies the object if it has been modified since the specified time.
- **CopySourceIfNoneMatch** (*string*) – Copies the object if its entity tag (ETag) is different than the specified ETag.
- **CopySourceIfUnmodifiedSince** (*datetime*) – Copies the object if it hasn't been modified since the specified time.
- **CopySourceRange** (*string*) – The range of bytes to copy from the source object. The range value must use the form bytes=first-last, where the first and last are the zero-based byte offsets to copy. For example, bytes=0-9 indicates that you want to copy the first ten bytes of the source. You can copy a range only if the source object is greater than 5 GB.
- **Key** (*string*) – [REQUIRED]
- **PartNumber** (*integer*) – [REQUIRED] Part number of part being copied. This is a positive integer between 1 and 10,000.
- **UploadId** (*string*) – [REQUIRED] Upload ID identifying the multipart upload whose part is being copied.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header. This must be the same encryption key specified in the initiate multipart upload request.
- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **CopySourceSSECustomerAlgorithm** (*string*) – Specifies the algorithm to use when decrypting the source object (e.g., AES256).
- **CopySourceSSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use to decrypt the source object. The encryption key provided in this header must be one that was used when the source object was created.
- **CopySourceSSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 di-

gest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error.

- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
    'CopySourceVersionId': 'string',
    'CopyPartResult': {
        'ETag': 'string',
        'LastModified': datetime(2015, 1, 1)
    },
    'ServerSideEncryption': 'AES256'|'aws:kms',
    'SSECustomerAlgorithm': 'string',
    'SSECustomerKeyMD5': 'string',
    'SSEKMSKeyId': 'string',
    'RequestCharged': 'requester'
}
```

Response Structure

- (*dict*) –
 - **CopySourceVersionId** (*string*) – The version of the source object that was copied, if you have enabled versioning on the source bucket.
 - **CopyPartResult** (*dict*) –
 - * **ETag** (*string*) – Entity tag of the object.
 - * **LastModified** (*datetime*) – Date and time at which the object was uploaded.
 - **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
 - **SSECustomerAlgorithm** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.
 - **SSECustomerKeyMD5** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.
 - **SSEKMSKeyId** (*string*) – If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

Paginators

The available paginators are:

- `S3.Paginator.ListMultipartUploads`
- `S3.Paginator.ListObjectVersions`

- `S3.Paginator.ListObjects`
- `S3.Paginator.ListParts`

class `S3.Paginator.ListMultipartUploads`

```
paginator = client.get_paginator('list_multipart_uploads')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `S3.Client.list_multipart_uploads()`.

Request Syntax

```
response_iterator = paginator.paginate(
    Bucket='string',
    Delimiter='string',
    EncodingType='url',
    Prefix='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **Delimiter** (*string*) – Character you use to group keys.
- **EncodingType** (*string*) – Requests Amazon S3 to encode the object keys in the response and specifies the encoding method to use. An object key may contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.
- **Prefix** (*string*) – Lists in-progress uploads only for those keys that begin with the specified prefix.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
    'Bucket': 'string',
    'KeyMarker': 'string',
    'UploadIdMarker': 'string',
    'Prefix': 'string',
    'Delimiter': 'string',
    'MaxUploads': 123,
    'IsTruncated': True|False,
    'Uploads': [
        {
            'UploadId': 'string',
            'Key': 'string',
            'Initiated': datetime(2015, 1, 1),
            'StorageClass': 'STANDARD'|'REDUCED_REDUNDANCY'|'STANDARD_IA',
            'Owner': {
                'DisplayName': 'string',
                'ID': 'string'
            },
            'Initiator': {
                'ID': 'string',
                'DisplayName': 'string'
            }
        },
    ],
    'CommonPrefixes': [
        {
            'Prefix': 'string'
        },
    ],
    'EncodingType': 'url',
    'NextToken': 'string'
}
```

Response Structure

- *(dict)* –
 - **Bucket** (*string*) – Name of the bucket to which the multipart upload was initiated.
 - **KeyMarker** (*string*) – The key at or after which the listing began.
 - **UploadIdMarker** (*string*) – Upload ID after which listing began.
 - **Prefix** (*string*) – When a prefix is provided in the request, this field contains the specified prefix. The result contains only keys starting with the specified prefix.
 - **Delimiter** (*string*) –
 - **MaxUploads** (*integer*) – Maximum number of multipart uploads that could have been included in the response.
 - **IsTruncated** (*boolean*) – Indicates whether the returned list of multipart uploads is truncated. A value of true indicates that the list was truncated. The list can be truncated if the number of multipart uploads exceeds the limit allowed or specified by max uploads.
 - **Uploads** (*list*) –
 - * *(dict)* –
 - **UploadId** (*string*) – Upload ID that identifies the multipart upload.
 - **Key** (*string*) – Key of the object for which the multipart upload was initiated.
 - **Initiated** (*datetime*) – Date and time at which the multipart

- upload was initiated.
- **StorageClass** (*string*) – The class of storage used to store the object.
- **Owner** (*dict*) –
- **DisplayName** (*string*) –
- **ID** (*string*) –
- **Initiator** (*dict*) – Identifies who initiated the multipart upload.
- **ID** (*string*) – If the principal is an AWS account, it provides the Canonical User ID. If the principal is an IAM User, it provides a user ARN value.
- **DisplayName** (*string*) – Name of the Principal.
- **CommonPrefixes** (*list*) –
 - * (*dict*) –
 - **Prefix** (*string*) –
- **EncodingType** (*string*) – Encoding type used by Amazon S3 to encode object keys in the response.
- **NextToken** (*string*) –
 - A token to resume pagination.

class S3.Paginator.ListObjectVersions

```
paginator = client.get_paginator('list_object_versions')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `S3.Client.list_object_versions()`.

Request Syntax

```
response_iterator = paginator.paginate(
    Bucket='string',
    Delimiter='string',
    EncodingType='url',
    Prefix='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **Delimiter** (*string*) – A delimiter is a character you use to group keys.
- **EncodingType** (*string*) – Requests Amazon S3 to encode the object keys in the response and specifies the encoding method to use. An object key may contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.
- **Prefix** (*string*) – Limits the response to keys that begin with the specified prefix.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

– **PageSize** (*integer*) –

The size of each page.

– **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'IsTruncated': True|False,
  'KeyMarker': 'string',
  'VersionIdMarker': 'string',
  'Versions': [
    {
      'ETag': 'string',
      'Size': 123,
      'StorageClass': 'STANDARD',
      'Key': 'string',
      'VersionId': 'string',
      'IsLatest': True|False,
      'LastModified': datetime(2015, 1, 1),
      'Owner': {
        'DisplayName': 'string',
        'ID': 'string'
      }
    },
  ],
  'DeleteMarkers': [
    {
      'Owner': {
        'DisplayName': 'string',
        'ID': 'string'
      },
      'Key': 'string',
      'VersionId': 'string',
      'IsLatest': True|False,
      'LastModified': datetime(2015, 1, 1)
    },
  ],
  'Name': 'string',
  'Prefix': 'string',
  'Delimiter': 'string',
  'MaxKeys': 123,
  'CommonPrefixes': [
    {
      'Prefix': 'string'
    },
  ],
  'EncodingType': 'url',
  'NextToken': 'string'
}
```

Response Structure

- *(dict)* –
 - **IsTruncated** (*boolean*) – A flag that indicates whether or not Amazon S3 returned all of the results that satisfied the search criteria. If your results were truncated, you can make a follow-up paginated request using the `NextKeyMarker` and `NextVersionIdMarker` response parameters as a starting place in another request to return the rest of the results.
 - **KeyMarker** (*string*) – Marks the last Key returned in a truncated response.
 - **VersionIdMarker** (*string*) –
 - **Versions** (*list*) –
 - * *(dict)* –
 - **ETag** (*string*) –
 - **Size** (*integer*) – Size in bytes of the object.
 - **StorageClass** (*string*) – The class of storage used to store the object.
 - **Key** (*string*) – The object key.
 - **VersionId** (*string*) – Version ID of an object.
 - **IsLatest** (*boolean*) – Specifies whether the object is (true) or is not (false) the latest version of an object.
 - **LastModified** (*datetime*) – Date and time the object was last modified.
 - **Owner** (*dict*) –
 - **DisplayName** (*string*) –
 - **ID** (*string*) –
 - **DeleteMarkers** (*list*) –
 - * *(dict)* –
 - **Owner** (*dict*) –
 - **DisplayName** (*string*) –
 - **ID** (*string*) –
 - **Key** (*string*) – The object key.
 - **VersionId** (*string*) – Version ID of an object.
 - **IsLatest** (*boolean*) – Specifies whether the object is (true) or is not (false) the latest version of an object.
 - **LastModified** (*datetime*) – Date and time the object was last modified.
 - **Name** (*string*) –
 - **Prefix** (*string*) –
 - **Delimiter** (*string*) –
 - **MaxKeys** (*integer*) –
 - **CommonPrefixes** (*list*) –
 - * *(dict)* –
 - **Prefix** (*string*) –
 - **EncodingType** (*string*) – Encoding type used by Amazon S3 to encode object keys in the response.
 - **NextToken** (*string*) –

A token to resume pagination.

```
class S3.Paginator.ListObjects
```

```
paginator = client.get_paginator('list_objects')
```

```
paginate (**kwargs)
```

Creates an iterator that will paginate through responses from `S3.Client.list_objects()`.

Request Syntax

```
response_iterator = paginator.paginate(
    Bucket='string',
    Delimiter='string',
    EncodingType='url',
    Prefix='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **Delimiter** (*string*) – A delimiter is a character you use to group keys.
- **EncodingType** (*string*) – Requests Amazon S3 to encode the object keys in the response and specifies the encoding method to use. An object key may contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.
- **Prefix** (*string*) – Limits the response to keys that begin with the specified prefix.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
    'IsTruncated': True|False,
    'Marker': 'string',
    'NextMarker': 'string',
    'Contents': [
        {
            'Key': 'string',
            'LastModified': datetime(2015, 1, 1),
            'ETag': 'string',
            'Size': 123,
            'StorageClass': 'STANDARD'|'REDUCED_REDUNDANCY'|'GLACIER',
            'Owner': {
                'DisplayName': 'string',
                'ID': 'string'
            }
        }
    ]
}
```

```

    },
    ],
    'Name': 'string',
    'Prefix': 'string',
    'Delimiter': 'string',
    'MaxKeys': 123,
    'CommonPrefixes': [
        {
            'Prefix': 'string'
        },
    ],
    'EncodingType': 'url',
    'NextToken': 'string'
}

```

Response Structure

- *(dict)* –
 - **IsTruncated** (*boolean*) – A flag that indicates whether or not Amazon S3 returned all of the results that satisfied the search criteria.
 - **Marker** (*string*) –
 - **NextMarker** (*string*) – When response is truncated (the IsTruncated element value in the response is true), you can use the key name in this field as marker in the subsequent request to get next set of objects. Amazon S3 lists objects in alphabetical order Note: This element is returned only if you have delimiter request parameter specified. If response does not include the NextMaker and it is truncated, you can use the value of the last Key in the response as the marker in the subsequent request to get the next set of object keys.
 - **Contents** (*list*) –
 - * *(dict)* –
 - **Key** (*string*) –
 - **LastModified** (*datetime*) –
 - **ETag** (*string*) –
 - **Size** (*integer*) –
 - **StorageClass** (*string*) – The class of storage used to store the object.
 - **Owner** (*dict*) –
 - **DisplayName** (*string*) –
 - **ID** (*string*) –
 - **Name** (*string*) –
 - **Prefix** (*string*) –
 - **Delimiter** (*string*) –
 - **MaxKeys** (*integer*) –
 - **CommonPrefixes** (*list*) –
 - * *(dict)* –
 - **Prefix** (*string*) –
 - **EncodingType** (*string*) – Encoding type used by Amazon S3 to encode object keys in the response.
 - **NextToken** (*string*) –

A token to resume pagination.

```
class S3.Paginator.ListParts
```

```
paginator = client.get_paginator('list_parts')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `S3.Client.list_parts()`.

Request Syntax

```
response_iterator = paginator.paginate(
    Bucket='string',
    Key='string',
    UploadId='string',
    RequestPayer='requester',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **Key** (*string*) – [REQUIRED]
- **UploadId** (*string*) – [REQUIRED] Upload ID identifying the multipart upload whose parts are being listed.
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns**Response Syntax**

```
{
    'Bucket': 'string',
    'Key': 'string',
    'UploadId': 'string',
    'PartNumberMarker': 123,
    'MaxParts': 123,
    'IsTruncated': True|False,
    'Parts': [
        {
```

```

        'PartNumber': 123,
        'LastModified': datetime(2015, 1, 1),
        'ETag': 'string',
        'Size': 123
    },
],
'Initiator': {
    'ID': 'string',
    'DisplayName': 'string'
},
'Owner': {
    'DisplayName': 'string',
    'ID': 'string'
},
'StorageClass': 'STANDARD' | 'REDUCED_REDUNDANCY' | 'STANDARD_IA',
'RequestCharged': 'requester',
'NextToken': 'string'
}

```

Response Structure

- *(dict)* –
 - **Bucket** (*string*) – Name of the bucket to which the multipart upload was initiated.
 - **Key** (*string*) – Object key for which the multipart upload was initiated.
 - **UploadId** (*string*) – Upload ID identifying the multipart upload whose parts are being listed.
 - **PartNumberMarker** (*integer*) – Part number after which listing begins.
 - **MaxParts** (*integer*) – Maximum number of parts that were allowed in the response.
 - **IsTruncated** (*boolean*) – Indicates whether the returned list of parts is truncated.
 - **Parts** (*list*) –
 - * *(dict)* –
 - **PartNumber** (*integer*) – Part number identifying the part. This is a positive integer between 1 and 10,000.
 - **LastModified** (*datetime*) – Date and time at which the part was uploaded.
 - **ETag** (*string*) – Entity tag returned when the part was uploaded.
 - **Size** (*integer*) – Size of the uploaded part data.
 - **Initiator** (*dict*) – Identifies who initiated the multipart upload.
 - * **ID** (*string*) – If the principal is an AWS account, it provides the Canonical User ID. If the principal is an IAM User, it provides a user ARN value.
 - * **DisplayName** (*string*) – Name of the Principal.
 - **Owner** (*dict*) –
 - * **DisplayName** (*string*) –
 - * **ID** (*string*) –
 - **StorageClass** (*string*) – The class of storage used to store the object.
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.
 - **NextToken** (*string*) –
 - A token to resume pagination.

Waiters

The available waiters are:

- `S3.Waiter.BucketExists`
- `S3.Waiter.BucketNotExists`
- `S3.Waiter.ObjectExists`
- `S3.Waiter.ObjectNotExists`

class `S3.Waiter.BucketExists`

```
waiter = client.get_waiter('bucket_exists')
```

wait (***kwargs*)

Polls `S3.Client.head_bucket()` every 5 seconds until a successful state is reached. An error is returned after 20 failed checks.

Request Syntax

```
waiter.wait(  
    Bucket='string'  
)
```

Parameters `Bucket` (*string*) – [REQUIRED]

Returns None

class `S3.Waiter.BucketNotExists`

```
waiter = client.get_waiter('bucket_not_exists')
```

wait (***kwargs*)

Polls `S3.Client.head_bucket()` every 5 seconds until a successful state is reached. An error is returned after 20 failed checks.

Request Syntax

```
waiter.wait(  
    Bucket='string'  
)
```

Parameters `Bucket` (*string*) – [REQUIRED]

Returns None

class `S3.Waiter.ObjectExists`

```
waiter = client.get_waiter('object_exists')
```

wait (***kwargs*)

Polls `S3.Client.head_object()` every 5 seconds until a successful state is reached. An error is returned after 20 failed checks.

Request Syntax


```
waiter.wait(
    Bucket='string',
    IfMatch='string',
    IfModifiedSince=datetime(2015, 1, 1),
    IfNoneMatch='string',
    IfUnmodifiedSince=datetime(2015, 1, 1),
    Key='string',
    Range='string',
    VersionId='string',
    SSECustomerAlgorithm='string',
    SSECustomerKey='string',
    RequestPayer='requester'
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **IfMatch** (*string*) – Return the object only if its entity tag (ETag) is the same as the one specified, otherwise return a 412 (precondition failed).
- **IfModifiedSince** (*datetime*) – Return the object only if it has been modified since the specified time, otherwise return a 304 (not modified).
- **IfNoneMatch** (*string*) – Return the object only if its entity tag (ETag) is different from the one specified, otherwise return a 304 (not modified).
- **IfUnmodifiedSince** (*datetime*) – Return the object only if it has not been modified since the specified time, otherwise return a 412 (precondition failed).
- **Key** (*string*) – [REQUIRED]
- **Range** (*string*) – Downloads the specified range bytes of an object. For more information about the HTTP Range header, go to <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35>.
- **VersionId** (*string*) – VersionId used to reference a specific version of the object.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.
- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Returns None

class S3.Waiter.ObjectNotExists

```
waiter = client.get_waiter('object_not_exists')
```

wait (***kwargs*)

Polls `S3.Client.head_object()` every 5 seconds until a successful state is reached. An error is

returned after 20 failed checks.

Request Syntax

```
waiter.wait(  
    Bucket='string',  
    IfMatch='string',  
    IfModifiedSince=datetime(2015, 1, 1),  
    IfNoneMatch='string',  
    IfUnmodifiedSince=datetime(2015, 1, 1),  
    Key='string',  
    Range='string',  
    VersionId='string',  
    SSECustomerAlgorithm='string',  
    SSECustomerKey='string',  
    RequestPayer='requester'  
)
```

Parameters

- **Bucket** (*string*) – [REQUIRED]
- **IfMatch** (*string*) – Return the object only if its entity tag (ETag) is the same as the one specified, otherwise return a 412 (precondition failed).
- **IfModifiedSince** (*datetime*) – Return the object only if it has been modified since the specified time, otherwise return a 304 (not modified).
- **IfNoneMatch** (*string*) – Return the object only if its entity tag (ETag) is different from the one specified, otherwise return a 304 (not modified).
- **IfUnmodifiedSince** (*datetime*) – Return the object only if it has not been modified since the specified time, otherwise return a 412 (precondition failed).
- **Key** (*string*) – [REQUIRED]
- **Range** (*string*) – Downloads the specified range bytes of an object. For more information about the HTTP Range header, go to <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35>.
- **VersionId** (*string*) – VersionId used to reference a specific version of the object.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.
- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Returns None

Service Resource

class S3.ServiceResource

A resource representing Amazon Simple Storage Service (S3):

```
import boto3

s3 = boto3.resource('s3')
```

These are the resource's available actions:

- `create_bucket()`

These are the resource's available sub-resources:

- `Bucket()`
- `BucketAcl()`
- `BucketCors()`
- `BucketLifecycle()`
- `BucketLogging()`
- `BucketNotification()`
- `BucketPolicy()`
- `BucketRequestPayment()`
- `BucketTagging()`
- `BucketVersioning()`
- `BucketWebsite()`
- `MultipartUpload()`
- `MultipartUploadPart()`
- `Object()`
- `ObjectAcl()`
- `ObjectSummary()`
- `ObjectVersion()`

These are the resource's available collections:

- `buckets`

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

`create_bucket(**kwargs)`

Creates a new bucket.

Request Syntax

```
bucket = s3.create_bucket(
    ACL='private'|'public-read'|'public-read-write'|'authenticated-read',
    Bucket='string',
    CreateBucketConfiguration={
        'LocationConstraint': 'EU'|'eu-west-1'|'us-west-1'|'us-west-2'|'ap-southeast-1'|'ap-
    },
    GrantFullControl='string',
    GrantRead='string',
    GrantReadACP='string',
    GrantWrite='string',
    GrantWriteACP='string'
)
```

Parameters

- **ACL** (*string*) – The canned ACL to apply to the bucket.

- **Bucket** (*string*) – [REQUIRED]
- **CreateBucketConfiguration** (*dict*) –
 - **LocationConstraint** (*string*) – Specifies the region where the bucket will be created. If you don't specify a region, the bucket will be created in US Standard.
- **GrantFullControl** (*string*) – Allows grantee the read, write, read ACP, and write ACP permissions on the bucket.
- **GrantRead** (*string*) – Allows grantee to list the objects in the bucket.
- **GrantReadACP** (*string*) – Allows grantee to read the bucket ACL.
- **GrantWrite** (*string*) – Allows grantee to create, overwrite, and delete any object in the bucket.
- **GrantWriteACP** (*string*) – Allows grantee to write the ACL for the applicable bucket.

Return type `s3.Bucket`

Returns Bucket resource

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Bucket (*name*)

Creates a Bucket resource.:

```
bucket = s3.Bucket('name')
```

Parameters **name** (*string*) – The Bucket's name identifier. This **must** be set.

Return type `S3.Bucket`

Returns A Bucket resource

BucketAcl (*bucket_name*)

Creates a BucketAcl resource.:

```
bucket_acl = s3.BucketAcl('bucket_name')
```

Parameters **bucket_name** (*string*) – The BucketAcl's bucket_name identifier. This **must** be set.

Return type `S3.BucketAcl`

Returns A BucketAcl resource

BucketCors (*bucket_name*)

Creates a BucketCors resource.:

```
bucket_cors = s3.BucketCors('bucket_name')
```

Parameters **bucket_name** (*string*) – The BucketCors's bucket_name identifier. This **must** be set.

Return type `S3.BucketCors`

Returns A BucketCors resource

BucketLifecycle (*bucket_name*)

Creates a BucketLifecycle resource.:

```
bucket_lifecycle = s3.BucketLifecycle('bucket_name')
```

Parameters **bucket_name** (*string*) – The BucketLifecycle's bucket_name identifier. This **must** be set.

Return type `S3.BucketLifecycle`

Returns A BucketLifecycle resource

BucketLogging (*bucket_name*)

Creates a BucketLogging resource.:

```
bucket_logging = s3.BucketLogging('bucket_name')
```

Parameters **bucket_name** (*string*) – The BucketLogging’s bucket_name identifier. This **must** be set.

Return type `S3.BucketLogging`

Returns A BucketLogging resource

BucketNotification (*bucket_name*)

Creates a BucketNotification resource.:

```
bucket_notification = s3.BucketNotification('bucket_name')
```

Parameters **bucket_name** (*string*) – The BucketNotification’s bucket_name identifier. This **must** be set.

Return type `S3.BucketNotification`

Returns A BucketNotification resource

BucketPolicy (*bucket_name*)

Creates a BucketPolicy resource.:

```
bucket_policy = s3.BucketPolicy('bucket_name')
```

Parameters **bucket_name** (*string*) – The BucketPolicy’s bucket_name identifier. This **must** be set.

Return type `S3.BucketPolicy`

Returns A BucketPolicy resource

BucketRequestPayment (*bucket_name*)

Creates a BucketRequestPayment resource.:

```
bucket_request_payment = s3.BucketRequestPayment('bucket_name')
```

Parameters **bucket_name** (*string*) – The BucketRequestPayment’s bucket_name identifier. This **must** be set.

Return type `S3.BucketRequestPayment`

Returns A BucketRequestPayment resource

BucketTagging (*bucket_name*)

Creates a BucketTagging resource.:

```
bucket_tagging = s3.BucketTagging('bucket_name')
```

Parameters **bucket_name** (*string*) – The BucketTagging’s bucket_name identifier. This **must** be set.

Return type `S3.BucketTagging`

Returns A BucketTagging resource

BucketVersioning (*bucket_name*)

Creates a BucketVersioning resource.:

```
bucket_versioning = s3.BucketVersioning('bucket_name')
```

Parameters `bucket_name` (*string*) – The BucketVersioning’s `bucket_name` identifier. This **must** be set.

Return type `S3.BucketVersioning`

Returns A BucketVersioning resource

BucketWebsite (*bucket_name*)

Creates a BucketWebsite resource.:

```
bucket_website = s3.BucketWebsite('bucket_name')
```

Parameters `bucket_name` (*string*) – The BucketWebsite’s `bucket_name` identifier. This **must** be set.

Return type `S3.BucketWebsite`

Returns A BucketWebsite resource

MultipartUpload (*bucket_name, object_key, id*)

Creates a MultipartUpload resource.:

```
multipart_upload = s3.MultipartUpload('bucket_name', 'object_key', 'id')
```

Parameters

- **bucket_name** (*string*) – The MultipartUpload’s `bucket_name` identifier. This **must** be set.
- **object_key** (*string*) – The MultipartUpload’s `object_key` identifier. This **must** be set.
- **id** (*string*) – The MultipartUpload’s `id` identifier. This **must** be set.

Return type `S3.MultipartUpload`

Returns A MultipartUpload resource

MultipartUploadPart (*bucket_name, object_key, multipart_upload_id, part_number*)

Creates a MultipartUploadPart resource.:

```
multipart_upload_part = s3.MultipartUploadPart('bucket_name', 'object_key', 'multipart_upload_id', 'part_number')
```

Parameters

- **bucket_name** (*string*) – The MultipartUploadPart’s `bucket_name` identifier. This **must** be set.
- **object_key** (*string*) – The MultipartUploadPart’s `object_key` identifier. This **must** be set.
- **multipart_upload_id** (*string*) – The MultipartUploadPart’s `multipart_upload_id` identifier. This **must** be set.
- **part_number** (*string*) – The MultipartUploadPart’s `part_number` identifier. This **must** be set.

Return type `S3.MultipartUploadPart`

Returns A MultipartUploadPart resource

Object (*bucket_name, key*)

Creates a Object resource.:

```
object = s3.Object('bucket_name', 'key')
```

Parameters

- **bucket_name** (*string*) – The Object’s `bucket_name` identifier. This **must** be set.
- **key** (*string*) – The Object’s `key` identifier. This **must** be set.

Return type `S3.Object`

Returns A Object resource

ObjectAcl (*bucket_name, object_key*)

Creates a ObjectAcl resource.:

```
object_acl = s3.ObjectAcl('bucket_name', 'object_key')
```

Parameters

- **bucket_name** (*string*) – The ObjectAcl’s bucket_name identifier. This **must** be set.
- **object_key** (*string*) – The ObjectAcl’s object_key identifier. This **must** be set.

Return type `S3.ObjectAcl`

Returns A ObjectAcl resource

ObjectSummary (*bucket_name, key*)

Creates a ObjectSummary resource.:

```
object_summary = s3.ObjectSummary('bucket_name', 'key')
```

Parameters

- **bucket_name** (*string*) – The ObjectSummary’s bucket_name identifier. This **must** be set.
- **key** (*string*) – The ObjectSummary’s key identifier. This **must** be set.

Return type `S3.ObjectSummary`

Returns A ObjectSummary resource

ObjectVersion (*bucket_name, object_key, id*)

Creates a ObjectVersion resource.:

```
object_version = s3.ObjectVersion('bucket_name', 'object_key', 'id')
```

Parameters

- **bucket_name** (*string*) – The ObjectVersion’s bucket_name identifier. This **must** be set.
- **object_key** (*string*) – The ObjectVersion’s object_key identifier. This **must** be set.
- **id** (*string*) – The ObjectVersion’s id identifier. This **must** be set.

Return type `S3.ObjectVersion`

Returns A ObjectVersion resource

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

buckets

all()

Creates an iterable of all Bucket resources in the collection.

Request Syntax

```
bucket_iterator = s3.buckets.all()
```

Return type `list(s3.Bucket)`

Returns A list of Bucket resources

filter()

Creates an iterable of all Bucket resources in the collection filtered by kwargs passed to method.

Request Syntax

```
bucket_iterator = s3.buckets.filter()
```

Return type list(s3.Bucket)

Returns A list of Bucket resources

limit(kwargs)**

Creates an iterable up to a specified amount of Bucket resources in the collection.

Request Syntax

```
bucket_iterator = s3.buckets.limit()
```

Return type list(s3.Bucket)

Returns A list of Bucket resources

page_size(kwargs)**

Creates an iterable of all Bucket resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
bucket_iterator = s3.buckets.page_size()
```

Return type list(s3.Bucket)

Returns A list of Bucket resources

Bucket

class S3.Bucket (*name*)

A resource representing an Amazon Simple Storage Service (S3) Bucket:

```
import boto3
```

```
s3 = boto3.resource('s3')
bucket = s3.Bucket('name')
```

Parameters *name* (*string*) – The Bucket’s name identifier. This **must** be set.

These are the resource’s available identifiers:

- `name`

These are the resource’s available attributes:

- `creation_date`

These are the resource’s available actions:

- `create()`
- `delete()`
- `delete_objects()`
- `download_file()`
- `load()`
- `put_object()`
- `upload_file()`

These are the resource’s available sub-resources:

- `Acl()`

- `Cors()`
- `Lifecycle()`
- `Logging()`
- `Notification()`
- `Object()`
- `Policy()`
- `RequestPayment()`
- `Tagging()`
- `Versioning()`
- `Website()`

These are the resource's available collections:

- `multipart_uploads`
- `object_versions`
- `objects`

These are the resource's available waiters:

- `wait_until_exists()`
- `wait_until_not_exists()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

name

(*string*) The Bucket's name identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

creation_date

(*datetime*) Date the bucket was created.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

download_file (*Key, Filename, ExtraArgs=None, Callback=None, Config=None*)

load (**args, **kwargs*)

upload_file (*Filename, Key, ExtraArgs=None, Callback=None, Config=None*)

create (***kwargs*)

Creates a new bucket.

Request Syntax

```
response = bucket.create(
    ACL='private'|'public-read'|'public-read-write'|'authenticated-read',
    CreateBucketConfiguration={
        'LocationConstraint': 'EU'|'eu-west-1'|'us-west-1'|'us-west-2'|'ap-southeast-1'|'ap-
    },
    GrantFullControl='string',
    GrantRead='string',
    GrantReadACP='string',
    GrantWrite='string',
```

```
GrantWriteACP='string'  
)
```

Parameters

- **ACL** (*string*) – The canned ACL to apply to the bucket.
- **CreateBucketConfiguration** (*dict*) –
 - **LocationConstraint** (*string*) – Specifies the region where the bucket will be created. If you don't specify a region, the bucket will be created in US Standard.
- **GrantFullControl** (*string*) – Allows grantee the read, write, read ACP, and write ACP permissions on the bucket.
- **GrantRead** (*string*) – Allows grantee to list the objects in the bucket.
- **GrantReadACP** (*string*) – Allows grantee to read the bucket ACL.
- **GrantWrite** (*string*) – Allows grantee to create, overwrite, and delete any object in the bucket.
- **GrantWriteACP** (*string*) – Allows grantee to write the ACL for the applicable bucket.

Return type dict**Returns****Response Syntax**

```
{  
    'Location': 'string'  
}
```

Response Structure

- (*dict*) –
 - **Location** (*string*) –

delete()

Deletes the bucket. All objects (including all object versions and Delete Markers) in the bucket must be deleted before the bucket itself can be deleted.

Request Syntax

```
response = bucket.delete()
```

Returns None**delete_objects(**kwargs)**

This operation enables you to delete multiple objects from a bucket using a single HTTP request. You may specify up to 1000 keys.

Request Syntax

```
response = bucket.delete_objects(  
    Delete={  
        'Objects': [  
            {  
                'Key': 'string',  
                'VersionId': 'string'  
            },  
        ],  
        'Quiet': True|False  
    },  
    MFA='string',
```

```
RequestPayer='requester'
)
```

Parameters

- **Delete** (*dict*) – [REQUIRED]
 - **Objects** (*list*) – [REQUIRED]
 - * (*dict*) –
 - **Key** (*string*) – [REQUIRED] Key name of the object to delete.
 - **VersionId** (*string*) – VersionId for the specific version of the object to delete.
 - **Quiet** (*boolean*) – Element to enable quiet mode for the request. When you add this element, you must set its value to true.
- **MFA** (*string*) – The concatenation of the authentication device’s serial number, a space, and the value that is displayed on your authentication device.
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
  'Deleted': [
    {
      'Key': 'string',
      'VersionId': 'string',
      'DeleteMarker': True|False,
      'DeleteMarkerVersionId': 'string'
    },
  ],
  'RequestCharged': 'requester',
  'Errors': [
    {
      'Key': 'string',
      'VersionId': 'string',
      'Code': 'string',
      'Message': 'string'
    },
  ],
}
```

Response Structure

- (*dict*) –
 - **Deleted** (*list*) –
 - * (*dict*) –
 - **Key** (*string*) –
 - **VersionId** (*string*) –
 - **DeleteMarker** (*boolean*) –
 - **DeleteMarkerVersionId** (*string*) –
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.
 - **Errors** (*list*) –

```
* (dict) –  
    · Key (string) –  
    · VersionId (string) –  
    · Code (string) –  
    · Message (string) –
```

Upload a file to an S3 object.

Usage:

```
import boto3  
s3 = boto3.resource('s3')  
s3.Bucket('mybucket').upload_file('/tmp/hello.txt', 'hello.txt')
```

Similar behavior as `S3Transfer`'s `upload_file()` method, except that parameters are capitalized. Detailed examples can be found at *S3Transfer's Usage*.

put_object (**kwargs)

Adds an object to a bucket.

Request Syntax

```
object = bucket.put_object(  
    ACL='private'|'public-read'|'public-read-write'|'authenticated-read'|'bucket-owner-override',  
    Body=b'bytes',  
    CacheControl='string',  
    ContentDisposition='string',  
    ContentEncoding='string',  
    ContentLanguage='string',  
    ContentLength=123,  
    ContentMD5='string',  
    ContentType='string',  
    Expires=datetime(2015, 1, 1),  
    GrantFullControl='string',  
    GrantRead='string',  
    GrantReadACP='string',  
    GrantWriteACP='string',  
    Key='string',  
    Metadata={  
        'string': 'string'  
    },  
    ServerSideEncryption='AES256'|'aws:kms',  
    StorageClass='STANDARD'|'REDUCED_REDUNDANCY'|'STANDARD_IA',  
    WebsiteRedirectLocation='string',  
    SSECustomerAlgorithm='string',  
    SSECustomerKey='string',  
    SSEKMSKeyId='string',  
    RequestPayer='requester'  
)
```

Parameters

- **ACL** (string) – The canned ACL to apply to the object.
- **Body** (bytes) – Object data.
- **CacheControl** (string) – Specifies caching behavior along the request/reply chain.
- **ContentDisposition** (string) – Specifies presentational information for the object.
- **ContentEncoding** (string) – Specifies what content encodings have been applied to the object and thus

what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.

- **ContentLanguage** (*string*) – The language the content is in.
- **ContentLength** (*integer*) – Size of the body in bytes. This parameter is useful when the size of the body cannot be determined automatically.
- **ContentMD5** (*string*) –
- **ContentType** (*string*) – A standard MIME type describing the format of the object data.
- **Expires** (*datetime*) – The date and time at which the object is no longer cacheable.
- **GrantFullControl** (*string*) – Gives the grantee READ, READ_ACP, and WRITE_ACP permissions on the object.
- **GrantRead** (*string*) – Allows grantee to read the object data and its metadata.
- **GrantReadACP** (*string*) – Allows grantee to read the object ACL.
- **GrantWriteACP** (*string*) – Allows grantee to write the ACL for the applicable object.
- **Key** (*string*) – [REQUIRED]
- **Metadata** (*dict*) – A map of metadata to store with the object in S3.
 - (*string*) –
 - * (*string*) –
- **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
- **StorageClass** (*string*) – The type of storage to use for the object. Defaults to 'STANDARD'.
- **WebsiteRedirectLocation** (*string*) – If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.
- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required

- **SSEKMSKeyId** (*string*) – Specifies the AWS KMS key ID to use for object encryption. All GET and PUT requests for an object protected by AWS KMS will fail if not made via SSL or using SigV4. Documentation on configuring any of the officially supported AWS SDKs and CLI can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingAWSSDK.html#specify-signature-version>
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at [http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.h](http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html)

Return type `s3.Object`

Returns Object resource

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the *[Resources Introduction Guide](#)*.

Acl()

Creates a BucketAcl resource.:

```
bucket_acl = bucket.Acl()
```

Return type `S3.BucketAcl`

Returns A BucketAcl resource

Cors()

Creates a BucketCors resource.:

```
bucket_cors = bucket.Cors()
```

Return type `S3.BucketCors`

Returns A BucketCors resource

Lifecycle()

Creates a BucketLifecycle resource.:

```
bucket_lifecycle = bucket.Lifecycle()
```

Return type `S3.BucketLifecycle`

Returns A BucketLifecycle resource

Logging()

Creates a BucketLogging resource.:

```
bucket_logging = bucket.Logging()
```

Return type `S3.BucketLogging`

Returns A BucketLogging resource

Notification()

Creates a BucketNotification resource.:

```
bucket_notification = bucket.Notification()
```

Return type `S3.BucketNotification`

Returns A BucketNotification resource

Object (*key*)

Creates a Object resource.:

```
object = bucket.Object('key')
```

Parameters *key* (*string*) – The Object’s key identifier. This **must** be set.

Return type `S3.Object`

Returns A Object resource

Policy ()

Creates a BucketPolicy resource.:

```
bucket_policy = bucket.Policy()
```

Return type `S3.BucketPolicy`

Returns A BucketPolicy resource

RequestPayment ()

Creates a BucketRequestPayment resource.:

```
bucket_request_payment = bucket.RequestPayment()
```

Return type `S3.BucketRequestPayment`

Returns A BucketRequestPayment resource

Tagging ()

Creates a BucketTagging resource.:

```
bucket_tagging = bucket.Tagging()
```

Return type `S3.BucketTagging`

Returns A BucketTagging resource

Versioning ()

Creates a BucketVersioning resource.:

```
bucket_versioning = bucket.Versioning()
```

Return type `S3.BucketVersioning`

Returns A BucketVersioning resource

Website ()

Creates a BucketWebsite resource.:

```
bucket_website = bucket.Website()
```

Return type `S3.BucketWebsite`

Returns A BucketWebsite resource

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

multipart_uploads**all()**

Creates an iterable of all MultipartUpload resources in the collection.

Request Syntax

```
multipart_upload_iterator = bucket.multipart_uploads.all()
```

Return type list(s3.MultipartUpload)

Returns A list of MultipartUpload resources

filter(kwargs)**

Creates an iterable of all MultipartUpload resources in the collection filtered by kwargs passed to method.

Request Syntax

```
multipart_upload_iterator = bucket.multipart_uploads.filter(
    Delimiter='string',
    EncodingType='url',
    KeyMarker='string',
    MaxUploads=123,
    Prefix='string',
    UploadIdMarker='string'
)
```

Parameters

- **Delimiter** (*string*) – Character you use to group keys.
- **EncodingType** (*string*) – Requests Amazon S3 to encode the object keys in the response and specifies the encoding method to use. An object key may contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.
- **KeyMarker** (*string*) – Together with upload-id-marker, this parameter specifies the multipart upload after which listing should begin.
- **MaxUploads** (*integer*) – Sets the maximum number of multipart uploads, from 1 to 1,000, to return in the response body. 1,000 is the maximum number of uploads that can be returned in a response.
- **Prefix** (*string*) – Lists in-progress uploads only for those keys that begin with the specified prefix.
- **UploadIdMarker** (*string*) – Together with key-marker, specifies the multipart upload after which listing should begin. If key-marker is not specified, the upload-id-marker parameter is ignored.

Return type list(s3.MultipartUpload)

Returns A list of MultipartUpload resources

limit(kwargs)**

Creates an iterable up to a specified amount of MultipartUpload resources in the collection.

Request Syntax

```
multipart_upload_iterator = bucket.multipart_uploads.limit(
    count=123
)
```


Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(s3.MultipartUpload)`

Returns A list of MultipartUpload resources

page_size (***kwargs*)

Creates an iterable of all MultipartUpload resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
multipart_upload_iterator = bucket.multipart_uploads.page_size(
    count=123
)
```

Parameters `count` (*integer*) – The number of items returned by each service call

Return type `list(s3.MultipartUpload)`

Returns A list of MultipartUpload resources

object_versions

all ()

Creates an iterable of all ObjectVersion resources in the collection.

Request Syntax

```
object_version_iterator = bucket.object_versions.all()
```

Return type `list(s3.ObjectVersion)`

Returns A list of ObjectVersion resources

delete (***kwargs*)

This operation enables you to delete multiple objects from a bucket using a single HTTP request. You may specify up to 1000 keys.

Request Syntax

```
response = bucket.object_versions.delete(
    MFA='string',
    RequestPayer='requester'
)
```

Parameters

- **MFA** (*string*) – The concatenation of the authentication device’s serial number, a space, and the value that is displayed on your authentication device.
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type `dict`

Returns

Response Syntax

```
{
  'Deleted': [
    {
      'Key': 'string',
      'VersionId': 'string',
      'DeleteMarker': True|False,
      'DeleteMarkerVersionId': 'string'
    },
  ],
  'RequestCharged': 'requester',
  'Errors': [
    {
      'Key': 'string',
      'VersionId': 'string',
      'Code': 'string',
      'Message': 'string'
    },
  ],
}
```

Response Structure

- *(dict)* –
 - **Deleted** (*list*) –
 - * *(dict)* –
 - **Key** (*string*) –
 - **VersionId** (*string*) –
 - **DeleteMarker** (*boolean*) –
 - **DeleteMarkerVersionId** (*string*) –
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.
 - **Errors** (*list*) –
 - * *(dict)* –
 - **Key** (*string*) –
 - **VersionId** (*string*) –
 - **Code** (*string*) –
 - **Message** (*string*) –

filter (***kwargs*)

Creates an iterable of all ObjectVersion resources in the collection filtered by kwargs passed to method.

Request Syntax

```
object_version_iterator = bucket.object_versions.filter(
    Delimiter='string',
    EncodingType='url',
    KeyMarker='string',
    MaxKeys=123,
    Prefix='string',
    VersionIdMarker='string'
)
```

Parameters

- **Delimiter** (*string*) – A delimiter is a character you use to group keys.
- **EncodingType** (*string*) – Requests Amazon S3 to encode the object keys in the response and specifies the encoding method to use. An object key may contain any Unicode character; however, XML 1.0 parser cannot

parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.

- **KeyMarker** (*string*) – Specifies the key to start with when listing objects in a bucket.
- **MaxKeys** (*integer*) – Sets the maximum number of keys returned in the response. The response might contain fewer keys but will never contain more.
- **Prefix** (*string*) – Limits the response to keys that begin with the specified prefix.
- **VersionIdMarker** (*string*) – Specifies the object version you want to start listing from.

Return type `list(s3.ObjectVersion)`

Returns A list of `ObjectVersion` resources

limit (***kwargs*)

Creates an iterable up to a specified amount of `ObjectVersion` resources in the collection.

Request Syntax

```
object_version_iterator = bucket.object_versions.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(s3.ObjectVersion)`

Returns A list of `ObjectVersion` resources

page_size (***kwargs*)

Creates an iterable of all `ObjectVersion` resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
object_version_iterator = bucket.object_versions.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(s3.ObjectVersion)`

Returns A list of `ObjectVersion` resources

objects

all ()

Creates an iterable of all `ObjectSummary` resources in the collection.

Request Syntax

```
object_summary_iterator = bucket.objects.all()
```

Return type `list(s3.ObjectSummary)`

Returns A list of `ObjectSummary` resources

delete (***kwargs*)

This operation enables you to delete multiple objects from a bucket using a single HTTP request. You may specify up to 1000 keys.

Request Syntax

```
response = bucket.objects.delete(
    MFA='string',
    RequestPayer='requester'
)
```

Parameters

- **MFA** (*string*) – The concatenation of the authentication device’s serial number, a space, and the value that is displayed on your authentication device.
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
  'Deleted': [
    {
      'Key': 'string',
      'VersionId': 'string',
      'DeleteMarker': True|False,
      'DeleteMarkerVersionId': 'string'
    },
  ],
  'RequestCharged': 'requester',
  'Errors': [
    {
      'Key': 'string',
      'VersionId': 'string',
      'Code': 'string',
      'Message': 'string'
    },
  ],
}
```

Response Structure

- (*dict*) –
 - **Deleted** (*list*) –
 - * (*dict*) –
 - **Key** (*string*) –
 - **VersionId** (*string*) –
 - **DeleteMarker** (*boolean*) –
 - **DeleteMarkerVersionId** (*string*) –
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.
 - **Errors** (*list*) –
 - * (*dict*) –
 - **Key** (*string*) –
 - **VersionId** (*string*) –
 - **Code** (*string*) –

· **Message** (*string*) –

filter (***kwargs*)

Creates an iterable of all ObjectSummary resources in the collection filtered by kwargs passed to method.

Request Syntax

```
object_summary_iterator = bucket.objects.filter(
    Delimiter='string',
    EncodingType='url',
    Marker='string',
    MaxKeys=123,
    Prefix='string'
)
```

Parameters

- **Delimiter** (*string*) – A delimiter is a character you use to group keys.
- **EncodingType** (*string*) – Requests Amazon S3 to encode the object keys in the response and specifies the encoding method to use. An object key may contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.
- **Marker** (*string*) – Specifies the key to start with when listing objects in a bucket.
- **MaxKeys** (*integer*) – Sets the maximum number of keys returned in the response. The response might contain fewer keys but will never contain more.
- **Prefix** (*string*) – Limits the response to keys that begin with the specified prefix.

Return type list(s3.ObjectSummary)

Returns A list of ObjectSummary resources

limit (***kwargs*)

Creates an iterable up to a specified amount of ObjectSummary resources in the collection.

Request Syntax

```
object_summary_iterator = bucket.objects.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type list(s3.ObjectSummary)

Returns A list of ObjectSummary resources

page_size (***kwargs*)

Creates an iterable of all ObjectSummary resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
object_summary_iterator = bucket.objects.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(s3.ObjectSummary)`
Returns A list of `ObjectSummary` resources

Waiters

Waiters provide an interface to wait for a resource to reach a specific state. For more information about waiters refer to the [Resources Introduction Guide](#).

`wait_until_exists()`

Waits until this Bucket exists. This method calls `S3.Waiter.bucket_exists.wait()` which polls `S3.Client.head_bucket()` every 5 seconds until a successful state is reached. An error is returned after 20 failed checks.

Request Syntax

```
bucket.wait_until_exists()
```

Returns None

`wait_until_not_exists()`

Waits until this Bucket is not exists. This method calls `S3.Waiter.bucket_not_exists.wait()` which polls `S3.Client.head_bucket()` every 5 seconds until a successful state is reached. An error is returned after 20 failed checks.

Request Syntax

```
bucket.wait_until_not_exists()
```

Returns None

BucketAcl

class `S3.BucketAcl(bucket_name)`

A resource representing an Amazon Simple Storage Service (S3) `BucketAcl`:

```
import boto3
```

```
s3 = boto3.resource('s3')  
bucket_acl = s3.BucketAcl('bucket_name')
```

Parameters `bucket_name` (*string*) – The `BucketAcl`'s `bucket_name` identifier. This **must** be set.

These are the resource's available identifiers:

- `bucket_name`

These are the resource's available attributes:

- `grants`
- `owner`

These are the resource's available actions:

- `load()`
- `put()`
- `reload()`

These are the resource's available sub-resources:

- `Bucket()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

bucket_name

(*string*) The BucketAcl's bucket_name identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

grants

(*list*) A list of grants.

owner

(*dict*)

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

load()

Calls `s3.Client.get_bucket_acl()` to update the attributes of the BucketAcl resource

Request Syntax

```
bucket_acl.load()
```

Returns None**put (**kwargs)**

Sets the permissions on a bucket using access control lists (ACL).

Request Syntax

```
response = bucket_acl.put(
    ACL='private'|'public-read'|'public-read-write'|'authenticated-read',
    AccessControlPolicy={
        'Grants': [
            {
                'Grantee': {
                    'DisplayName': 'string',
                    'EmailAddress': 'string',
                    'ID': 'string',
                    'Type': 'CanonicalUser'|'AmazonCustomerByEmail'|'Group',
                    'URI': 'string'
                },
                'Permission': 'FULL_CONTROL'|'WRITE'|'WRITE_ACP'|'READ'|'READ_ACP'
            },
        ],
        'Owner': {
            'DisplayName': 'string',
            'ID': 'string'
        }
    },
    GrantFullControl='string',
    GrantRead='string',
    GrantReadACP='string',
    GrantWrite='string',
    GrantWriteACP='string'
)
```

Parameters

- **ACL** (*string*) – The canned ACL to apply to the bucket.
- **AccessControlPolicy** (*dict*) –
 - **Grants** (*list*) – A list of grants.
 - * (*dict*) –
 - **Grantee** (*dict*) –
 - **DisplayName** (*string*) – Screen name of the grantee.
 - **EmailAddress** (*string*) – Email address of the grantee.
 - **ID** (*string*) – The canonical user ID of the grantee.
 - **Type** (*string*) – [REQUIRED] Type of grantee
 - **URI** (*string*) – URI of the grantee group.
 - **Permission** (*string*) – Specifies the permission given to the grantee.
 - **Owner** (*dict*) –
 - * **DisplayName** (*string*) –
 - * **ID** (*string*) –
- **GrantFullControl** (*string*) – Allows grantee the read, write, read ACP, and write ACP permissions on the bucket.
- **GrantRead** (*string*) – Allows grantee to list the objects in the bucket.
- **GrantReadACP** (*string*) – Allows grantee to read the bucket ACL.
- **GrantWrite** (*string*) – Allows grantee to create, overwrite, and delete any object in the bucket.
- **GrantWriteACP** (*string*) – Allows grantee to write the ACL for the applicable bucket.

Returns None

reload()

Calls `s3.Client.get_bucket_acl()` to update the attributes of the `BucketAcl` resource

Request Syntax

```
bucket_acl.reload()
```

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Bucket()

Creates a `Bucket` resource.:

```
bucket = bucket_acl.Bucket()
```

Return type `S3.Bucket`

Returns A `Bucket` resource

BucketCors

class `S3.BucketCors` (*bucket_name*)

A resource representing an Amazon Simple Storage Service (S3) `BucketCors`:

```
import boto3
```

```
s3 = boto3.resource('s3')
bucket_cors = s3.BucketCors('bucket_name')
```


Parameters `bucket_name` (*string*) – The BucketCors’s `bucket_name` identifier. This **must** be set.

These are the resource’s available identifiers:

- `bucket_name`

These are the resource’s available attributes:

- `cors_rules`

These are the resource’s available actions:

- `delete()`
- `load()`
- `put()`
- `reload()`

These are the resource’s available sub-resources:

- `Bucket()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

bucket_name

(*string*) The BucketCors’s `bucket_name` identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

cors_rules

(*list*)

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

delete()

Deletes the cors configuration information set for the bucket.

Request Syntax

```
response = bucket_cors.delete()
```

Returns None

load()

Calls `s3.Client.get_bucket_cors()` to update the attributes of the BucketCors resource

Request Syntax

```
bucket_cors.load()
```

Returns None

put (kwargs)**

Sets the cors configuration for a bucket.

Request Syntax

```
response = bucket_cors.put(
    CORSConfiguration={
        'CORSRules': [
            {
```

```
        'AllowedHeaders': [
            'string',
        ],
        'AllowedMethods': [
            'string',
        ],
        'AllowedOrigins': [
            'string',
        ],
        'ExposeHeaders': [
            'string',
        ],
        'MaxAgeSeconds': 123
    },
]
},
)
```

Parameters **CORSConfiguration** (*dict*) – [REQUIRED]

• **CORSRules** (*list*) – [REQUIRED]

– (*dict*) –

* **AllowedHeaders** (*list*) – Specifies which headers are allowed in a pre-flight OPTIONS request.

· (*string*) –

* **AllowedMethods** (*list*) – [REQUIRED] Identifies HTTP methods that the domain/origin specified in the rule is allowed to execute.

· (*string*) –

* **AllowedOrigins** (*list*) – [REQUIRED] One or more origins you want customers to be able to access the bucket from.

· (*string*) –

* **ExposeHeaders** (*list*) – One or more headers in the response that you want customers to be able to access from their applications (for example, from a JavaScript XMLHttpRequest object).

· (*string*) –

* **MaxAgeSeconds** (*integer*) – The time in seconds that your browser is to cache the preflight response for the specified resource.

Returns None

reload()

Calls `s3.Client.get_bucket_cors()` to update the attributes of the BucketCors resource

Request Syntax

```
bucket_cors.reload()
```

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource’s identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Bucket()

Creates a Bucket resource.:

```
bucket = bucket_cors.Bucket()
```

Return type `S3.Bucket`
Returns A Bucket resource

BucketLifecycle

class `S3.BucketLifecycle` (*bucket_name*)

A resource representing an Amazon Simple Storage Service (S3) BucketLifecycle:

```
import boto3
```

```
s3 = boto3.resource('s3')
bucket_lifecycle = s3.BucketLifecycle('bucket_name')
```

Parameters `bucket_name` (*string*) – The BucketLifecycle’s `bucket_name` identifier. This **must** be set.

These are the resource’s available identifiers:

- `bucket_name`

These are the resource’s available attributes:

- `rules`

These are the resource’s available actions:

- `delete()`
- `load()`
- `put()`
- `reload()`

These are the resource’s available sub-resources:

- `Bucket()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

bucket_name

(*string*) The BucketLifecycle’s `bucket_name` identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

rules

(*list*)

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

delete()

Deletes the lifecycle configuration from the bucket.

Request Syntax

```
response = bucket_lifecycle.delete()
```

Returns `None`

load()

Calls `s3.Client.get_bucket_lifecycle()` to update the attributes of the `BucketLifecycle` resource

Request Syntax

```
bucket_lifecycle.load()
```

Returns None

put (kwargs)**

Deprecated, see the `PutBucketLifecycleConfiguration` operation.

Request Syntax

```
response = bucket_lifecycle.put(
    LifecycleConfiguration={
        'Rules': [
            {
                'Expiration': {
                    'Date': datetime(2015, 1, 1),
                    'Days': 123
                },
                'ID': 'string',
                'Prefix': 'string',
                'Status': 'Enabled'|'Disabled',
                'Transition': {
                    'Date': datetime(2015, 1, 1),
                    'Days': 123,
                    'StorageClass': 'GLACIER'|'STANDARD_IA'
                },
                'NoncurrentVersionTransition': {
                    'NoncurrentDays': 123,
                    'StorageClass': 'GLACIER'|'STANDARD_IA'
                },
                'NoncurrentVersionExpiration': {
                    'NoncurrentDays': 123
                }
            }
        ]
    }
)
```

Parameters `LifecycleConfiguration` (*dict*) –

- **Rules** (*list*) – **[REQUIRED]**

- (*dict*) –

- * **Expiration** (*dict*) –

- **Date** (*datetime*) – Indicates at what date the object is to be moved or deleted. Should be in GMT ISO 8601 Format.

- **Days** (*integer*) – Indicates the lifetime, in days, of the objects that are subject to the rule. The value must be a non-zero positive integer.

- * **ID** (*string*) – Unique identifier for the rule. The value cannot be longer than 255 characters.

- * **Prefix** (*string*) – **[REQUIRED]** Prefix identifying one or more objects to which the rule applies.

- * **Status** (*string*) – **[REQUIRED]** If 'Enabled', the rule is currently being applied. If 'Disabled', the rule is not currently being applied.

- * **Transition** (*dict*) –
 - **Date** (*datetime*) – Indicates at what date the object is to be moved or deleted. Should be in GMT ISO 8601 Format.
 - **Days** (*integer*) – Indicates the lifetime, in days, of the objects that are subject to the rule. The value must be a non-zero positive integer.
 - **StorageClass** (*string*) – The class of storage used to store the object.
- * **NoncurrentVersionTransition** (*dict*) – Container for the transition rule that describes when noncurrent objects transition to the STANDARD_IA or GLACIER storage class. If your bucket is versioning-enabled (or versioning is suspended), you can set this action to request that Amazon S3 transition noncurrent object versions to the STANDARD_IA or GLACIER storage class at a specific period in the object's lifetime.
 - **NoncurrentDays** (*integer*) – Specifies the number of days an object is noncurrent before Amazon S3 can perform the associated action. For information about the noncurrent days calculations, see [How Amazon S3 Calculates When an Object Became Noncurrent in the Amazon Simple Storage Service Developer Guide](#).
 - **StorageClass** (*string*) – The class of storage used to store the object.
- * **NoncurrentVersionExpiration** (*dict*) – Specifies when noncurrent object versions expire. Upon expiration, Amazon S3 permanently deletes the noncurrent object versions. You set this lifecycle configuration action on a bucket that has versioning enabled (or suspended) to request that Amazon S3 delete noncurrent object versions at a specific period in the object's lifetime.
 - **NoncurrentDays** (*integer*) – Specifies the number of days an object is noncurrent before Amazon S3 can perform the associated action. For information about the noncurrent days calculations, see [How Amazon S3 Calculates When an Object Became Noncurrent in the Amazon Simple Storage Service Developer Guide](#).

Returns None

reload()

Calls `s3.Client.get_bucket_lifecycle()` to update the attributes of the `BucketLifecycle` resource

Request Syntax

```
bucket_lifecycle.reload()
```

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Bucket()

Creates a `Bucket` resource.:

```
bucket = bucket_lifecycle.Bucket()
```

Return type `S3.Bucket`
Returns A Bucket resource

BucketLogging

class `S3.BucketLogging` (*bucket_name*)

A resource representing an Amazon Simple Storage Service (S3) BucketLogging:

```
import boto3

s3 = boto3.resource('s3')
bucket_logging = s3.BucketLogging('bucket_name')
```

Parameters `bucket_name` (*string*) – The BucketLogging’s `bucket_name` identifier. This **must** be set.

These are the resource’s available identifiers:

- `bucket_name`

These are the resource’s available attributes:

- `logging_enabled`

These are the resource’s available actions:

- `load()`
- `put()`
- `reload()`

These are the resource’s available sub-resources:

- `Bucket()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

bucket_name

(*string*) The BucketLogging’s `bucket_name` identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

logging_enabled

(*dict*)

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

load()

Calls `s3.Client.get_bucket_logging()` to update the attributes of the BucketLogging resource

Request Syntax

```
bucket_logging.load()
```

Returns None

put (***kwargs*)

Set the logging parameters for a bucket and to specify permissions for who can view and modify the logging parameters. To set the logging status of a bucket, you must be the bucket owner.

Request Syntax

```
response = bucket_logging.put(
    BucketLoggingStatus={
        'LoggingEnabled': {
            'TargetBucket': 'string',
            'TargetGrants': [
                {
                    'Grantee': {
                        'DisplayName': 'string',
                        'EmailAddress': 'string',
                        'ID': 'string',
                        'Type': 'CanonicalUser'|'AmazonCustomerByEmail'|'Group',
                        'URI': 'string'
                    },
                    'Permission': 'FULL_CONTROL'|'READ'|'WRITE'
                },
            ],
            'TargetPrefix': 'string'
        }
    },
)
```

Parameters `BucketLoggingStatus` (*dict*) – [REQUIRED]

• `LoggingEnabled` (*dict*) –

- `TargetBucket` (*string*) – Specifies the bucket where you want Amazon S3 to store server access logs. You can have your logs delivered to any bucket that you own, including the same bucket that is being logged. You can also configure multiple buckets to deliver their logs to the same target bucket. In this case you should choose a different `TargetPrefix` for each source bucket so that the delivered log files can be distinguished by key.
- `TargetGrants` (*list*) –
 - * (*dict*) –
 - `Grantee` (*dict*) –
 - `DisplayName` (*string*) – Screen name of the grantee.
 - `EmailAddress` (*string*) – Email address of the grantee.
 - `ID` (*string*) – The canonical user ID of the grantee.
 - `Type` (*string*) – [REQUIRED] Type of grantee
 - `URI` (*string*) – URI of the grantee group.
 - `Permission` (*string*) – Logging permissions assigned to the Grantee for the bucket.
- `TargetPrefix` (*string*) – This element lets you specify a prefix for the keys that the log files will be stored under.

Returns None

`reload()`

Calls `s3.Client.get_bucket_logging()` to update the attributes of the `BucketLogging` resource

Request Syntax

```
bucket_logging.reload()
```

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Bucket ()

Creates a Bucket resource.:

```
bucket = bucket_logging.Bucket()
```

Return type `S3.Bucket`

Returns A Bucket resource

BucketNotification

class S3.BucketNotification (*bucket_name*)

A resource representing an Amazon Simple Storage Service (S3) BucketNotification:

```
import boto3
```

```
s3 = boto3.resource('s3')
```

```
bucket_notification = s3.BucketNotification('bucket_name')
```

Parameters **bucket_name** (*string*) – The BucketNotification's bucket_name identifier. This **must** be set.

These are the resource's available identifiers:

- `bucket_name`

These are the resource's available attributes:

- `lambda_function_configurations`
- `queue_configurations`
- `topic_configurations`

These are the resource's available actions:

- `load()`
- `put()`
- `reload()`

These are the resource's available sub-resources:

- `Bucket()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

bucket_name

(*string*) The BucketNotification's bucket_name identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

lambda_function_configurations

(*list*)

queue_configurations

(*list*)

topic_configurations

(*list*)

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

load()

Calls `s3.Client.get_bucket_notification_configuration()` to update the attributes of the `BucketNotification` resource

Request Syntax

```
bucket_notification.load()
```

Returns None

put (**kwargs)

Enables notifications of specified events for a bucket.

Request Syntax

```
response = bucket_notification.put(
    NotificationConfiguration={
        'TopicConfigurations': [
            {
                'Id': 'string',
                'TopicArn': 'string',
                'Events': [
                    's3:ReducedRedundancyLostObject'|'s3:ObjectCreated:*'|'s3:ObjectCreated:
                ],
                'Filter': {
                    'Key': {
                        'FilterRules': [
                            {
                                'Name': 'prefix'|'suffix',
                                'Value': 'string'
                            },
                        ],
                    },
                },
            },
        ],
        'QueueConfigurations': [
            {
                'Id': 'string',
                'QueueArn': 'string',
                'Events': [
                    's3:ReducedRedundancyLostObject'|'s3:ObjectCreated:*'|'s3:ObjectCreated:
                ],
                'Filter': {
                    'Key': {
                        'FilterRules': [
                            {
                                'Name': 'prefix'|'suffix',
                                'Value': 'string'
                            },
                        ],
                    },
                },
            },
        ],
    },
```

```
],
'LambdaFunctionConfigurations': [
    {
        'Id': 'string',
        'LambdaFunctionArn': 'string',
        'Events': [
            's3:ReducedRedundancyLostObject'|'s3:ObjectCreated:*'|'s3:ObjectCreated:
        ],
        'Filter': {
            'Key': {
                'FilterRules': [
                    {
                        'Name': 'prefix'|'suffix',
                        'Value': 'string'
                    },
                ],
            },
        },
    },
],
},
),
```

Parameters NotificationConfiguration (*dict*) – **[REQUIRED]** Container for specifying the notification configuration of the bucket. If this element is empty, notifications are turned off on the bucket.

- **TopicConfigurations** (*list*) –

- (*dict*) – Container for specifying the configuration when you want Amazon S3 to publish events to an Amazon Simple Notification Service (Amazon SNS) topic.

- * **Id** (*string*) – Optional unique identifier for configurations in a notification configuration. If you don't provide one, Amazon S3 will assign an ID.

- * **TopicArn** (*string*) – **[REQUIRED]** Amazon SNS topic ARN to which Amazon S3 will publish a message when it detects events of specified type.

- * **Events** (*list*) – **[REQUIRED]**

- (*string*) – Bucket event for which to send notifications.

- * **Filter** (*dict*) – Container for object key name filtering rules. For information about key name filtering, go to [Configuring Event Notifications](#) in the Amazon Simple Storage Service Developer Guide.

- **Key** (*dict*) – Container for object key name prefix and suffix filtering rules.

- **FilterRules** (*list*) – A list of containers for key value pair that defines the criteria for the filter rule.

- (*dict*) – Container for key value pair that defines the criteria for the filter rule.

- **Name** (*string*) – Object key name prefix or suffix identifying one or more objects to which the filtering rule applies. Maximum prefix length can be up to 1,024 characters. Overlapping prefixes and suffixes are not supported. For more information, go to [Configuring Event Notifications](#) in the Amazon Simple Storage Service Developer Guide.

- **Value** (*string*) –

- **QueueConfigurations** (*list*) –

- (*dict*) – Container for specifying an configuration when you want Amazon S3 to publish events to an Amazon Simple Queue Service (Amazon SQS) queue.
 - * **Id** (*string*) – Optional unique identifier for configurations in a notification configuration. If you don’t provide one, Amazon S3 will assign an ID.
 - * **QueueArn** (*string*) – **[REQUIRED]** Amazon SQS queue ARN to which Amazon S3 will publish a message when it detects events of specified type.
 - * **Events** (*list*) – **[REQUIRED]**
 - (*string*) – Bucket event for which to send notifications.
 - * **Filter** (*dict*) – Container for object key name filtering rules. For information about key name filtering, go to [Configuring Event Notifications](#) in the Amazon Simple Storage Service Developer Guide.
 - **Key** (*dict*) – Container for object key name prefix and suffix filtering rules.
 - **FilterRules** (*list*) – A list of containers for key value pair that defines the criteria for the filter rule.
 - (*dict*) – Container for key value pair that defines the criteria for the filter rule.
 - **Name** (*string*) – Object key name prefix or suffix identifying one or more objects to which the filtering rule applies. Maximum prefix length can be up to 1,024 characters. Overlapping prefixes and suffixes are not supported. For more information, go to [Configuring Event Notifications](#) in the Amazon Simple Storage Service Developer Guide.
 - **Value** (*string*) –
- **LambdaFunctionConfigurations** (*list*) –
 - (*dict*) – Container for specifying the AWS Lambda notification configuration.
 - * **Id** (*string*) – Optional unique identifier for configurations in a notification configuration. If you don’t provide one, Amazon S3 will assign an ID.
 - * **LambdaFunctionArn** (*string*) – **[REQUIRED]** Lambda cloud function ARN that Amazon S3 can invoke when it detects events of the specified type.
 - * **Events** (*list*) – **[REQUIRED]**
 - (*string*) – Bucket event for which to send notifications.
 - * **Filter** (*dict*) – Container for object key name filtering rules. For information about key name filtering, go to [Configuring Event Notifications](#) in the Amazon Simple Storage Service Developer Guide.
 - **Key** (*dict*) – Container for object key name prefix and suffix filtering rules.
 - **FilterRules** (*list*) – A list of containers for key value pair that defines the criteria for the filter rule.
 - (*dict*) – Container for key value pair that defines the criteria for the filter rule.
 - **Name** (*string*) – Object key name prefix or suffix identifying one or more objects to which the filtering rule applies. Maximum prefix length can be up to 1,024 characters. Overlapping prefixes and suffixes are not supported. For more information, go to [Configuring Event Notifications](#) in the Amazon Simple Storage Service Developer Guide.
 - **Value** (*string*) –

Returns None

reload()

Calls `s3.Client.get_bucket_notification_configuration()` to update the attributes of the `BucketNotification` resource

Request Syntax

```
bucket_notification.reload()
```

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Bucket()

Creates a `Bucket` resource.:

```
bucket = bucket_notification.Bucket()
```

Return type `S3.Bucket`

Returns A `Bucket` resource

BucketPolicy

class `S3.BucketPolicy` (*bucket_name*)

A resource representing an Amazon Simple Storage Service (S3) `BucketPolicy`:

```
import boto3
```

```
s3 = boto3.resource('s3')
bucket_policy = s3.BucketPolicy('bucket_name')
```

Parameters `bucket_name` (*string*) – The `BucketPolicy`'s `bucket_name` identifier. This **must** be set.

These are the resource's available identifiers:

- `bucket_name`

These are the resource's available attributes:

- `policy`

These are the resource's available actions:

- `delete()`
- `load()`
- `put()`
- `reload()`

These are the resource's available sub-resources:

- `Bucket()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

bucket_name

(*string*) The `BucketPolicy`'s `bucket_name` identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

policy

(*string*) The bucket policy as a JSON document.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

delete()

Deletes the policy from the bucket.

Request Syntax

```
response = bucket_policy.delete()
```

Returns None

load()

Calls `s3.Client.get_bucket_policy()` to update the attributes of the BucketPolicy resource

Request Syntax

```
bucket_policy.load()
```

Returns None

put(kwargs)**

Replaces a policy on a bucket. If the bucket already has a policy, the one in this request completely replaces it.

Request Syntax

```
response = bucket_policy.put(
    Policy='string'
)
```

Parameters **Policy** (*string*) – **[REQUIRED]** The bucket policy as a JSON document.

Returns None

reload()

Calls `s3.Client.get_bucket_policy()` to update the attributes of the BucketPolicy resource

Request Syntax

```
bucket_policy.reload()
```

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Bucket()

Creates a Bucket resource.:

```
bucket = bucket_policy.Bucket()
```

Return type `S3.Bucket`

Returns A Bucket resource

BucketRequestPayment

class `S3.BucketRequestPayment` (*bucket_name*)

A resource representing an Amazon Simple Storage Service (S3) BucketRequestPayment:

```
import boto3
```

```
s3 = boto3.resource('s3')
```

```
bucket_request_payment = s3.BucketRequestPayment('bucket_name')
```

Parameters `bucket_name` (*string*) – The BucketRequestPayment’s `bucket_name` identifier. This **must** be set.

These are the resource’s available identifiers:

- `bucket_name`

These are the resource’s available attributes:

- `payer`

These are the resource’s available actions:

- `load()`
- `put()`
- `reload()`

These are the resource’s available sub-resources:

- `Bucket()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

bucket_name

(*string*) The BucketRequestPayment’s `bucket_name` identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

payer

(*string*) Specifies who pays for the download and request fees.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

load()

Calls `s3.Client.get_bucket_request_payment()` to update the attributes of the BucketRequestPayment resource

Request Syntax

```
bucket_request_payment.load()
```

Returns None

put (***kwargs*)

Sets the request payment configuration for a bucket. By default, the bucket owner pays for downloads from the bucket. This configuration parameter enables the bucket owner (only) to specify that the person requesting the download will be charged for the download. Documentation on requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/RequesterPaysBuckets.html>

Request Syntax

```
response = bucket_request_payment.put (
    RequestPaymentConfiguration={
        'Payer': 'Requester'|'BucketOwner'
    }
)
```

Parameters **RequestPaymentConfiguration** (*dict*) – [REQUIRED]

- **Payer** (*string*) – [REQUIRED] Specifies who pays for the download and request fees.

Returns None

reload ()

Calls `s3.Client.get_bucket_request_payment ()` to update the attributes of the `BucketRequestPayment` resource

Request Syntax

```
bucket_request_payment.reload()
```

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Bucket ()

Creates a Bucket resource.:

```
bucket = bucket_request_payment.Bucket ()
```

Return type `S3.Bucket`

Returns A Bucket resource

BucketTagging

class `S3.BucketTagging` (*bucket_name*)

A resource representing an Amazon Simple Storage Service (S3) BucketTagging:

```
import boto3
```

```
s3 = boto3.resource('s3')
bucket_tagging = s3.BucketTagging('bucket_name')
```

Parameters **bucket_name** (*string*) – The BucketTagging's bucket_name identifier. This **must** be set.

These are the resource's available identifiers:

- `bucket_name`

These are the resource's available attributes:

- `tag_set`

These are the resource's available actions:

- `delete()`
- `load()`
- `put()`
- `reload()`

These are the resource's available sub-resources:

- `Bucket()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

bucket_name

(*string*) The BucketTagging's `bucket_name` identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

tag_set

(*list*)

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

delete()

Deletes the tags from the bucket.

Request Syntax

```
response = bucket_tagging.delete()
```

Returns None

load()

Calls `s3.Client.get_bucket_tagging()` to update the attributes of the BucketTagging resource

Request Syntax

```
bucket_tagging.load()
```

Returns None

put (kwargs)**

Sets the tags for a bucket.

Request Syntax

```
response = bucket_tagging.put(
    Tagging={
        'TagSet': [
            {
                'Key': 'string',
                'Value': 'string'
            },
        ]
    }
```



```
}
)
```

Parameters **Tagging** (*dict*) – [REQUIRED]

- **TagSet** (*list*) – [REQUIRED]

- (*dict*) –

- * **Key** (*string*) – [REQUIRED] Name of the tag.

- * **Value** (*string*) – [REQUIRED] Value of the tag.

Returns None

reload()

Calls `s3.Client.get_bucket_tagging()` to update the attributes of the BucketTagging resource

Request Syntax

```
bucket_tagging.reload()
```

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource’s identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Bucket ()

Creates a Bucket resource.:

```
bucket = bucket_tagging.Bucket()
```

Return type `S3.Bucket`

Returns A Bucket resource

BucketVersioning

class `S3.BucketVersioning` (*bucket_name*)

A resource representing an Amazon Simple Storage Service (S3) BucketVersioning:

```
import boto3
```

```
s3 = boto3.resource('s3')
```

```
bucket_versioning = s3.BucketVersioning('bucket_name')
```

Parameters **bucket_name** (*string*) – The BucketVersioning’s bucket_name identifier. This **must** be set.

These are the resource’s available identifiers:

- `bucket_name`

These are the resource’s available attributes:

- `mfa_delete`
- `status`

These are the resource’s available actions:

- `enable()`
- `load()`
- `put()`
- `reload()`
- `suspend()`

These are the resource's available sub-resources:

- `Bucket()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

bucket_name

(*string*) The BucketVersioning's bucket_name identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

mfa_delete

(*string*) Specifies whether MFA delete is enabled in the bucket versioning configuration. This element is only returned if the bucket has been configured with MFA delete. If the bucket has never been so configured, this element is not returned.

status

(*string*) The versioning state of the bucket.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

enable (***kwargs*)

Sets the versioning state of an existing bucket. To set the versioning state, you must be the bucket owner.

Request Syntax

```
response = bucket_versioning.enable(  
    MFA='string',  
)
```

Parameters **MFA** (*string*) – The concatenation of the authentication device's serial number, a space, and the value that is displayed on your authentication device.

Returns `None`

load()

Calls `s3.Client.get_bucket_versioning()` to update the attributes of the BucketVersioning resource

Request Syntax

```
bucket_versioning.load()
```

Returns `None`

put (***kwargs*)

Sets the versioning state of an existing bucket. To set the versioning state, you must be the bucket owner.

Request Syntax

```
response = bucket_versioning.put(  
    MFA='string',  
    VersioningConfiguration={  
        'MFADelete': 'Enabled'|'Disabled',
```

```
        'Status': 'Enabled'|'Suspended'
    }
)
```

Parameters

- **MFA** (*string*) – The concatenation of the authentication device’s serial number, a space, and the value that is displayed on your authentication device.
- **VersioningConfiguration** (*dict*) – **[REQUIRED]**
 - **MFADelete** (*string*) – Specifies whether MFA delete is enabled in the bucket versioning configuration. This element is only returned if the bucket has been configured with MFA delete. If the bucket has never been so configured, this element is not returned.
 - **Status** (*string*) – The versioning state of the bucket.

Returns None**reload()**

Calls `s3.Client.get_bucket_versioning()` to update the attributes of the `BucketVersioning` resource

Request Syntax

```
bucket_versioning.reload()
```

Returns None**suspend (**kwargs)**

Sets the versioning state of an existing bucket. To set the versioning state, you must be the bucket owner.

Request Syntax

```
response = bucket_versioning.suspend(
    MFA='string',
)
```

Parameters **MFA** (*string*) – The concatenation of the authentication device’s serial number, a space, and the value that is displayed on your authentication device.

Returns None**Sub-resources**

Sub-resources are methods that create a new instance of a child resource. This resource’s identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Bucket ()

Creates a `Bucket` resource.:

```
bucket = bucket_versioning.Bucket()
```

Return type `S3.Bucket`**Returns** A `Bucket` resource**BucketWebsite**

```
class S3.BucketWebsite (bucket_name)
```

A resource representing an Amazon Simple Storage Service (S3) `BucketWebsite`:

```
import boto3

s3 = boto3.resource('s3')
bucket_website = s3.BucketWebsite('bucket_name')
```

Parameters **bucket_name** (*string*) – The BucketWebsite’s bucket_name identifier. This **must** be set.

These are the resource’s available identifiers:

- `bucket_name`

These are the resource’s available attributes:

- `error_document`
- `index_document`
- `redirect_all_requests_to`
- `routing_rules`

These are the resource’s available actions:

- `delete()`
- `load()`
- `put()`
- `reload()`

These are the resource’s available sub-resources:

- `Bucket()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

bucket_name

(*string*) The BucketWebsite’s bucket_name identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

error_document

(*dict*)

index_document

(*dict*)

redirect_all_requests_to

(*dict*)

routing_rules

(*list*)

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

delete()

This operation removes the website configuration from the bucket.

Request Syntax

```
response = bucket_website.delete()
```

Returns None

load()

Calls `s3.Client.get_bucket_website()` to update the attributes of the `BucketWebsite` resource

Request Syntax

```
bucket_website.load()
```

Returns None

put (kwargs)**

Set the website configuration for a bucket.

Request Syntax

```
response = bucket_website.put(
    WebsiteConfiguration={
        'ErrorDocument': {
            'Key': 'string'
        },
        'IndexDocument': {
            'Suffix': 'string'
        },
        'RedirectAllRequestsTo': {
            'HostName': 'string',
            'Protocol': 'http'|'https'
        },
        'RoutingRules': [
            {
                'Condition': {
                    'HttpErrorCodeReturnedEquals': 'string',
                    'KeyPrefixEquals': 'string'
                },
                'Redirect': {
                    'HostName': 'string',
                    'HttpRedirectCode': 'string',
                    'Protocol': 'http'|'https',
                    'ReplaceKeyPrefixWith': 'string',
                    'ReplaceKeyWith': 'string'
                }
            }
        ]
    }
)
```

Parameters WebsiteConfiguration (dict) – [REQUIRED]

- **ErrorDocument (dict) –**
 - **Key (string) – [REQUIRED]** The object key name to use when a 4XX class error occurs.
- **IndexDocument (dict) –**
 - **Suffix (string) – [REQUIRED]** A suffix that is appended to a request that is for a directory on the website endpoint (e.g. if the suffix is `index.html` and you make a request to `samplebucket/images/` the data that is returned will be for the object with the key name `images/index.html`) The suffix must not be empty and must not include a slash character.
- **RedirectAllRequestsTo (dict) –**
 - **HostName (string) – [REQUIRED]** Name of the host where requests will be redirected.

- **Protocol** (*string*) – Protocol to use (http, https) when redirecting requests. The default is the protocol that is used in the original request.
- **RoutingRules** (*list*) –
 - (*dict*) –
 - * **Condition** (*dict*) – A container for describing a condition that must be met for the specified redirect to apply. For example, 1. If request is for pages in the /docs folder, redirect to the /documents folder. 2. If request results in HTTP error 4xx, redirect request to another host where you might process the error.
 - **HttpErrorCodeReturnedEquals** (*string*) – The HTTP error code when the redirect is applied. In the event of an error, if the error code equals this value, then the specified redirect is applied. Required when parent element Condition is specified and sibling KeyPrefixEquals is not specified. If both are specified, then both must be true for the redirect to be applied.
 - **KeyPrefixEquals** (*string*) – The object key name prefix when the redirect is applied. For example, to redirect requests for ExamplePage.html, the key prefix will be ExamplePage.html. To redirect request for all pages with the prefix docs/, the key prefix will be /docs, which identifies all objects in the docs/ folder. Required when the parent element Condition is specified and sibling HttpErrorCodeReturnedEquals is not specified. If both conditions are specified, both must be true for the redirect to be applied.
 - * **Redirect** (*dict*) – **[REQUIRED]** Container for redirect information. You can redirect requests to another host, to another page, or with another protocol. In the event of an error, you can specify a different error code to return.
 - **HostName** (*string*) – The host name to use in the redirect request.
 - **HttpRedirectCode** (*string*) – The HTTP redirect code to use on the response. Not required if one of the siblings is present.
 - **Protocol** (*string*) – Protocol to use (http, https) when redirecting requests. The default is the protocol that is used in the original request.
 - **ReplaceKeyPrefixWith** (*string*) – The object key prefix to use in the redirect request. For example, to redirect requests for all pages with prefix docs/ (objects in the docs/ folder) to documents/, you can set a condition block with KeyPrefixEquals set to docs/ and in the Redirect set ReplaceKeyPrefixWith to /documents. Not required if one of the siblings is present. Can be present only if ReplaceKeyWith is not provided.
 - **ReplaceKeyWith** (*string*) – The specific object key to use in the redirect request. For example, redirect request to error.html. Not required if one of the sibling is present. Can be present only if ReplaceKeyPrefixWith is not provided.

Returns None

reload()

Calls `s3.Client.get_bucket_website()` to update the attributes of the BucketWebsite resource

Request Syntax

```
bucket_website.reload()
```

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Bucket ()

Creates a Bucket resource.:

```
bucket = bucket_website.Bucket()
```

Return type `S3.Bucket`

Returns A Bucket resource

MultipartUpload

class `S3.MultipartUpload` (*bucket_name*, *object_key*, *id*)

A resource representing an Amazon Simple Storage Service (S3) MultipartUpload:

```
import boto3
```

```
s3 = boto3.resource('s3')
```

```
multipart_upload = s3.MultipartUpload('bucket_name', 'object_key', 'id')
```

Parameters

- **bucket_name** (*string*) – The MultipartUpload's bucket_name identifier. This **must** be set.
- **object_key** (*string*) – The MultipartUpload's object_key identifier. This **must** be set.
- **id** (*string*) – The MultipartUpload's id identifier. This **must** be set.

These are the resource's available identifiers:

- `bucket_name`
- `object_key`
- `id`

These are the resource's available attributes:

- `initiated`
- `initiator`
- `key`
- `owner`
- `storage_class`
- `upload_id`

These are the resource's available actions:

- `abort()`
- `complete()`

These are the resource's available sub-resources:

- `Object()`
- `Part()`

These are the resource's available collections:

- `parts`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

bucket_name

(*string*) The MultipartUpload's bucket_name identifier. This **must** be set.

object_key

(*string*) The MultipartUpload's object_key identifier. This **must** be set.

id

(*string*) The MultipartUpload's id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

initiated

(*datetime*) Date and time at which the multipart upload was initiated.

initiator

(*dict*) Identifies who initiated the multipart upload.

key

(*string*) Key of the object for which the multipart upload was initiated.

owner

(*dict*)

storage_class

(*string*) The class of storage used to store the object.

upload_id

(*string*) Upload ID that identifies the multipart upload.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

abort (***kwargs*)

Aborts a multipart upload.

To verify that all parts have been removed, so you don't get charged for the part storage, you should call the List Parts operation and ensure the parts list is empty.

Request Syntax

```
response = multipart_upload.abort(  
    RequestPayer='requester'  
)
```

Parameters **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns**Response Syntax**


```
{
    'RequestCharged': 'requester'
}
```

Response Structure

- (*dict*) –
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

complete (***kwargs*)

Completes a multipart upload by assembling previously uploaded parts.

Request Syntax

```
object = multipart_upload.complete(
    MultipartUpload={
        'Parts': [
            {
                'ETag': 'string',
                'PartNumber': 123
            },
        ]
    },
    RequestPayer='requester'
)
```

Parameters

- **MultipartUpload** (*dict*) –
 - **Parts** (*list*) –
 - * (*dict*) –
 - **ETag** (*string*) – Entity tag returned when the part was uploaded.
 - **PartNumber** (*integer*) – Part number that identifies the part. This is a positive integer between 1 and 10,000.
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type `s3.Object`

Returns Object resource

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Object ()

Creates a Object resource.:

```
object = multipart_upload.Object()
```

Return type `S3.Object`

Returns A Object resource

Part (*part_number*)

Creates a MultipartUploadPart resource.:

```
multipart_upload_part = multipart_upload.Part('part_number')
```

Parameters `part_number` (*string*) – The Part’s `part_number` identifier. This **must** be set.

Return type `S3.MultipartUploadPart`

Returns A `MultipartUploadPart` resource

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

`parts`

`all()`

Creates an iterable of all `MultipartUploadPart` resources in the collection.

Request Syntax

```
multipart_upload_part_iterator = multipart_upload.parts.all()
```

Return type `list(s3.MultipartUploadPart)`

Returns A list of `MultipartUploadPart` resources

`filter(**kwargs)`

Creates an iterable of all `MultipartUploadPart` resources in the collection filtered by `kwargs` passed to method.

Request Syntax

```
multipart_upload_part_iterator = multipart_upload.parts.filter(
    MaxParts=123,
    PartNumberMarker=123,
    RequestPayer='requester'
)
```

Parameters

- **MaxParts** (*integer*) – Sets the maximum number of parts to return.
- **PartNumberMarker** (*integer*) – Specifies the part after which listing should begin. Only parts with higher part numbers will be listed.
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type `list(s3.MultipartUploadPart)`

Returns A list of `MultipartUploadPart` resources

`limit(**kwargs)`

Creates an iterable up to a specified amount of `MultipartUploadPart` resources in the collection.

Request Syntax

```
multipart_upload_part_iterator = multipart_upload.parts.limit(
    count=123
)
```

Parameters `count` (*integer*) – The limit to the number of resources in the iterable.

Return type `list(s3.MultipartUploadPart)`

Returns A list of MultipartUploadPart resources

page_size (***kwargs*)

Creates an iterable of all MultipartUploadPart resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
multipart_upload_part_iterator = multipart_upload.parts.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type list(s3.MultipartUploadPart)

Returns A list of MultipartUploadPart resources

MultipartUploadPart

class S3.MultipartUploadPart (*bucket_name, object_key, multipart_upload_id, part_number*)

A resource representing an Amazon Simple Storage Service (S3) MultipartUploadPart:

```
import boto3
```

```
s3 = boto3.resource('s3')
```

```
multipart_upload_part = s3.MultipartUploadPart('bucket_name', 'object_key', 'multipart_upload_id',
```

Parameters

- **bucket_name** (*string*) – The MultipartUploadPart’s bucket_name identifier. This **must** be set.
- **object_key** (*string*) – The MultipartUploadPart’s object_key identifier. This **must** be set.
- **multipart_upload_id** (*string*) – The MultipartUploadPart’s multipart_upload_id identifier. This **must** be set.
- **part_number** (*string*) – The MultipartUploadPart’s part_number identifier. This **must** be set.

These are the resource’s available identifiers:

- bucket_name
- object_key
- multipart_upload_id
- part_number

These are the resource’s available attributes:

- e_tag
- last_modified
- size

These are the resource’s available actions:

- copy_from()
- upload()

These are the resource’s available sub-resources:

- MultipartUpload()

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

bucket_name

(*string*) The MultipartUploadPart's bucket_name identifier. This **must** be set.

object_key

(*string*) The MultipartUploadPart's object_key identifier. This **must** be set.

multipart_upload_id

(*string*) The MultipartUploadPart's multipart_upload_id identifier. This **must** be set.

part_number

(*string*) The MultipartUploadPart's part_number identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

e_tag

(*string*) Entity tag returned when the part was uploaded.

last_modified

(*datetime*) Date and time at which the part was uploaded.

size

(*integer*) Size of the uploaded part data.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

copy_from (***kwargs*)

Uploads a part by copying data from an existing object as data source.

Request Syntax

```
response = multipart_upload_part.copy_from(
    CopySource='string',
    CopySourceIfMatch='string',
    CopySourceIfModifiedSince=datetime(2015, 1, 1),
    CopySourceIfNoneMatch='string',
    CopySourceIfUnmodifiedSince=datetime(2015, 1, 1),
    CopySourceRange='string',
    SSECustomerAlgorithm='string',
    SSECustomerKey='string',
    CopySourceSSECustomerAlgorithm='string',
    CopySourceSSECustomerKey='string',
    CopySourceSSECustomerKeyMD5='string',
    RequestPayer='requester'
)
```

Parameters

- **CopySource** (*string*) – **[REQUIRED]** The name of the source bucket and key name of the source object, separated by a slash (/). Must be URL-encoded.
- **CopySourceIfMatch** (*string*) – Copies the object if its entity tag (ETag) matches the specified tag.
- **CopySourceIfModifiedSince** (*datetime*) – Copies the object if it has been modified since the specified time.
- **CopySourceIfNoneMatch** (*string*) – Copies the object if its entity tag (ETag) is different than the specified ETag.

- **CopySourceIfUnmodifiedSince** (*datetime*) – Copies the object if it hasn't been modified since the specified time.
- **CopySourceRange** (*string*) – The range of bytes to copy from the source object. The range value must use the form bytes=first-last, where the first and last are the zero-based byte offsets to copy. For example, bytes=0-9 indicates that you want to copy the first ten bytes of the source. You can copy a range only if the source object is greater than 5 GB.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header. This must be the same encryption key specified in the initiate multipart upload request.
- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **CopySourceSSECustomerAlgorithm** (*string*) – Specifies the algorithm to use when decrypting the source object (e.g., AES256).
- **CopySourceSSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use to decrypt the source object. The encryption key provided in this header must be one that was used when the source object was created.
- **CopySourceSSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error.
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
  'CopySourceVersionId': 'string',
  'CopyPartResult': {
    'ETag': 'string',
    'LastModified': datetime(2015, 1, 1)
  },
  'ServerSideEncryption': 'AES256'|'aws:kms',
  'SSECustomerAlgorithm': 'string',
  'SSECustomerKeyMD5': 'string',
  'SSEKMSKeyId': 'string',
  'RequestCharged': 'requester'
}
```

Response Structure

- (*dict*) –

- **CopySourceVersionId** (*string*) – The version of the source object that was copied, if you have enabled versioning on the source bucket.
- **CopyPartResult** (*dict*) –
 - * **ETag** (*string*) – Entity tag of the object.
 - * **LastModified** (*datetime*) – Date and time at which the object was uploaded.
- **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
- **SSECustomerAlgorithm** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.
- **SSECustomerKeyMD5** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.
- **SSEKMSKeyId** (*string*) – If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.
- **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

upload (***kwargs*)

Uploads a part in a multipart upload.

Note: After you initiate multipart upload and upload one or more parts, you must either complete or abort multipart upload in order to stop getting charged for storage of the uploaded parts. Only after you either complete or abort multipart upload, Amazon S3 frees up the parts storage and stops charging you for the parts storage.

Request Syntax

```
response = multipart_upload_part.upload(  
    Body=b'bytes',  
    ContentLength=123,  
    ContentMD5='string',  
    SSECustomerAlgorithm='string',  
    SSECustomerKey='string',  
    RequestPayer='requester'  
)
```

Parameters

- **Body** (*bytes*) –
- **ContentLength** (*integer*) – Size of the body in bytes. This parameter is useful when the size of the body cannot be determined automatically.
- **ContentMD5** (*string*) –
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header. This must be the same encryption key specified in the initiate multipart upload request.
- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error.

Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required

- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
  'ServerSideEncryption': 'AES256'|'aws:kms',
  'ETag': 'string',
  'SSECustomerAlgorithm': 'string',
  'SSECustomerKeyMD5': 'string',
  'SSEKMSKeyId': 'string',
  'RequestCharged': 'requester'
}
```

Response Structure

- (*dict*) –
 - **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
 - **ETag** (*string*) – Entity tag for the uploaded object.
 - **SSECustomerAlgorithm** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.
 - **SSECustomerKeyMD5** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.
 - **SSEKMSKeyId** (*string*) – If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

MultipartUpload()

Creates a MultipartUpload resource.:

```
multipart_upload = multipart_upload_part.MultipartUpload()
```

Return type `S3.MultipartUpload`

Returns A MultipartUpload resource

Object

class `S3.Object` (*bucket_name*, *key*)

A resource representing an Amazon Simple Storage Service (S3) Object:

```
import boto3

s3 = boto3.resource('s3')
object = s3.Object('bucket_name', 'key')
```

Parameters

- **bucket_name** (*string*) – The Object’s bucket_name identifier. This **must** be set.
- **key** (*string*) – The Object’s key identifier. This **must** be set.

These are the resource’s available identifiers:

- bucket_name
- key

These are the resource’s available attributes:

- accept_ranges
- cache_control
- content_disposition
- content_encoding
- content_language
- content_length
- content_type
- delete_marker
- e_tag
- expiration
- expires
- last_modified
- metadata
- missing_meta
- replication_status
- request_charged
- restore
- server_side_encryption
- sse_customer_algorithm
- sse_customer_key_md5
- ssekms_key_id
- storage_class
- version_id
- website_redirect_location

These are the resource’s available actions:

- copy_from()
- delete()
- download_file()
- get()
- initiate_multipart_upload()
- load()
- put()
- reload()
- upload_file()

These are the resource’s available sub-resources:

- Acl()
- Bucket()
- MultipartUpload()
- Version()

These are the resource’s available waiters:

- wait_until_exists()

- `wait_until_not_exists()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

bucket_name

(*string*) The Object's bucket_name identifier. This **must** be set.

key

(*string*) The Object's key identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

accept_ranges

(*string*)

cache_control

(*string*) Specifies caching behavior along the request/reply chain.

content_disposition

(*string*) Specifies presentational information for the object.

content_encoding

(*string*) Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.

content_language

(*string*) The language the content is in.

content_length

(*integer*) Size of the body in bytes.

content_type

(*string*) A standard MIME type describing the format of the object data.

delete_marker

(*boolean*) Specifies whether the object retrieved was (true) or was not (false) a Delete Marker. If false, this response header does not appear in the response.

e_tag

(*string*) An ETag is an opaque identifier assigned by a web server to a specific version of a resource found at a URL

expiration

(*string*) If the object expiration is configured (see PUT Bucket lifecycle), the response includes this header. It includes the expiry-date and rule-id key value pairs providing object expiration information. The value of the rule-id is URL encoded.

expires

(*datetime*) The date and time at which the object is no longer cacheable.

last_modified

(*datetime*) Last modified date of the object

metadata

(*dict*) A map of metadata to store with the object in S3.

missing_meta

(*integer*) This is set to the number of metadata entries not returned in x-amz-meta headers. This can

happen if you create metadata using an API like SOAP that supports more flexible metadata than the REST API. For example, using SOAP, you can create metadata whose values are not legal HTTP headers.

replication_status

(string)

request_charged

(string) If present, indicates that the requester was successfully charged for the request.

restore

(string) Provides information about object restoration operation and expiration time of the restored object copy.

server_side_encryption

(string) The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).

sse_customer_algorithm

(string) If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.

sse_customer_key_md5

(string) If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.

ssekms_key_id

(string) If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.

storage_class

(string)

version_id

(string) Version of the object.

website_redirect_location

(string) If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

download_file (Filename, ExtraArgs=None, Callback=None, Config=None)

upload_file (Filename, ExtraArgs=None, Callback=None, Config=None)

copy_from (**kwargs)

Creates a copy of an object that is already stored in Amazon S3.

Request Syntax

```
response = object.copy_from(  
    ACL='private'|'public-read'|'public-read-write'|'authenticated-read'|'bucket-owner-read',  
    CacheControl='string',  
    ContentDisposition='string',  
    ContentEncoding='string',  
    ContentLanguage='string',  
    ContentType='string',
```

```

CopySource='string',
CopySourceIfMatch='string',
CopySourceIfModifiedSince=datetime(2015, 1, 1),
CopySourceIfNoneMatch='string',
CopySourceIfUnmodifiedSince=datetime(2015, 1, 1),
Expires=datetime(2015, 1, 1),
GrantFullControl='string',
GrantRead='string',
GrantReadACP='string',
GrantWriteACP='string',
Metadata={
    'string': 'string'
},
MetadataDirective='COPY'|'REPLACE',
ServerSideEncryption='AES256'|'aws:kms',
StorageClass='STANDARD'|'REDUCED_REDUNDANCY'|'STANDARD_IA',
WebsiteRedirectLocation='string',
SSECustomerAlgorithm='string',
SSECustomerKey='string',
SSEKMSKeyId='string',
CopySourceSSECustomerAlgorithm='string',
CopySourceSSECustomerKey='string',
CopySourceSSECustomerKeyMD5='string',
RequestPayer='requester'
)

```

Parameters

- **ACL** (*string*) – The canned ACL to apply to the object.
- **CacheControl** (*string*) – Specifies caching behavior along the request/reply chain.
- **ContentDisposition** (*string*) – Specifies presentational information for the object.
- **ContentEncoding** (*string*) – Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.
- **ContentLanguage** (*string*) – The language the content is in.
- **ContentType** (*string*) – A standard MIME type describing the format of the object data.
- **CopySource** (*string*) – **[REQUIRED]** The name of the source bucket and key name of the source object, separated by a slash (/). Must be URL-encoded.
- **CopySourceIfMatch** (*string*) – Copies the object if its entity tag (ETag) matches the specified tag.
- **CopySourceIfModifiedSince** (*datetime*) – Copies the object if it has been modified since the specified time.
- **CopySourceIfNoneMatch** (*string*) – Copies the object if its entity tag (ETag) is different than the specified ETag.
- **CopySourceIfUnmodifiedSince** (*datetime*) – Copies the object if it hasn't been modified since the specified time.
- **Expires** (*datetime*) – The date and time at which the object is no longer cacheable.
- **GrantFullControl** (*string*) – Gives the grantee READ, READ_ACP, and WRITE_ACP permissions on the object.
- **GrantRead** (*string*) – Allows grantee to read the object data and its metadata.
- **GrantReadACP** (*string*) – Allows grantee to read the object ACL.
- **GrantWriteACP** (*string*) – Allows grantee to write the ACL for the applicable object.

- **Metadata** (*dict*) – A map of metadata to store with the object in S3.
 - (*string*) –
 - * (*string*) –
- **MetadataDirective** (*string*) – Specifies whether the metadata is copied from the source object or replaced with metadata provided in the request.
- **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
- **StorageClass** (*string*) – The type of storage to use for the object. Defaults to 'STANDARD'.
- **WebsiteRedirectLocation** (*string*) – If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.
- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **SSEKMSKeyId** (*string*) – Specifies the AWS KMS key ID to use for object encryption. All GET and PUT requests for an object protected by AWS KMS will fail if not made via SSL or using SigV4. Documentation on configuring any of the officially supported AWS SDKs and CLI can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingAWSSDK.html#specify-signature-version>
- **CopySourceSSECustomerAlgorithm** (*string*) – Specifies the algorithm to use when decrypting the source object (e.g., AES256).
- **CopySourceSSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use to decrypt the source object. The encryption key provided in this header must be one that was used when the source object was created.
- **CopySourceSSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error.
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
  'CopyObjectResult': {
    'ETag': 'string',
    'LastModified': datetime(2015, 1, 1)
```

```

    },
    'Expiration': 'string',
    'CopySourceVersionId': 'string',
    'ServerSideEncryption': 'AES256'|'aws:kms',
    'SSECustomerAlgorithm': 'string',
    'SSECustomerKeyMD5': 'string',
    'SSEKMSKeyId': 'string',
    'RequestCharged': 'requester'
}

```

Response Structure

- (*dict*) –
 - **CopyObjectResult** (*dict*) –
 - * **ETag** (*string*) –
 - * **LastModified** (*datetime*) –
 - **Expiration** (*string*) – If the object expiration is configured, the response includes this header.
 - **CopySourceVersionId** (*string*) –
 - **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
 - **SSECustomerAlgorithm** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.
 - **SSECustomerKeyMD5** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.
 - **SSEKMSKeyId** (*string*) – If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

`delete` (***kwargs*)

Removes the null version (if there is one) of an object and inserts a delete marker, which becomes the latest version of the object. If there isn't a null version, Amazon S3 does not remove any objects.

Request Syntax

```

response = object.delete(
    MFA='string',
    VersionId='string',
    RequestPayer='requester'
)

```

Parameters

- **MFA** (*string*) – The concatenation of the authentication device's serial number, a space, and the value that is displayed on your authentication device.
- **VersionId** (*string*) – VersionId used to reference a specific version of the object.
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
    'DeleteMarker': True|False,
    'VersionId': 'string',
    'RequestCharged': 'requester'
}
```

Response Structure

- (*dict*) –
 - **DeleteMarker** (*boolean*) – Specifies whether the versioned object that was permanently deleted was (true) or was not (false) a delete marker.
 - **VersionId** (*string*) – Returns the version ID of the delete marker created as a result of the DELETE operation.
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

Upload a file to an S3 object.

Usage:

```
import boto3
s3 = boto3.resource('s3')
s3.Object('mybucket', 'hello.txt').upload_file('/tmp/hello.txt')
```

Similar behavior as `S3Transfer`'s `upload_file()` method, except that parameters are capitalized. Detailed examples can be found at *S3Transfer's Usage*.

get (***kwargs*)

Retrieves objects from Amazon S3.

Request Syntax

```
response = object.get(
    IfMatch='string',
    IfModifiedSince=datetime(2015, 1, 1),
    IfNoneMatch='string',
    IfUnmodifiedSince=datetime(2015, 1, 1),
    Range='string',
    ResponseCacheControl='string',
    ResponseContentDisposition='string',
    ResponseContentEncoding='string',
    ResponseContentLanguage='string',
    ResponseContentType='string',
    ResponseExpires=datetime(2015, 1, 1),
    VersionId='string',
    SSECustomerAlgorithm='string',
    SSECustomerKey='string',
    RequestPayer='requester'
)
```

Parameters

- **IfMatch** (*string*) – Return the object only if its entity tag (ETag) is the same as the one specified, otherwise return a 412 (precondition failed).
- **IfModifiedSince** (*datetime*) – Return the object only if it has been modified since the specified time, otherwise return a 304 (not modified).
- **IfNoneMatch** (*string*) – Return the object only if its entity tag (ETag) is different from the one specified, otherwise return a 304 (not modified).

- **IfUnmodifiedSince** (*datetime*) – Return the object only if it has not been modified since the specified time, otherwise return a 412 (precondition failed).
- **Range** (*string*) – Downloads the specified range bytes of an object. For more information about the HTTP Range header, go to <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35>.
- **ResponseCacheControl** (*string*) – Sets the Cache-Control header of the response.
- **ResponseContentDisposition** (*string*) – Sets the Content-Disposition header of the response
- **ResponseContentEncoding** (*string*) – Sets the Content-Encoding header of the response.
- **ResponseContentLanguage** (*string*) – Sets the Content-Language header of the response.
- **ResponseContentType** (*string*) – Sets the Content-Type header of the response.
- **ResponseExpires** (*datetime*) – Sets the Expires header of the response.
- **VersionId** (*string*) – VersionId used to reference a specific version of the object.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.
- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
    'Body': b'bytes',
    'DeleteMarker': True|False,
    'AcceptRanges': 'string',
    'Expiration': 'string',
    'Restore': 'string',
    'LastModified': datetime(2015, 1, 1),
    'ContentLength': 123,
    'ETag': 'string',
    'MissingMeta': 123,
    'VersionId': 'string',
    'CacheControl': 'string',
    'ContentDisposition': 'string',
    'ContentEncoding': 'string',
```

```
'ContentLanguage': 'string',
'ContentRange': 'string',
'ContentType': 'string',
'Expires': datetime(2015, 1, 1),
'WebsiteRedirectLocation': 'string',
'ServerSideEncryption': 'AES256'|'aws:kms',
'Metadata': {
    'string': 'string'
},
'SSECustomerAlgorithm': 'string',
'SSECustomerKeyMD5': 'string',
'SSEKMSKeyId': 'string',
'StorageClass': 'STANDARD'|'REDUCED_REDUNDANCY'|'STANDARD_IA',
'RequestCharged': 'requester',
'ReplicationStatus': 'COMPLETE'|'PENDING'|'FAILED'|'REPLICA'
}
```

Response Structure

- (*dict*) –
 - **Body** (*bytes*) – Object data.
 - **DeleteMarker** (*boolean*) – Specifies whether the object retrieved was (true) or was not (false) a Delete Marker. If false, this response header does not appear in the response.
 - **AcceptRanges** (*string*) –
 - **Expiration** (*string*) – If the object expiration is configured (see PUT Bucket lifecycle), the response includes this header. It includes the expiry-date and rule-id key value pairs providing object expiration information. The value of the rule-id is URL encoded.
 - **Restore** (*string*) – Provides information about object restoration operation and expiration time of the restored object copy.
 - **LastModified** (*datetime*) – Last modified date of the object
 - **ContentLength** (*integer*) – Size of the body in bytes.
 - **ETag** (*string*) – An ETag is an opaque identifier assigned by a web server to a specific version of a resource found at a URL
 - **MissingMeta** (*integer*) – This is set to the number of metadata entries not returned in x-amz-meta headers. This can happen if you create metadata using an API like SOAP that supports more flexible metadata than the REST API. For example, using SOAP, you can create metadata whose values are not legal HTTP headers.
 - **VersionId** (*string*) – Version of the object.
 - **CacheControl** (*string*) – Specifies caching behavior along the request/reply chain.
 - **ContentDisposition** (*string*) – Specifies presentational information for the object.
 - **ContentEncoding** (*string*) – Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.
 - **ContentLanguage** (*string*) – The language the content is in.
 - **ContentRange** (*string*) – The portion of the object returned in the response.
 - **ContentType** (*string*) – A standard MIME type describing the format of the object data.
 - **Expires** (*datetime*) – The date and time at which the object is no longer cacheable.

- **WebsiteRedirectLocation** (*string*) – If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.
- **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
- **Metadata** (*dict*) – A map of metadata to store with the object in S3.
 - * (*string*) –
 - (*string*) –
- **SSECustomerAlgorithm** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.
- **SSECustomerKeyMD5** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.
- **SSEKMSKeyId** (*string*) – If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.
- **StorageClass** (*string*) –
- **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.
- **ReplicationStatus** (*string*) –

initiate_multipart_upload (***kwargs*)

Initiates a multipart upload and returns an upload ID.

Note: After you initiate multipart upload and upload one or more parts, you must either complete or abort multipart upload in order to stop getting charged for storage of the uploaded parts. Only after you either complete or abort multipart upload, Amazon S3 frees up the parts storage and stops charging you for the parts storage.

Request Syntax

```
multipart_upload = object.initiate_multipart_upload(
    ACL='private'|'public-read'|'public-read-write'|'authenticated-read'|'bucket-owner-
    CacheControl='string',
    ContentDisposition='string',
    ContentEncoding='string',
    ContentLanguage='string',
    ContentType='string',
    Expires=datetime(2015, 1, 1),
    GrantFullControl='string',
    GrantRead='string',
    GrantReadACP='string',
    GrantWriteACP='string',
    Metadata={
        'string': 'string'
    },
    ServerSideEncryption='AES256'|'aws:kms',
    StorageClass='STANDARD'|'REDUCED_REDUNDANCY'|'STANDARD_IA',
    WebsiteRedirectLocation='string',
    SSECustomerAlgorithm='string',
    SSECustomerKey='string',
    SSEKMSKeyId='string',
    RequestPayer='requester'
)
```

Parameters

- **ACL** (*string*) – The canned ACL to apply to the object.
- **CacheControl** (*string*) – Specifies caching behavior along the request/reply chain.
- **ContentDisposition** (*string*) – Specifies presentational information for the object.
- **ContentEncoding** (*string*) – Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.
- **ContentLanguage** (*string*) – The language the content is in.
- **ContentType** (*string*) – A standard MIME type describing the format of the object data.
- **Expires** (*datetime*) – The date and time at which the object is no longer cacheable.
- **GrantFullControl** (*string*) – Gives the grantee READ, READ_ACP, and WRITE_ACP permissions on the object.
- **GrantRead** (*string*) – Allows grantee to read the object data and its meta-data.
- **GrantReadACP** (*string*) – Allows grantee to read the object ACL.
- **GrantWriteACP** (*string*) – Allows grantee to write the ACL for the applicable object.
- **Metadata** (*dict*) – A map of metadata to store with the object in S3.
 - (*string*) –
 - * (*string*) –
- **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
- **StorageClass** (*string*) – The type of storage to use for the object. Defaults to 'STANDARD'.
- **WebsiteRedirectLocation** (*string*) – If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.
- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **SSEKMSKeyId** (*string*) – Specifies the AWS KMS key ID to use for object encryption. All GET and PUT requests for an object protected by AWS KMS will fail if not made via SSL or using SigV4. Documentation on configuring any of the officially supported AWS SDKs and CLI can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingAWSSDK.html#specify-signature-version>
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on

downloading objects from requester pays buckets can be found at
<http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type `s3.MultipartUpload`

Returns MultipartUpload resource

load()

Calls `s3.Client.head_object()` to update the attributes of the Object resource

Request Syntax

```
object.load()
```

Returns None

put (kwargs)**

Adds an object to a bucket.

Request Syntax

```
response = object.put(
    ACL='private'|'public-read'|'public-read-write'|'authenticated-read'|'bucket-owner-
    Body=b'bytes',
    CacheControl='string',
    ContentDisposition='string',
    ContentEncoding='string',
    ContentLanguage='string',
    ContentLength=123,
    ContentMD5='string',
    ContentType='string',
    Expires=datetime(2015, 1, 1),
    GrantFullControl='string',
    GrantRead='string',
    GrantReadACP='string',
    GrantWriteACP='string',
    Metadata={
        'string': 'string'
    },
    ServerSideEncryption='AES256'|'aws:kms',
    StorageClass='STANDARD'|'REDUCED_REDUNDANCY'|'STANDARD_IA',
    WebsiteRedirectLocation='string',
    SSECustomerAlgorithm='string',
    SSECustomerKey='string',
    SSEKMSKeyId='string',
    RequestPayer='requester'
)
```

Parameters

- **ACL** (*string*) – The canned ACL to apply to the object.
- **Body** (*bytes*) – Object data.
- **CacheControl** (*string*) – Specifies caching behavior along the request/reply chain.
- **ContentDisposition** (*string*) – Specifies presentational information for the object.
- **ContentEncoding** (*string*) – Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.
- **ContentLanguage** (*string*) – The language the content is in.

- **ContentLength** (*integer*) – Size of the body in bytes. This parameter is useful when the size of the body cannot be determined automatically.
- **ContentMD5** (*string*) –
- **ContentType** (*string*) – A standard MIME type describing the format of the object data.
- **Expires** (*datetime*) – The date and time at which the object is no longer cacheable.
- **GrantFullControl** (*string*) – Gives the grantee READ, READ_ACP, and WRITE_ACP permissions on the object.
- **GrantRead** (*string*) – Allows grantee to read the object data and its meta-data.
- **GrantReadACP** (*string*) – Allows grantee to read the object ACL.
- **GrantWriteACP** (*string*) – Allows grantee to write the ACL for the applicable object.
- **Metadata** (*dict*) – A map of metadata to store with the object in S3.
 - (*string*) –
 - * (*string*) –
- **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
- **StorageClass** (*string*) – The type of storage to use for the object. Defaults to 'STANDARD'.
- **WebsiteRedirectLocation** (*string*) – If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.
- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **SSEKMSKeyId** (*string*) – Specifies the AWS KMS key ID to use for object encryption. All GET and PUT requests for an object protected by AWS KMS will fail if not made via SSL or using SigV4. Documentation on configuring any of the officially supported AWS SDKs and CLI can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingAWS SDK.html#specify-signature-version>
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
    'Expiration': 'string',
    'ETag': 'string',
    'ServerSideEncryption': 'AES256'|'aws:kms',
    'VersionId': 'string',
    'SSECustomerAlgorithm': 'string',
    'SSECustomerKeyMD5': 'string',
    'SSEKMSKeyId': 'string',
    'RequestCharged': 'requester'
}
```

Response Structure

- (*dict*) –
 - **Expiration** (*string*) – If the object expiration is configured, this will contain the expiration date (expiry-date) and rule ID (rule-id). The value of rule-id is URL encoded.
 - **ETag** (*string*) – Entity tag for the uploaded object.
 - **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
 - **VersionId** (*string*) – Version of the object.
 - **SSECustomerAlgorithm** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.
 - **SSECustomerKeyMD5** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.
 - **SSEKMSKeyId** (*string*) – If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

reload()

Calls `s3.Client.head_object()` to update the attributes of the Object resource

Request Syntax

```
object.reload()
```

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Acl()

Creates a ObjectAcl resource.:

```
object_acl = object.Acl()
```

Return type `S3.ObjectAcl`

Returns A ObjectAcl resource

Bucket()

Creates a Bucket resource.:

```
bucket = object.Bucket()
```

Return type `S3.Bucket`
Returns A Bucket resource

MultipartUpload (*id*)

Creates a MultipartUpload resource.:

```
multipart_upload = object.MultipartUpload('id')
```

Parameters *id* (*string*) – The MultipartUpload’s id identifier. This **must** be set.
Return type `S3.MultipartUpload`
Returns A MultipartUpload resource

Version (*id*)

Creates a ObjectVersion resource.:

```
object_version = object.Version('id')
```

Parameters *id* (*string*) – The Version’s id identifier. This **must** be set.
Return type `S3.ObjectVersion`
Returns A ObjectVersion resource

Waiters

Waiters provide an interface to wait for a resource to reach a specific state. For more information about waiters refer to the [Resources Introduction Guide](#).

wait_until_exists (***kwargs*)

Waits until this Object is exists. This method calls `S3.Waiter.object_exists.wait()` which polls. `S3.Client.head_object()` every 5 seconds until a successful state is reached. An error is returned after 20 failed checks.

Request Syntax

```
object.wait_until_exists(  
    IfMatch='string',  
    IfModifiedSince=datetime(2015, 1, 1),  
    IfNoneMatch='string',  
    IfUnmodifiedSince=datetime(2015, 1, 1),  
    Range='string',  
    VersionId='string',  
    SSECustomerAlgorithm='string',  
    SSECustomerKey='string',  
    RequestPayer='requester'  
)
```

Parameters

- **IfMatch** (*string*) – Return the object only if its entity tag (ETag) is the same as the one specified, otherwise return a 412 (precondition failed).
- **IfModifiedSince** (*datetime*) – Return the object only if it has been modified since the specified time, otherwise return a 304 (not modified).
- **IfNoneMatch** (*string*) – Return the object only if its entity tag (ETag) is different from the one specified, otherwise return a 304 (not modified).
- **IfUnmodifiedSince** (*datetime*) – Return the object only if it has not been modified since the specified time, otherwise return a 412 (precondition failed).

- **Range** (*string*) – Downloads the specified range bytes of an object. For more information about the HTTP Range header, go to <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35>.
- **VersionId** (*string*) – VersionId used to reference a specific version of the object.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.
- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Returns None

wait_until_not_exists (***kwargs*)

Waits until this Object is not exists. This method calls `S3.Waiter.object_not_exists.wait()` which polls `S3.Client.head_object()` every 5 seconds until a successful state is reached. An error is returned after 20 failed checks.

Request Syntax

```
object.wait_until_not_exists(
    IfMatch='string',
    IfModifiedSince=datetime(2015, 1, 1),
    IfNoneMatch='string',
    IfUnmodifiedSince=datetime(2015, 1, 1),
    Range='string',
    VersionId='string',
    SSECustomerAlgorithm='string',
    SSECustomerKey='string',
    RequestPayer='requester'
)
```

Parameters

- **IfMatch** (*string*) – Return the object only if its entity tag (ETag) is the same as the one specified, otherwise return a 412 (precondition failed).
- **IfModifiedSince** (*datetime*) – Return the object only if it has been modified since the specified time, otherwise return a 304 (not modified).
- **IfNoneMatch** (*string*) – Return the object only if its entity tag (ETag) is different from the one specified, otherwise return a 304 (not modified).
- **IfUnmodifiedSince** (*datetime*) – Return the object only if it has not been modified since the specified time, otherwise return a 412 (precondition failed).
- **Range** (*string*) – Downloads the specified range bytes of an object. For more information about the HTTP Range header, go to <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35>.
- **VersionId** (*string*) – VersionId used to reference a specific version of the object.

- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.
- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Returns None

ObjectAcl

class S3.**ObjectAcl** (*bucket_name*, *object_key*)

A resource representing an Amazon Simple Storage Service (S3) ObjectAcl:

```
import boto3

s3 = boto3.resource('s3')
object_acl = s3.ObjectAcl('bucket_name', 'object_key')
```

Parameters

- **bucket_name** (*string*) – The ObjectAcl’s bucket_name identifier. This **must** be set.
- **object_key** (*string*) – The ObjectAcl’s object_key identifier. This **must** be set.

These are the resource’s available identifiers:

- `bucket_name`
- `object_key`

These are the resource’s available attributes:

- `grants`
- `owner`
- `request_charged`

These are the resource’s available actions:

- `load()`
- `put()`
- `reload()`

These are the resource’s available sub-resources:

- `Object()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

bucket_name

(*string*) The ObjectAcl’s bucket_name identifier. This **must** be set.

object_key

(*string*) The ObjectAcl's object_key identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

grants

(*list*) A list of grants.

owner

(*dict*)

request_charged

(*string*) If present, indicates that the requester was successfully charged for the request.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

load()

Calls `s3.Client.get_object_acl()` to update the attributes of the ObjectAcl resource

Request Syntax

```
object_acl.load()
```

Returns None

put (kwargs)**

uses the acl subresource to set the access control list (ACL) permissions for an object that already exists in a bucket

Request Syntax

```
response = object_acl.put(
    ACL='private'|'public-read'|'public-read-write'|'authenticated-read'|'bucket-owner-read',
    AccessControlPolicy={
        'Grants': [
            {
                'Grantee': {
                    'DisplayName': 'string',
                    'EmailAddress': 'string',
                    'ID': 'string',
                    'Type': 'CanonicalUser'|'AmazonCustomerByEmail'|'Group',
                    'URI': 'string'
                },
                'Permission': 'FULL_CONTROL'|'WRITE'|'WRITE_ACP'|'READ'|'READ_ACP'
            },
        ],
        'Owner': {
            'DisplayName': 'string',
            'ID': 'string'
        }
    },
    GrantFullControl='string',
    GrantRead='string',
    GrantReadACP='string',
    GrantWrite='string',
```

```
GrantWriteACP='string',
RequestPayer='requester'
)
```

Parameters

- **ACL** (*string*) – The canned ACL to apply to the object.
- **AccessControlPolicy** (*dict*) –
 - **Grants** (*list*) – A list of grants.
 - * (*dict*) –
 - **Grantee** (*dict*) –
 - **DisplayName** (*string*) – Screen name of the grantee.
 - **EmailAddress** (*string*) – Email address of the grantee.
 - **ID** (*string*) – The canonical user ID of the grantee.
 - **Type** (*string*) – [REQUIRED] Type of grantee
 - **URI** (*string*) – URI of the grantee group.
 - **Permission** (*string*) – Specifies the permission given to the grantee.
 - **Owner** (*dict*) –
 - * **DisplayName** (*string*) –
 - * **ID** (*string*) –
- **GrantFullControl** (*string*) – Allows grantee the read, write, read ACP, and write ACP permissions on the bucket.
- **GrantRead** (*string*) – Allows grantee to list the objects in the bucket.
- **GrantReadACP** (*string*) – Allows grantee to read the bucket ACL.
- **GrantWrite** (*string*) – Allows grantee to create, overwrite, and delete any object in the bucket.
- **GrantWriteACP** (*string*) – Allows grantee to write the ACL for the applicable bucket.
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
    'RequestCharged': 'requester'
}
```

Response Structure

- (*dict*) –
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

reload()

Calls `s3.Client.get_object_acl()` to update the attributes of the `ObjectAcl` resource

Request Syntax

```
object_acl.reload()
```

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

`Object()`

Creates a `Object` resource.:

```
object = object_acl.Object()
```

Return type `S3.Object`

Returns A `Object` resource

ObjectSummary

class `S3.ObjectSummary(bucket_name, key)`

A resource representing an Amazon Simple Storage Service (S3) `ObjectSummary`:

```
import boto3
```

```
s3 = boto3.resource('s3')
```

```
object_summary = s3.ObjectSummary('bucket_name', 'key')
```

Parameters

- **bucket_name** (*string*) – The `ObjectSummary`'s `bucket_name` identifier. This **must** be set.
- **key** (*string*) – The `ObjectSummary`'s `key` identifier. This **must** be set.

These are the resource's available identifiers:

- `bucket_name`
- `key`

These are the resource's available attributes:

- `e_tag`
- `last_modified`
- `owner`
- `size`
- `storage_class`

These are the resource's available actions:

- `copy_from()`
- `delete()`
- `get()`
- `initiate_multipart_upload()`
- `put()`

These are the resource's available sub-resources:

- `Acl()`
- `Bucket()`
- `MultipartUpload()`
- `Object()`
- `Version()`

These are the resource's available waiters:

- `wait_until_exists()`
- `wait_until_not_exists()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

bucket_name

(string) The ObjectSummary's bucket_name identifier. This **must** be set.

key

(string) The ObjectSummary's key identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

e_tag

(string)

last_modified

(datetime)

owner

(dict)

size

(integer)

storage_class

(string) The class of storage used to store the object.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

copy_from(kwargs)**

Creates a copy of an object that is already stored in Amazon S3.

Request Syntax

```
response = object_summary.copy_from(
    ACL='private'|'public-read'|'public-read-write'|'authenticated-read'|'bucket-owner-read',
    CacheControl='string',
    ContentDisposition='string',
    ContentEncoding='string',
    ContentLanguage='string',
    ContentType='string',
    CopySource='string',
    CopySourceIfMatch='string',
    CopySourceIfModifiedSince=datetime(2015, 1, 1),
    CopySourceIfNoneMatch='string',
    CopySourceIfUnmodifiedSince=datetime(2015, 1, 1),
    Expires=datetime(2015, 1, 1),
    GrantFullControl='string',
    GrantRead='string',
    GrantReadACP='string',
    GrantWriteACP='string',
    Metadata={
        'string': 'string'
    },
    MetadataDirective='COPY'|'REPLACE',
    ServerSideEncryption='AES256'|'aws:kms',
    StorageClass='STANDARD'|'REDUCED_REDUNDANCY'|'STANDARD_IA',
    WebsiteRedirectLocation='string',
    SSECustomerAlgorithm='string',
    SSECustomerKey='string',
```

```

SSEKMSKeyId='string',
CopySourceSSECustomerAlgorithm='string',
CopySourceSSECustomerKey='string',
CopySourceSSECustomerKeyMD5='string',
RequestPayer='requester'
)

```

Parameters

- **ACL** (*string*) – The canned ACL to apply to the object.
- **CacheControl** (*string*) – Specifies caching behavior along the request/reply chain.
- **ContentDisposition** (*string*) – Specifies presentational information for the object.
- **ContentEncoding** (*string*) – Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.
- **ContentLanguage** (*string*) – The language the content is in.
- **ContentType** (*string*) – A standard MIME type describing the format of the object data.
- **CopySource** (*string*) – **[REQUIRED]** The name of the source bucket and key name of the source object, separated by a slash (/). Must be URL-encoded.
- **CopySourceIfMatch** (*string*) – Copies the object if its entity tag (ETag) matches the specified tag.
- **CopySourceIfModifiedSince** (*datetime*) – Copies the object if it has been modified since the specified time.
- **CopySourceIfNoneMatch** (*string*) – Copies the object if its entity tag (ETag) is different than the specified ETag.
- **CopySourceIfUnmodifiedSince** (*datetime*) – Copies the object if it hasn't been modified since the specified time.
- **Expires** (*datetime*) – The date and time at which the object is no longer cacheable.
- **GrantFullControl** (*string*) – Gives the grantee READ, READ_ACP, and WRITE_ACP permissions on the object.
- **GrantRead** (*string*) – Allows grantee to read the object data and its metadata.
- **GrantReadACP** (*string*) – Allows grantee to read the object ACL.
- **GrantWriteACP** (*string*) – Allows grantee to write the ACL for the applicable object.
- **Metadata** (*dict*) – A map of metadata to store with the object in S3.
 - (*string*) –
 - * (*string*) –
- **MetadataDirective** (*string*) – Specifies whether the metadata is copied from the source object or replaced with metadata provided in the request.
- **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
- **StorageClass** (*string*) – The type of storage to use for the object. Defaults to 'STANDARD'.
- **WebsiteRedirectLocation** (*string*) – If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key

must be appropriate for use with the algorithm specified in the `x-amz-server-side-encryption-customer-algorithm` header.

- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **SSEKMSKeyId** (*string*) – Specifies the AWS KMS key ID to use for object encryption. All GET and PUT requests for an object protected by AWS KMS will fail if not made via SSL or using SigV4. Documentation on configuring any of the officially supported AWS SDKs and CLI can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingAWSSDK.html#specify-signature-version>
- **CopySourceSSECustomerAlgorithm** (*string*) – Specifies the algorithm to use when decrypting the source object (e.g., AES256).
- **CopySourceSSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use to decrypt the source object. The encryption key provided in this header must be one that was used when the source object was created.
- **CopySourceSSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error.
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
  'CopyObjectResult': {
    'ETag': 'string',
    'LastModified': datetime(2015, 1, 1)
  },
  'Expiration': 'string',
  'CopySourceVersionId': 'string',
  'ServerSideEncryption': 'AES256'|'aws:kms',
  'SSECustomerAlgorithm': 'string',
  'SSECustomerKeyMD5': 'string',
  'SSEKMSKeyId': 'string',
  'RequestCharged': 'requester'
}
```

Response Structure

- (*dict*) –
 - **CopyObjectResult** (*dict*) –
 - * **ETag** (*string*) –
 - * **LastModified** (*datetime*) –
 - **Expiration** (*string*) – If the object expiration is configured, the response includes this header.
 - **CopySourceVersionId** (*string*) –

- **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
- **SSECustomerAlgorithm** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.
- **SSECustomerKeyMD5** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.
- **SSEKMSKeyId** (*string*) – If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.
- **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

delete (***kwargs*)

Removes the null version (if there is one) of an object and inserts a delete marker, which becomes the latest version of the object. If there isn't a null version, Amazon S3 does not remove any objects.

Request Syntax

```
response = object_summary.delete(
    MFA='string',
    VersionId='string',
    RequestPayer='requester'
)
```

Parameters

- **MFA** (*string*) – The concatenation of the authentication device's serial number, a space, and the value that is displayed on your authentication device.
- **VersionId** (*string*) – VersionId used to reference a specific version of the object.
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
    'DeleteMarker': True|False,
    'VersionId': 'string',
    'RequestCharged': 'requester'
}
```

Response Structure

- (*dict*) –
 - **DeleteMarker** (*boolean*) – Specifies whether the versioned object that was permanently deleted was (true) or was not (false) a delete marker.
 - **VersionId** (*string*) – Returns the version ID of the delete marker created as a result of the DELETE operation.
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

`get (**kwargs)`

Retrieves objects from Amazon S3.

Request Syntax

```
response = object_summary.get(
    IfMatch='string',
    IfModifiedSince=datetime(2015, 1, 1),
    IfNoneMatch='string',
    IfUnmodifiedSince=datetime(2015, 1, 1),
    Range='string',
    ResponseCacheControl='string',
    ResponseContentDisposition='string',
    ResponseContentEncoding='string',
    ResponseContentLanguage='string',
    ResponseContentType='string',
    ResponseExpires=datetime(2015, 1, 1),
    VersionId='string',
    SSECustomerAlgorithm='string',
    SSECustomerKey='string',
    RequestPayer='requester'
)
```

Parameters

- **IfMatch** (*string*) – Return the object only if its entity tag (ETag) is the same as the one specified, otherwise return a 412 (precondition failed).
- **IfModifiedSince** (*datetime*) – Return the object only if it has been modified since the specified time, otherwise return a 304 (not modified).
- **IfNoneMatch** (*string*) – Return the object only if its entity tag (ETag) is different from the one specified, otherwise return a 304 (not modified).
- **IfUnmodifiedSince** (*datetime*) – Return the object only if it has not been modified since the specified time, otherwise return a 412 (precondition failed).
- **Range** (*string*) – Downloads the specified range bytes of an object. For more information about the HTTP Range header, go to <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35>.
- **ResponseCacheControl** (*string*) – Sets the Cache-Control header of the response.
- **ResponseContentDisposition** (*string*) – Sets the Content-Disposition header of the response.
- **ResponseContentEncoding** (*string*) – Sets the Content-Encoding header of the response.
- **ResponseContentLanguage** (*string*) – Sets the Content-Language header of the response.
- **ResponseContentType** (*string*) – Sets the Content-Type header of the response.
- **ResponseExpires** (*datetime*) – Sets the Expires header of the response.
- **VersionId** (*string*) – VersionId used to reference a specific version of the object.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.
- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error.

Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required

- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
    'Body': b'bytes',
    'DeleteMarker': True|False,
    'AcceptRanges': 'string',
    'Expiration': 'string',
    'Restore': 'string',
    'LastModified': datetime(2015, 1, 1),
    'ContentLength': 123,
    'ETag': 'string',
    'MissingMeta': 123,
    'VersionId': 'string',
    'CacheControl': 'string',
    'ContentDisposition': 'string',
    'ContentEncoding': 'string',
    'ContentLanguage': 'string',
    'ContentRange': 'string',
    'ContentType': 'string',
    'Expires': datetime(2015, 1, 1),
    'WebsiteRedirectLocation': 'string',
    'ServerSideEncryption': 'AES256'|'aws:kms',
    'Metadata': {
        'string': 'string'
    },
    'SSECustomerAlgorithm': 'string',
    'SSECustomerKeyMD5': 'string',
    'SSEKMSKeyId': 'string',
    'StorageClass': 'STANDARD'|'REDUCED_REDUNDANCY'|'STANDARD_IA',
    'RequestCharged': 'requester',
    'ReplicationStatus': 'COMPLETE'|'PENDING'|'FAILED'|'REPLICA'
}
```

Response Structure

- (*dict*) –
 - **Body** (*bytes*) – Object data.
 - **DeleteMarker** (*boolean*) – Specifies whether the object retrieved was (true) or was not (false) a Delete Marker. If false, this response header does not appear in the response.
 - **AcceptRanges** (*string*) –
 - **Expiration** (*string*) – If the object expiration is configured (see PUT Bucket lifecycle), the response includes this header. It includes the expiry-date and rule-id key value pairs providing object expiration information. The value of the rule-id is URL encoded.
 - **Restore** (*string*) – Provides information about object restoration operation and expiration time of the restored object copy.
 - **LastModified** (*datetime*) – Last modified date of the object

- **ContentLength** (*integer*) – Size of the body in bytes.
- **ETag** (*string*) – An ETag is an opaque identifier assigned by a web server to a specific version of a resource found at a URL
- **MissingMeta** (*integer*) – This is set to the number of metadata entries not returned in x-amz-meta headers. This can happen if you create metadata using an API like SOAP that supports more flexible metadata than the REST API. For example, using SOAP, you can create metadata whose values are not legal HTTP headers.
- **VersionId** (*string*) – Version of the object.
- **CacheControl** (*string*) – Specifies caching behavior along the request/reply chain.
- **ContentDisposition** (*string*) – Specifies presentational information for the object.
- **ContentEncoding** (*string*) – Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.
- **ContentLanguage** (*string*) – The language the content is in.
- **ContentRange** (*string*) – The portion of the object returned in the response.
- **ContentType** (*string*) – A standard MIME type describing the format of the object data.
- **Expires** (*datetime*) – The date and time at which the object is no longer cacheable.
- **WebsiteRedirectLocation** (*string*) – If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.
- **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
- **Metadata** (*dict*) – A map of metadata to store with the object in S3.
 - * (*string*) –
 - (*string*) –
- **SSECustomerAlgorithm** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.
- **SSECustomerKeyMD5** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.
- **SSEKMSKeyId** (*string*) – If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.
- **StorageClass** (*string*) –
- **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.
- **ReplicationStatus** (*string*) –

initiate_multipart_upload (**kwargs)

Initiates a multipart upload and returns an upload ID.

Note: After you initiate multipart upload and upload one or more parts, you must either complete or abort multipart upload in order to stop getting charged for storage of the uploaded parts. Only after you either complete or abort multipart upload, Amazon S3 frees up the parts storage and stops charging you for the parts storage.

Request Syntax

```
multipart_upload = object_summary.initiate_multipart_upload(
    ACL='private'|'public-read'|'public-read-write'|'authenticated-read'|'bucket-owner-read',
    CacheControl='string',
    ContentDisposition='string',
    ContentEncoding='string',
    ContentLanguage='string',
    ContentType='string',
    Expires=datetime(2015, 1, 1),
    GrantFullControl='string',
    GrantRead='string',
    GrantReadACP='string',
    GrantWriteACP='string',
    Metadata={
        'string': 'string'
    },
    ServerSideEncryption='AES256'|'aws:kms',
    StorageClass='STANDARD'|'REDUCED_REDUNDANCY'|'STANDARD_IA',
    WebsiteRedirectLocation='string',
    SSECustomerAlgorithm='string',
    SSECustomerKey='string',
    SSEKMSKeyId='string',
    RequestPayer='requester'
)
```

Parameters

- **ACL** (*string*) – The canned ACL to apply to the object.
- **CacheControl** (*string*) – Specifies caching behavior along the request/reply chain.
- **ContentDisposition** (*string*) – Specifies presentational information for the object.
- **ContentEncoding** (*string*) – Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.
- **ContentLanguage** (*string*) – The language the content is in.
- **ContentType** (*string*) – A standard MIME type describing the format of the object data.
- **Expires** (*datetime*) – The date and time at which the object is no longer cacheable.
- **GrantFullControl** (*string*) – Gives the grantee READ, READ_ACP, and WRITE_ACP permissions on the object.
- **GrantRead** (*string*) – Allows grantee to read the object data and its metadata.
- **GrantReadACP** (*string*) – Allows grantee to read the object ACL.
- **GrantWriteACP** (*string*) – Allows grantee to write the ACL for the applicable object.
- **Metadata** (*dict*) – A map of metadata to store with the object in S3.
 - (*string*) –
 - * (*string*) –
- **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
- **StorageClass** (*string*) – The type of storage to use for the object. Defaults to 'STANDARD'.
- **WebsiteRedirectLocation** (*string*) – If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.

- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.
- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **SSEKMSKeyId** (*string*) – Specifies the AWS KMS key ID to use for object encryption. All GET and PUT requests for an object protected by AWS KMS will fail if not made via SSL or using SigV4. Documentation on configuring any of the officially supported AWS SDKs and CLI can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingAWSSDK.html#specify-signature-version>
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type s3.MultipartUpload

Returns MultipartUpload resource

put (***kwargs*)

Adds an object to a bucket.

Request Syntax

```
response = object_summary.put(
    ACL='private'|'public-read'|'public-read-write'|'authenticated-read'|'bucket-owner-read',
    Body=b'bytes',
    CacheControl='string',
    ContentDisposition='string',
    ContentEncoding='string',
    ContentLanguage='string',
    ContentLength=123,
    ContentMD5='string',
    ContentType='string',
    Expires=datetime(2015, 1, 1),
    GrantFullControl='string',
    GrantRead='string',
    GrantReadACP='string',
    GrantWriteACP='string',
    Metadata={
        'string': 'string'
    },
    ServerSideEncryption='AES256'|'aws:kms',
    StorageClass='STANDARD'|'REDUCED_REDUNDANCY'|'STANDARD_IA',
    WebsiteRedirectLocation='string',
    SSECustomerAlgorithm='string',
    SSECustomerKey='string',
    SSEKMSKeyId='string',
    RequestPayer='requester'
)
```

Parameters

- **ACL** (*string*) – The canned ACL to apply to the object.
- **Body** (*bytes*) – Object data.
- **CacheControl** (*string*) – Specifies caching behavior along the request/reply chain.
- **ContentDisposition** (*string*) – Specifies presentational information for the object.
- **ContentEncoding** (*string*) – Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.
- **ContentLanguage** (*string*) – The language the content is in.
- **ContentLength** (*integer*) – Size of the body in bytes. This parameter is useful when the size of the body cannot be determined automatically.
- **ContentMD5** (*string*) –
- **ContentType** (*string*) – A standard MIME type describing the format of the object data.
- **Expires** (*datetime*) – The date and time at which the object is no longer cacheable.
- **GrantFullControl** (*string*) – Gives the grantee READ, READ_ACP, and WRITE_ACP permissions on the object.
- **GrantRead** (*string*) – Allows grantee to read the object data and its metadata.
- **GrantReadACP** (*string*) – Allows grantee to read the object ACL.
- **GrantWriteACP** (*string*) – Allows grantee to write the ACL for the applicable object.
- **Metadata** (*dict*) – A map of metadata to store with the object in S3.
 - (*string*) –
 - * (*string*) –
- **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
- **StorageClass** (*string*) – The type of storage to use for the object. Defaults to 'STANDARD'.
- **WebsiteRedirectLocation** (*string*) – If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.
- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **SSEKMSKeyId** (*string*) – Specifies the AWS KMS key ID to use for object encryption. All GET and PUT requests for an object protected by AWS KMS will fail if not made via SSL or using SigV4. Documentation on configuring any of the officially supported AWS SDKs and CLI can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingAWSSDK.html#specify-signature-version>
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need

not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
    'Expiration': 'string',
    'ETag': 'string',
    'ServerSideEncryption': 'AES256'|'aws:kms',
    'VersionId': 'string',
    'SSECustomerAlgorithm': 'string',
    'SSECustomerKeyMD5': 'string',
    'SSEKMSKeyId': 'string',
    'RequestCharged': 'requester'
}
```

Response Structure

- (*dict*) –
 - **Expiration** (*string*) – If the object expiration is configured, this will contain the expiration date (expiry-date) and rule ID (rule-id). The value of rule-id is URL encoded.
 - **ETag** (*string*) – Entity tag for the uploaded object.
 - **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
 - **VersionId** (*string*) – Version of the object.
 - **SSECustomerAlgorithm** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.
 - **SSECustomerKeyMD5** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.
 - **SSEKMSKeyId** (*string*) – If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Acl ()

Creates a ObjectAcl resource.:

```
object_acl = object_summary.Acl()
```

Return type `S3.ObjectAcl`

Returns A ObjectAcl resource

Bucket ()

Creates a Bucket resource.:

```
bucket = object_summary.Bucket()
```

Return type `S3.Bucket`
Returns A Bucket resource

MultipartUpload (*id*)

Creates a MultipartUpload resource.:

```
multipart_upload = object_summary.MultipartUpload('id')
```

Parameters *id* (*string*) – The MultipartUpload’s id identifier. This **must** be set.
Return type `S3.MultipartUpload`
Returns A MultipartUpload resource

Object ()

Creates a Object resource.:

```
object = object_summary.Object()
```

Return type `S3.Object`
Returns A Object resource

Version (*id*)

Creates a ObjectVersion resource.:

```
object_version = object_summary.Version('id')
```

Parameters *id* (*string*) – The Version’s id identifier. This **must** be set.
Return type `S3.ObjectVersion`
Returns A ObjectVersion resource

Waiters

Waiters provide an interface to wait for a resource to reach a specific state. For more information about waiters refer to the [Resources Introduction Guide](#).

wait_until_exists (***kwargs*)

Waits until this ObjectSummary is exists. This method calls `S3.Waiter.object_exists.wait()` which polls. `S3.Client.head_object()` every 5 seconds until a successful state is reached. An error is returned after 20 failed checks.

Request Syntax

```
object_summary.wait_until_exists(
    IfMatch='string',
    IfModifiedSince=datetime(2015, 1, 1),
    IfNoneMatch='string',
    IfUnmodifiedSince=datetime(2015, 1, 1),
    Range='string',
    VersionId='string',
    SSECustomerAlgorithm='string',
    SSECustomerKey='string',
    RequestPayer='requester'
)
```

Parameters

- **IfMatch** (*string*) – Return the object only if its entity tag (ETag) is the same as the one specified, otherwise return a 412 (precondition failed).

- **IfModifiedSince** (*datetime*) – Return the object only if it has been modified since the specified time, otherwise return a 304 (not modified).
- **IfNoneMatch** (*string*) – Return the object only if its entity tag (ETag) is different from the one specified, otherwise return a 304 (not modified).
- **IfUnmodifiedSince** (*datetime*) – Return the object only if it has not been modified since the specified time, otherwise return a 412 (precondition failed).
- **Range** (*string*) – Downloads the specified range bytes of an object. For more information about the HTTP Range header, go to <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35>.
- **VersionId** (*string*) – VersionId used to reference a specific version of the object.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.
- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Returns None

wait_until_not_exists (***kwargs*)

Waits until this ObjectSummary is not exists. This method calls `S3.Waiter.object_not_exists.wait()` which polls `S3.Client.head_object()` every 5 seconds until a successful state is reached. An error is returned after 20 failed checks.

Request Syntax

```
object_summary.wait_until_not_exists(
    IfMatch='string',
    IfModifiedSince=datetime(2015, 1, 1),
    IfNoneMatch='string',
    IfUnmodifiedSince=datetime(2015, 1, 1),
    Range='string',
    VersionId='string',
    SSECustomerAlgorithm='string',
    SSECustomerKey='string',
    RequestPayer='requester'
)
```

Parameters

- **IfMatch** (*string*) – Return the object only if its entity tag (ETag) is the same as the one specified, otherwise return a 412 (precondition failed).
- **IfModifiedSince** (*datetime*) – Return the object only if it has been modified since the specified time, otherwise return a 304 (not modified).
- **IfNoneMatch** (*string*) – Return the object only if its entity tag (ETag) is different from the one specified, otherwise return a 304 (not modified).

- **IfUnmodifiedSince** (*datetime*) – Return the object only if it has not been modified since the specified time, otherwise return a 412 (precondition failed).
- **Range** (*string*) – Downloads the specified range bytes of an object. For more information about the HTTP Range header, go to <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35>.
- **VersionId** (*string*) – VersionId used to reference a specific version of the object.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.
- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Returns None

ObjectVersion

class `S3.ObjectVersion` (*bucket_name, object_key, id*)

A resource representing an Amazon Simple Storage Service (S3) ObjectVersion:

```
import boto3
```

```
s3 = boto3.resource('s3')
object_version = s3.ObjectVersion('bucket_name', 'object_key', 'id')
```

Parameters

- **bucket_name** (*string*) – The ObjectVersion’s bucket_name identifier. This **must** be set.
- **object_key** (*string*) – The ObjectVersion’s object_key identifier. This **must** be set.
- **id** (*string*) – The ObjectVersion’s id identifier. This **must** be set.

These are the resource’s available identifiers:

- `bucket_name`
- `object_key`
- `id`

These are the resource’s available attributes:

- `e_tag`
- `is_latest`
- `key`
- `last_modified`
- `owner`
- `size`
- `storage_class`
- `version_id`

These are the resource's available actions:

- `delete()`
- `get()`
- `head()`

These are the resource's available sub-resources:

- `Object()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

bucket_name

(*string*) The ObjectVersion's bucket_name identifier. This **must** be set.

object_key

(*string*) The ObjectVersion's object_key identifier. This **must** be set.

id

(*string*) The ObjectVersion's id identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

e_tag

(*string*)

is_latest

(*boolean*) Specifies whether the object is (true) or is not (false) the latest version of an object.

key

(*string*) The object key.

last_modified

(*datetime*) Date and time the object was last modified.

owner

(*dict*)

size

(*integer*) Size in bytes of the object.

storage_class

(*string*) The class of storage used to store the object.

version_id

(*string*) Version ID of an object.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

delete (***kwargs*)

Removes the null version (if there is one) of an object and inserts a delete marker, which becomes the latest version of the object. If there isn't a null version, Amazon S3 does not remove any objects.

Request Syntax

```
response = object_version.delete(  
    MFA='string',
```

```
RequestPayer='requester'
)
```

Parameters

- **MFA** (*string*) – The concatenation of the authentication device’s serial number, a space, and the value that is displayed on your authentication device.
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
    'DeleteMarker': True|False,
    'VersionId': 'string',
    'RequestCharged': 'requester'
}
```

Response Structure

- (*dict*) –
 - **DeleteMarker** (*boolean*) – Specifies whether the versioned object that was permanently deleted was (true) or was not (false) a delete marker.
 - **VersionId** (*string*) – Returns the version ID of the delete marker created as a result of the DELETE operation.
 - **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.

get (***kwargs*)

Retrieves objects from Amazon S3.

Request Syntax

```
response = object_version.get(
    IfMatch='string',
    IfModifiedSince=datetime(2015, 1, 1),
    IfNoneMatch='string',
    IfUnmodifiedSince=datetime(2015, 1, 1),
    Range='string',
    ResponseCacheControl='string',
    ResponseContentDisposition='string',
    ResponseContentEncoding='string',
    ResponseContentLanguage='string',
    ResponseContentType='string',
    ResponseExpires=datetime(2015, 1, 1),
    SSECustomerAlgorithm='string',
    SSECustomerKey='string',
    RequestPayer='requester'
)
```

Parameters

- **IfMatch** (*string*) – Return the object only if its entity tag (ETag) is the same as the one specified, otherwise return a 412 (precondition failed).

- **IfModifiedSince** (*datetime*) – Return the object only if it has been modified since the specified time, otherwise return a 304 (not modified).
- **IfNoneMatch** (*string*) – Return the object only if its entity tag (ETag) is different from the one specified, otherwise return a 304 (not modified).
- **IfUnmodifiedSince** (*datetime*) – Return the object only if it has not been modified since the specified time, otherwise return a 412 (precondition failed).
- **Range** (*string*) – Downloads the specified range bytes of an object. For more information about the HTTP Range header, go to <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35>.
- **ResponseCacheControl** (*string*) – Sets the Cache-Control header of the response.
- **ResponseContentDisposition** (*string*) – Sets the Content-Disposition header of the response.
- **ResponseContentEncoding** (*string*) – Sets the Content-Encoding header of the response.
- **ResponseContentLanguage** (*string*) – Sets the Content-Language header of the response.
- **ResponseContentType** (*string*) – Sets the Content-Type header of the response.
- **ResponseExpires** (*datetime*) – Sets the Expires header of the response.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).
- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.
- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required.
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
    'Body': b'bytes',
    'DeleteMarker': True|False,
    'AcceptRanges': 'string',
    'Expiration': 'string',
    'Restore': 'string',
    'LastModified': datetime(2015, 1, 1),
    'ContentLength': 123,
    'ETag': 'string',
    'MissingMeta': 123,
    'VersionId': 'string',
    'CacheControl': 'string',
    'ContentDisposition': 'string',
    'ContentEncoding': 'string',
```

```

'ContentLanguage': 'string',
'ContentRange': 'string',
'ContentType': 'string',
'Expires': datetime(2015, 1, 1),
'WebsiteRedirectLocation': 'string',
'ServerSideEncryption': 'AES256'|'aws:kms',
'Metadata': {
    'string': 'string'
},
'SSECustomerAlgorithm': 'string',
'SSECustomerKeyMD5': 'string',
'SSEKMSKeyId': 'string',
'StorageClass': 'STANDARD'|'REDUCED_REDUNDANCY'|'STANDARD_IA',
'RequestCharged': 'requester',
'ReplicationStatus': 'COMPLETE'|'PENDING'|'FAILED'|'REPLICA'
}

```

Response Structure

- (*dict*) –
 - **Body** (*bytes*) – Object data.
 - **DeleteMarker** (*boolean*) – Specifies whether the object retrieved was (true) or was not (false) a Delete Marker. If false, this response header does not appear in the response.
 - **AcceptRanges** (*string*) –
 - **Expiration** (*string*) – If the object expiration is configured (see PUT Bucket lifecycle), the response includes this header. It includes the expiry-date and rule-id key value pairs providing object expiration information. The value of the rule-id is URL encoded.
 - **Restore** (*string*) – Provides information about object restoration operation and expiration time of the restored object copy.
 - **LastModified** (*datetime*) – Last modified date of the object
 - **ContentLength** (*integer*) – Size of the body in bytes.
 - **ETag** (*string*) – An ETag is an opaque identifier assigned by a web server to a specific version of a resource found at a URL
 - **MissingMeta** (*integer*) – This is set to the number of metadata entries not returned in x-amz-meta headers. This can happen if you create metadata using an API like SOAP that supports more flexible metadata than the REST API. For example, using SOAP, you can create metadata whose values are not legal HTTP headers.
 - **VersionId** (*string*) – Version of the object.
 - **CacheControl** (*string*) – Specifies caching behavior along the request/reply chain.
 - **ContentDisposition** (*string*) – Specifies presentational information for the object.
 - **ContentEncoding** (*string*) – Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.
 - **ContentLanguage** (*string*) – The language the content is in.
 - **ContentRange** (*string*) – The portion of the object returned in the response.
 - **ContentType** (*string*) – A standard MIME type describing the format of the object data.
 - **Expires** (*datetime*) – The date and time at which the object is no longer cacheable.
 - **WebsiteRedirectLocation** (*string*) – If the bucket is configured as a web-

site, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.

- **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
- **Metadata** (*dict*) – A map of metadata to store with the object in S3.
 - * (*string*) –
 - (*string*) –
- **SSECustomerAlgorithm** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.
- **SSECustomerKeyMD5** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.
- **SSEKMSKeyId** (*string*) – If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.
- **StorageClass** (*string*) –
- **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.
- **ReplicationStatus** (*string*) –

head (***kwargs*)

The HEAD operation retrieves metadata from an object without returning the object itself. This operation is useful if you're only interested in an object's metadata. To use HEAD, you must have READ access to the object.

Request Syntax

```
response = object_version.head(
    IfMatch='string',
    IfModifiedSince=datetime(2015, 1, 1),
    IfNoneMatch='string',
    IfUnmodifiedSince=datetime(2015, 1, 1),
    Range='string',
    SSECustomerAlgorithm='string',
    SSECustomerKey='string',
    RequestPayer='requester'
)
```

Parameters

- **IfMatch** (*string*) – Return the object only if its entity tag (ETag) is the same as the one specified, otherwise return a 412 (precondition failed).
- **IfModifiedSince** (*datetime*) – Return the object only if it has been modified since the specified time, otherwise return a 304 (not modified).
- **IfNoneMatch** (*string*) – Return the object only if its entity tag (ETag) is different from the one specified, otherwise return a 304 (not modified).
- **IfUnmodifiedSince** (*datetime*) – Return the object only if it has not been modified since the specified time, otherwise return a 412 (precondition failed).
- **Range** (*string*) – Downloads the specified range bytes of an object. For more information about the HTTP Range header, go to <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35>.
- **SSECustomerAlgorithm** (*string*) – Specifies the algorithm to use to when encrypting the object (e.g., AES256).

- **SSECustomerKey** (*string*) – Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.
- **SSECustomerKeyMD5** (*string*) – Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Please note that this parameter is automatically populated if it is not provided. Including this parameter is not required
- **RequestPayer** (*string*) – Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at <http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html>

Return type dict

Returns

Response Syntax

```
{
  'DeleteMarker': True|False,
  'AcceptRanges': 'string',
  'Expiration': 'string',
  'Restore': 'string',
  'LastModified': datetime(2015, 1, 1),
  'ContentLength': 123,
  'ETag': 'string',
  'MissingMeta': 123,
  'VersionId': 'string',
  'CacheControl': 'string',
  'ContentDisposition': 'string',
  'ContentEncoding': 'string',
  'ContentLanguage': 'string',
  'ContentType': 'string',
  'Expires': datetime(2015, 1, 1),
  'WebsiteRedirectLocation': 'string',
  'ServerSideEncryption': 'AES256'|'aws:kms',
  'Metadata': {
    'string': 'string'
  },
  'SSECustomerAlgorithm': 'string',
  'SSECustomerKeyMD5': 'string',
  'SSEKMSKeyId': 'string',
  'StorageClass': 'STANDARD'|'REDUCED_REDUNDANCY'|'STANDARD_IA',
  'RequestCharged': 'requester',
  'ReplicationStatus': 'COMPLETE'|'PENDING'|'FAILED'|'REPLICA'
}
```

Response Structure

- (*dict*) –
 - **DeleteMarker** (*boolean*) – Specifies whether the object retrieved was (true) or was not (false) a Delete Marker. If false, this response header does not appear in the response.
 - **AcceptRanges** (*string*) –
 - **Expiration** (*string*) – If the object expiration is configured (see PUT Bucket lifecycle), the response includes this header. It includes the expiry-

date and rule-id key value pairs providing object expiration information. The value of the rule-id is URL encoded.

- **Restore** (*string*) – Provides information about object restoration operation and expiration time of the restored object copy.
- **LastModified** (*datetime*) – Last modified date of the object
- **ContentLength** (*integer*) – Size of the body in bytes.
- **ETag** (*string*) – An ETag is an opaque identifier assigned by a web server to a specific version of a resource found at a URL
- **MissingMeta** (*integer*) – This is set to the number of metadata entries not returned in x-amz-meta headers. This can happen if you create metadata using an API like SOAP that supports more flexible metadata than the REST API. For example, using SOAP, you can create metadata whose values are not legal HTTP headers.
- **VersionId** (*string*) – Version of the object.
- **CacheControl** (*string*) – Specifies caching behavior along the request/reply chain.
- **ContentDisposition** (*string*) – Specifies presentational information for the object.
- **ContentEncoding** (*string*) – Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.
- **ContentLanguage** (*string*) – The language the content is in.
- **ContentType** (*string*) – A standard MIME type describing the format of the object data.
- **Expires** (*datetime*) – The date and time at which the object is no longer cacheable.
- **WebsiteRedirectLocation** (*string*) – If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.
- **ServerSideEncryption** (*string*) – The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
- **Metadata** (*dict*) – A map of metadata to store with the object in S3.
 - * (*string*) –
 - (*string*) –
- **SSECustomerAlgorithm** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.
- **SSECustomerKeyMD5** (*string*) – If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.
- **SSEKMSKeyId** (*string*) – If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.
- **StorageClass** (*string*) –
- **RequestCharged** (*string*) – If present, indicates that the requester was successfully charged for the request.
- **ReplicationStatus** (*string*) –

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Object()

Creates a Object resource.:

```
object = object_version.Object()
```

Return type `S3.Object`

Returns A Object resource

SimpleDB**Table of Contents**

- SimpleDB
 - Client
 - Paginators

Client**class SimpleDB.Client**

A low-level client representing Amazon SimpleDB:

```
import boto3
```

```
client = boto3.client('sdb')
```

These are the available methods:

- `batch_delete_attributes()`
- `batch_put_attributes()`
- `can_paginate()`
- `create_domain()`
- `delete_attributes()`
- `delete_domain()`
- `domain_metadata()`
- `generate_presigned_url()`
- `get_attributes()`
- `get_paginator()`
- `get_waiter()`
- `list_domains()`
- `put_attributes()`
- `select()`

batch_delete_attributes(kwargs)**

Performs multiple DeleteAttributes operations in a single call, which reduces round trips and latencies. This enables Amazon SimpleDB to optimize requests, which generally yields better throughput.

The following limitations are enforced for this operation:

- 1 MB request size
- 25 item limit per BatchDeleteAttributes operation

Request Syntax

```
response = client.batch_delete_attributes(
    DomainName='string',
```

```
Items=[
    {
        'Name': 'string',
        'Attributes': [
            {
                'Name': 'string',
                'AlternateNameEncoding': 'string',
                'Value': 'string',
                'AlternateValueEncoding': 'string'
            },
        ]
    },
]
```

Parameters

- **DomainName** (*string*) – **[REQUIRED]** The name of the domain in which the attributes are being deleted.
- **Items** (*list*) – **[REQUIRED]** A list of items on which to perform the operation.
 - (*dict*) –
 - * **Name** (*string*) – **[REQUIRED]**
 - * **Attributes** (*list*) –
 - (*dict*) –
 - **Name** (*string*) – **[REQUIRED]** The name of the attribute.
 - **AlternateNameEncoding** (*string*) –
 - **Value** (*string*) – **[REQUIRED]** The value of the attribute.
 - **AlternateValueEncoding** (*string*) –

Returns None

`batch_put_attributes` (**kwargs)

The `BatchPutAttributes` operation creates or replaces attributes within one or more items. By using this operation, the client can perform multiple `PutAttribute` operation with a single call. This helps yield savings in round trips and latencies, enabling Amazon SimpleDB to optimize requests and generally produce better throughput.

The client may specify the item name with the `Item.X.ItemName` parameter. The client may specify new attributes using a combination of the `Item.X.Attribute.Y.Name` and `Item.X.Attribute.Y.Value` parameters. The client may specify the first attribute for the first item using the parameters `Item.0.Attribute.0.Name` and `Item.0.Attribute.0.Value`, and for the second attribute for the first item by the parameters `Item.0.Attribute.1.Name` and `Item.0.Attribute.1.Value`, and so on.

Attributes are uniquely identified within an item by their name/value combination. For example, a single item can have the attributes { "first_name", "first_value" } and { "first_name", "second_value" }. However, it cannot have two attribute instances where both the `Item.X.Attribute.Y.Name` and `Item.X.Attribute.Y.Value` are the same.

Optionally, the requester can supply the `Replace` parameter for each individual value. Setting this value to `true` will cause the new attribute values to replace the existing attribute values. For example, if an item `I` has the attributes { 'a', '1' }, { 'b', '2' } and { 'b', '3' } and the requester does a `BatchPutAttributes` of { 'I', 'b', '4' } with the `Replace` parameter set to `true`, the final attributes of the item will be { 'a', '1' } and { 'b', '4' }, replacing the previous values of the 'b' attribute with the new value.

Warning: This operation is vulnerable to exceeding the maximum URL size when making a REST request using the HTTP GET method. This operation does not support conditions using `Expected.X.Name`, `Expected.X.Value`, or `Expected.X.Exists`.

You can execute multiple `BatchPutAttributes` operations and other operations in parallel. However, large numbers of concurrent `BatchPutAttributes` calls can result in Service Unavailable (503) responses.

The following limitations are enforced for this operation:

- 256 attribute name-value pairs per item
- 1 MB request size
- 1 billion attributes per domain
- 10 GB of total user data storage per domain
- 25 item limit per `BatchPutAttributes` operation

Request Syntax

```
response = client.batch_put_attributes(
    DomainName='string',
    Items=[
        {
            'Name': 'string',
            'Attributes': [
                {
                    'Name': 'string',
                    'Value': 'string',
                    'Replace': True|False
                },
            ],
        },
    ],
)
```

Parameters

- **DomainName** (*string*) – **[REQUIRED]** The name of the domain in which the attributes are being stored.
- **Items** (*list*) – **[REQUIRED]** A list of items on which to perform the operation.
 - (*dict*) –
 - * **Name** (*string*) – **[REQUIRED]** The name of the replaceable item.
 - * **Attributes** (*list*) – **[REQUIRED]** The list of attributes for a replaceable item.
 - (*dict*) –
 - **Name** (*string*) – **[REQUIRED]** The name of the replaceable attribute.
 - **Value** (*string*) – **[REQUIRED]** The value of the replaceable attribute.
 - **Replace** (*boolean*) – A flag specifying whether or not to replace the attribute/value pair or to add a new attribute/value pair. The default setting is `false`.

Returns None

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

create_domain (**kwargs)

The CreateDomain operation creates a new domain. The domain name should be unique among the domains associated with the Access Key ID provided in the request. The CreateDomain operation may take 10 or more seconds to complete.

The client can create up to 100 domains per account.

If the client requires additional domains, go to <http://aws.amazon.com/contact-us/simplifiedb-limit-request/>.

Request Syntax

```
response = client.create_domain(
    DomainName='string'
)
```

Parameters **DomainName** (*string*) – **[REQUIRED]** The name of the domain to create. The name can range between 3 and 255 characters and can contain the following characters: a-z, A-Z, 0-9, ‘_’, ‘-’, and ‘.’.

Returns None

delete_attributes (**kwargs)

Deletes one or more attributes associated with an item. If all attributes of the item are deleted, the item is deleted.

DeleteAttributes is an idempotent operation; running it multiple times on the same item or attribute does not result in an error response.

Because Amazon SimpleDB makes multiple copies of item data and uses an eventual consistency update model, performing a GetAttributes or Select operation (read) immediately after a DeleteAttributes or PutAttributes operation (write) might not return updated item data.

Request Syntax

```
response = client.delete_attributes(
    DomainName='string',
    ItemName='string',
    Attributes=[
        {
            'Name': 'string',
            'AlternateNameEncoding': 'string',
            'Value': 'string',
            'AlternateValueEncoding': 'string'
        },
    ],
    Expected={
        'Name': 'string',
        'Value': 'string',
        'Exists': True|False
    }
)
```

Parameters

- **DomainName** (*string*) – **[REQUIRED]** The name of the domain in which to perform the operation.
- **ItemName** (*string*) – **[REQUIRED]** The name of the item. Similar to rows on a spreadsheet, items represent individual objects that contain one or more value-attribute pairs.
- **Attributes** (*list*) – A list of Attributes. Similar to columns on a spreadsheet, attributes represent categories of data that can be assigned to items.

- *(dict)* –
 - * **Name** (*string*) – [REQUIRED] The name of the attribute.
 - * **AlternateNameEncoding** (*string*) –
 - * **Value** (*string*) – [REQUIRED] The value of the attribute.
 - * **AlternateValueEncoding** (*string*) –
- **Expected** (*dict*) – The update condition which, if specified, determines whether the specified attributes will be deleted or not. The update condition must be satisfied in order for this request to be processed and the attributes to be deleted.
 - **Name** (*string*) –

The name of the attribute involved in the condition.
 - **Value** (*string*) –

The value of an attribute. This value can only be specified when the `Exists` parameter is equal to `true`.
 - **Exists** (*boolean*) –

A value specifying whether or not the specified attribute must exist with the specified value in order for the update condition to be satisfied. Specify `true` if the attribute must exist for the update condition to be satisfied. Specify `false` if the attribute should not exist in order for the update condition to be satisfied.

Returns None

delete_domain (**kwargs)

The DeleteDomain operation deletes a domain. Any items (and their attributes) in the domain are deleted as well. The DeleteDomain operation might take 10 or more seconds to complete.

Request Syntax

```
response = client.delete_domain(
    DomainName='string'
)
```

Parameters **DomainName** (*string*) – [REQUIRED] The name of the domain to delete.

Returns None

domain_metadata (**kwargs)

Returns information about the domain, including when the domain was created, the number of items and attributes in the domain, and the size of the attribute names and values.

Request Syntax

```
response = client.domain_metadata(
    DomainName='string'
)
```

Parameters **DomainName** (*string*) – [REQUIRED] The name of the domain for which to display the metadata of.

Return type dict

Returns

Response Syntax

```
{
    'ItemCount': 123,
    'ItemNamesSizeBytes': 123,
    'AttributeNameCount': 123,
    'AttributeNamesSizeBytes': 123,
```

```
'AttributeValueCount': 123,  
'AttributeValuesSizeBytes': 123,  
'Timestamp': 123  
}
```

Response Structure

- (*dict*) –
 - **ItemCount** (*integer*) – The number of all items in the domain.
 - **ItemNamesSizeBytes** (*integer*) – The total size of all item names in the domain, in bytes.
 - **AttributeNameCount** (*integer*) – The number of unique attribute names in the domain.
 - **AttributeNamesSizeBytes** (*integer*) – The total size of all unique attribute names in the domain, in bytes.
 - **AttributeValueCount** (*integer*) – The number of all attribute name/value pairs in the domain.
 - **AttributeValuesSizeBytes** (*integer*) – The total size of all attribute values in the domain, in bytes.
 - **Timestamp** (*integer*) – The data and time when metadata was calculated, in Epoch (UNIX) seconds.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_attributes (***kwargs*)

Returns all of the attributes associated with the specified item. Optionally, the attributes returned can be limited to one or more attributes by specifying an attribute name parameter.

If the item does not exist on the replica that was accessed for this operation, an empty set is returned. The system does not return an error as it cannot guarantee the item does not exist on other replicas.

Request Syntax

```
response = client.get_attributes(  
    DomainName='string',  
    ItemName='string',  
    AttributeNames=[  
        'string',  
    ],  
    ConsistentRead=True|False  
)
```

Parameters

- **DomainName** (*string*) – **[REQUIRED]** The name of the domain in which to perform the operation.
- **ItemName** (*string*) – **[REQUIRED]** The name of the item.
- **AttributeNames** (*list*) – The names of the attributes.
 - (*string*) –

- **ConsistentRead** (*boolean*) – Determines whether or not strong consistency should be enforced when data is read from SimpleDB. If `true`, any data previously written to SimpleDB will be returned. Otherwise, results will be consistent eventually, and the client may not see data that was written immediately before your read.

Return type dict

Returns

Response Syntax

```
{
  'Attributes': [
    {
      'Name': 'string',
      'AlternateNameEncoding': 'string',
      'Value': 'string',
      'AlternateValueEncoding': 'string'
    },
  ]
}
```

Response Structure

- (*dict*) –
 - **Attributes** (*list*) – The list of attributes returned by the operation.
 - * (*dict*) –
 - **Name** (*string*) – The name of the attribute.
 - **AlternateNameEncoding** (*string*) –
 - **Value** (*string*) – The value of the attribute.
 - **AlternateValueEncoding** (*string*) –

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type L[botocore.paginate.Paginator]

Returns A paginator object.

get_waiter (*waiter_name*)

list_domains (***kwargs*)

The `ListDomains` operation lists all domains associated with the Access Key ID. It returns domain names up to the limit set by `MaxNumberOfDomains`. A `NextToken` is returned if there are more than `MaxNumberOfDomains` domains. Calling `ListDomains` successive times with the `NextToken` provided by the operation returns up to `MaxNumberOfDomains` more domain names with each successive operation call.

Request Syntax

```
response = client.list_domains(
    MaxNumberOfDomains=123,
    NextToken='string'
)
```

Parameters

- **MaxNumberOfDomains** (*integer*) – The maximum number of domain names you want returned. The range is 1 to 100. The default setting is 100.
- **NextToken** (*string*) – A string informing Amazon SimpleDB where to start the next list of domain names.

Return type dict**Returns****Response Syntax**

```
{
    'DomainNames': [
        'string',
    ],
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **DomainNames** (*list*) – A list of domain names that match the expression.
 - * (*string*) –
 - **NextToken** (*string*) – An opaque token indicating that there are more domains than the specified `MaxNumberOfDomains` still available.

put_attributes (***kwargs*)

The `PutAttributes` operation creates or replaces attributes in an item. The client may specify new attributes using a combination of the `Attribute.X.Name` and `Attribute.X.Value` parameters. The client specifies the first attribute by the parameters `Attribute.0.Name` and `Attribute.0.Value`, the second attribute by the parameters `Attribute.1.Name` and `Attribute.1.Value`, and so on.

Attributes are uniquely identified in an item by their name/value combination. For example, a single item can have the attributes { "first_name", "first_value" } and { "first_name", "second_value" }. However, it cannot have two attribute instances where both the `Attribute.X.Name` and `Attribute.X.Value` are the same.

Optionally, the requestor can supply the `Replace` parameter for each individual attribute. Setting this value to `true` causes the new attribute value to replace the existing attribute value(s). For example, if an item has the attributes { 'a', '1' }, { 'b', '2' } and { 'b', '3' } and the requestor calls `PutAttributes` using the attributes { 'b', '4' } with the `Replace` parameter set to `true`, the final attributes of the item are changed to { 'a', '1' } and { 'b', '4' }, which replaces the previous values of the 'b' attribute with the new value.

You cannot specify an empty string as an attribute name.

Because Amazon SimpleDB makes multiple copies of client data and uses an eventual consistency update model, an immediate `GetAttributes` or `Select` operation (read) immediately after a `PutAttributes` or `DeleteAttributes` operation (write) might not return the updated data.

The following limitations are enforced for this operation:

- 256 total attribute name-value pairs per item
- One billion attributes per domain
- 10 GB of total user data storage per domain

Request Syntax

```
response = client.put_attributes(
    DomainName='string',
    ItemName='string',
    Attributes=[
```



```

        {
            'Name': 'string',
            'Value': 'string',
            'Replace': True|False
        },
    ],
    Expected={
        'Name': 'string',
        'Value': 'string',
        'Exists': True|False
    }
)

```

Parameters

- **DomainName** (*string*) – **[REQUIRED]** The name of the domain in which to perform the operation.
- **ItemName** (*string*) – **[REQUIRED]** The name of the item.
- **Attributes** (*list*) – **[REQUIRED]** The list of attributes.
 - (*dict*) –
 - * **Name** (*string*) – **[REQUIRED]** The name of the replaceable attribute.
 - * **Value** (*string*) – **[REQUIRED]** The value of the replaceable attribute.
 - * **Replace** (*boolean*) – A flag specifying whether or not to replace the attribute/value pair or to add a new attribute/value pair. The default setting is `false`.
- **Expected** (*dict*) – The update condition which, if specified, determines whether the specified attributes will be updated or not. The update condition must be satisfied in order for this request to be processed and the attributes to be updated.
 - **Name** (*string*) –

The name of the attribute involved in the condition.
 - **Value** (*string*) –

The value of an attribute. This value can only be specified when the `Exists` parameter is equal to `true`.
 - **Exists** (*boolean*) –

A value specifying whether or not the specified attribute must exist with the specified value in order for the update condition to be satisfied. Specify `true` if the attribute must exist for the update condition to be satisfied. Specify `false` if the attribute should not exist in order for the update condition to be satisfied.

Returns None

select (***kwargs*)

The `Select` operation returns a set of attributes for `ItemNames` that match the select expression. `Select` is similar to the standard SQL `SELECT` statement.

The total size of the response cannot exceed 1 MB in total size. Amazon SimpleDB automatically adjusts the number of items returned per page to enforce this limit. For example, if the client asks to retrieve 2500 items, but each individual item is 10 kB in size, the system returns 100 items and an appropriate `NextToken` so the client can access the next page of results.

For information on how to construct select expressions, see [Using Select to Create Amazon SimpleDB Queries in the Developer Guide](#).

Request Syntax

```
response = client.select(
    SelectExpression='string',
    NextToken='string',
    ConsistentRead=True|False
)
```

Parameters

- **SelectExpression** (*string*) – [REQUIRED] The expression used to query the domain.
- **NextToken** (*string*) – A string informing Amazon SimpleDB where to start the next list of `ItemNames`.
- **ConsistentRead** (*boolean*) – Determines whether or not strong consistency should be enforced when data is read from SimpleDB. If `true`, any data previously written to SimpleDB will be returned. Otherwise, results will be consistent eventually, and the client may not see data that was written immediately before your read.

Return type dict**Returns****Response Syntax**

```
{
  'Items': [
    {
      'Name': 'string',
      'AlternateNameEncoding': 'string',
      'Attributes': [
        {
          'Name': 'string',
          'AlternateNameEncoding': 'string',
          'Value': 'string',
          'AlternateValueEncoding': 'string'
        },
        ...
      ],
      ...
    },
    ...
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **Items** (*list*) – A list of items that match the select expression.
 - * (*dict*) –
 - **Name** (*string*) – The name of the item.
 - **AlternateNameEncoding** (*string*) –
 - **Attributes** (*list*) – A list of attributes.
 - (*dict*) –
 - **Name** (*string*) – The name of the attribute.
 - **AlternateNameEncoding** (*string*) –
 - **Value** (*string*) – The value of the attribute.
 - **AlternateValueEncoding** (*string*) –
 - **NextToken** (*string*) – An opaque token indicating that more items than `MaxNumberOfItems` were matched, the response size exceeded 1 megabyte, or the execution time exceeded 5 seconds.

Paginators

The available paginators are:

- `SimpleDB.Paginator.ListDomains`
- `SimpleDB.Paginator.Select`

class `SimpleDB.Paginator.ListDomains`

```
paginator = client.get_paginator('list_domains')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `SimpleDB.Client.list_domains()`.

Request Syntax

```
response_iterator = paginator.paginate(
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
    'DomainNames': [
        'string',
    ],
}
```

Response Structure

- (*dict*) –
 - **DomainNames** (*list*) – A list of domain names that match the expression.
 - * (*string*) –

`class SimpleDB.Paginator.Select`

```
paginator = client.get_paginator('select')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `SimpleDB.Client.select()`.

Request Syntax

```
response_iterator = paginator.paginate(
    SelectExpression='string',
    ConsistentRead=True|False,
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **SelectExpression** (*string*) – **[REQUIRED]** The expression used to query the domain.
- **ConsistentRead** (*boolean*) – Determines whether or not strong consistency should be enforced when data is read from SimpleDB. If `true`, any data previously written to SimpleDB will be returned. Otherwise, results will be consistent eventually, and the client may not see data that was written immediately before your read.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Items': [
    {
      'Name': 'string',
      'AlternateNameEncoding': 'string',
      'Attributes': [
        {
          'Name': 'string',
          'AlternateNameEncoding': 'string',
          'Value': 'string',
          'AlternateValueEncoding': 'string'
        }
      ]
    }
  ]
}
```

```

        },
    ],
},
],
}

```

Response Structure

- *(dict)* –
 - **Items** (*list*) – A list of items that match the select expression.
 - * *(dict)* –
 - **Name** (*string*) – The name of the item.
 - **AlternateNameEncoding** (*string*) –
 - **Attributes** (*list*) – A list of attributes.
 - *(dict)* –
 - **Name** (*string*) – The name of the attribute.
 - **AlternateNameEncoding** (*string*) –
 - **Value** (*string*) – The value of the attribute.
 - **AlternateValueEncoding** (*string*) –

SES

Table of Contents

- [SES](#)
 - [Client](#)
 - [Paginators](#)
 - [Waiters](#)

Client

class SES.Client

A low-level client representing Amazon Simple Email Service (SES):

```
import boto3
```

```
client = boto3.client('ses')
```

These are the available methods:

- `can_paginate()`
- `delete_identity()`
- `delete_identity_policy()`
- `delete_verified_email_address()`
- `generate_presigned_url()`
- `get_identity_dkim_attributes()`
- `get_identity_notification_attributes()`
- `get_identity_policies()`
- `get_identity_verification_attributes()`
- `get_paginator()`
- `get_send_quota()`
- `get_send_statistics()`

- `get_waiter()`
- `list_identities()`
- `list_identity_policies()`
- `list_verified_email_addresses()`
- `put_identity_policy()`
- `send_email()`
- `send_raw_email()`
- `set_identity_dkim_enabled()`
- `set_identity_feedback_forwarding_enabled()`
- `set_identity_notification_topic()`
- `verify_domain_dkim()`
- `verify_domain_identity()`
- `verify_email_address()`
- `verify_email_identity()`

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

delete_identity (***kwargs*)

Deletes the specified identity (email address or domain) from the list of verified identities.

This action is throttled at one request per second.

Request Syntax

```
response = client.delete_identity(  
    Identity='string'  
)
```

Parameters *Identity* (*string*) – [REQUIRED]

The identity to be removed from the list of identities for the AWS Account.

Return type dict

Returns

Response Syntax

```
{}
```

Response Structure

- (*dict*) –

An empty element. Receiving this element indicates that the request completed successfully.

delete_identity_policy (***kwargs*)

Deletes the specified sending authorization policy for the given identity (email address or domain). This API returns successfully even if a policy with the specified name does not exist.

Note: This API is for the identity owner only. If you have not verified the identity, this API will return an error.

Sending authorization is a feature that enables an identity owner to authorize other senders to use its identities. For information about using sending authorization, see the [Amazon SES Developer Guide](#).

This action is throttled at one request per second.

Request Syntax

```
response = client.delete_identity_policy(
    Identity='string',
    PolicyName='string'
)
```

Parameters

- **Identity** (*string*) – [REQUIRED]

The identity that is associated with the policy that you want to delete. You can specify the identity by using its name or by using its Amazon Resource Name (ARN). Examples: `user@example.com`, `example.com`, `arn:aws:ses:us-east-1:123456789012:identity/example.com`.

To successfully call this API, you must own the identity.

- **PolicyName** (*string*) – [REQUIRED]

The name of the policy to be deleted.

Return type dict

Returns

Response Syntax

```
{}
```

Response Structure

- (*dict*) –

An empty element. Receiving this element indicates that the request completed successfully.

delete_verified_email_address (***kwargs*)

Deletes the specified email address from the list of verified addresses.

Warning: The DeleteVerifiedEmailAddress action is deprecated as of the May 15, 2012 release of Domain Verification. The DeleteIdentity action is now preferred.

This action is throttled at one request per second.

Request Syntax

```
response = client.delete_verified_email_address(
    EmailAddress='string'
)
```

Parameters

EmailAddress (*string*) – [REQUIRED]

An email address to be removed from the list of verified addresses.

Returns None

generate_presigned_url (*ClientMethod, Params=None, ExpiresIn=3600, HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_identity_dkim_attributes (***kwargs*)

Returns the current status of Easy DKIM signing for an entity. For domain name identities, this action also returns the DKIM tokens that are required for Easy DKIM signing, and whether Amazon SES has successfully verified that these tokens have been published.

This action takes a list of identities as input and returns the following information for each:

- Whether Easy DKIM signing is enabled or disabled.
- A set of DKIM tokens that represent the identity. If the identity is an email address, the tokens represent the domain of that address.
- Whether Amazon SES has successfully verified the DKIM tokens published in the domain's DNS. This information is only returned for domain name identities, not for email addresses.

This action is throttled at one request per second and can only get DKIM attributes for up to 100 identities at a time.

For more information about creating DNS records using DKIM tokens, go to the [Amazon SES Developer Guide](#).

Request Syntax

```
response = client.get_identity_dkim_attributes(
    Identities=[
        'string',
    ]
)
```

Parameters **Identities** (*list*) – [REQUIRED]

A list of one or more verified identities - email addresses, domains, or both.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'DkimAttributes': {
        'string': {
            'DkimEnabled': True|False,
            'DkimVerificationStatus': 'Pending'|'Success'|'Failed'|'TemporaryFailu
            'DkimTokens': [
                'string',
            ]
        }
    }
}
```

Response Structure

- (*dict*) –

Represents a list of all the DKIM attributes for the specified identity.

– **DkimAttributes** (*dict*) –

The DKIM attributes for an email address or a domain.

* (*string*) –

· (*dict*) –

Represents the DKIM attributes of a verified email address or a domain.

· **DkimEnabled** (*boolean*) –

True if DKIM signing is enabled for email sent from the identity; false otherwise.

· **DkimVerificationStatus** (*string*) –

Describes whether Amazon SES has successfully verified the DKIM DNS records (tokens) published in the domain name's DNS. (This only applies to domain identities, not email address identities.)

· **DkimTokens** (*list*) –

A set of character strings that represent the domain's identity. Using these tokens, you will need to create DNS CNAME records that point to DKIM public keys hosted by Amazon SES. Amazon Web Services will eventually detect that you have updated your DNS records; this detection process may take up to 72 hours. Upon successful detection, Amazon SES will be able to DKIM-sign email originating from that domain. (This only applies to domain identities, not email address identities.)

For more information about creating DNS records using DKIM tokens, go to the [Amazon SES Developer Guide](#).

· (*string*) –

get_identity_notification_attributes (***kwargs*)

Given a list of verified identities (email addresses and/or domains), returns a structure describing identity notification attributes.

This action is throttled at one request per second and can only get notification attributes for up to 100 identities at a time.

For more information about using notifications with Amazon SES, see the [Amazon SES Developer Guide](#).

Request Syntax

```
response = client.get_identity_notification_attributes(
    Identities=[
        'string',
    ]
)
```

Parameters **Identities** (*list*) – **[REQUIRED]**

A list of one or more identities. You can specify an identity by using its name or by using its Amazon Resource Name (ARN). Examples: `user@example.com`, `example.com`, `arn:aws:ses:us-east-1:123456789012:identity/example.com`.

· (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'NotificationAttributes': {
    'string': {
      'BounceTopic': 'string',
      'ComplaintTopic': 'string',
      'DeliveryTopic': 'string',
      'ForwardingEnabled': True|False
    }
  }
}
```

Response Structure

- (*dict*) –

Describes whether an identity has Amazon Simple Notification Service (Amazon SNS) topics set for bounce, complaint, and/or delivery notifications, and specifies whether feedback forwarding is enabled for bounce and complaint notifications.

- **NotificationAttributes** (*dict*) –

A map of Identity to IdentityNotificationAttributes.

- * (*string*) –

- (*dict*) –

Represents the notification attributes of an identity, including whether an identity has Amazon Simple Notification Service (Amazon SNS) topics set for bounce, complaint, and/or delivery notifications, and whether feedback forwarding is enabled for bounce and complaint notifications.

- **BounceTopic** (*string*) –

The Amazon Resource Name (ARN) of the Amazon SNS topic where Amazon SES will publish bounce notifications.

- **ComplaintTopic** (*string*) –

The Amazon Resource Name (ARN) of the Amazon SNS topic where Amazon SES will publish complaint notifications.

- **DeliveryTopic** (*string*) –

The Amazon Resource Name (ARN) of the Amazon SNS topic where Amazon SES will publish delivery notifications.

- **ForwardingEnabled** (*boolean*) –

Describes whether Amazon SES will forward bounce and complaint notifications as email. `true` indicates that Amazon SES will forward bounce and complaint notifications as email, while `false` indicates that bounce and complaint notifications will be published only to the specified bounce and complaint Amazon SNS topics.

get_identity_policies (***kwargs*)

Returns the requested sending authorization policies for the given identity (email address or domain). The policies are returned as a map of policy names to policy contents. You can retrieve a maximum of 20 policies at a time.

Note: This API is for the identity owner only. If you have not verified the identity, this API will return an error.

Sending authorization is a feature that enables an identity owner to authorize other senders to use its identities. For information about using sending authorization, see the [Amazon SES Developer Guide](#).

This action is throttled at one request per second.

Request Syntax

```
response = client.get_identity_policies(
    Identity='string',
    PolicyNames=[
        'string',
    ]
)
```

Parameters

- **Identity** (*string*) – [REQUIRED]

The identity for which the policies will be retrieved. You can specify an identity by using its name or by using its Amazon Resource Name (ARN). Examples: `user@example.com`, `example.com`, `arn:aws:ses:us-east-1:123456789012:identity/example.com`.

To successfully call this API, you must own the identity.

- **PolicyNames** (*list*) – [REQUIRED]

A list of the names of policies to be retrieved. You can retrieve a maximum of 20 policies at a time. If you do not know the names of the policies that are attached to the identity, you can use `ListIdentityPolicies`.

- (*string*) – name of the policy.

Return type dict

Returns

Response Syntax

```
{
    'Policies': {
        'string': 'string'
    }
}
```

Response Structure

- (*dict*) –

Represents a map of policy names to policies returned from a successful `GetIdentityPolicies` request.

- **Policies** (*dict*) –

A map of policy names to policies.

- * (*string*) – name of the policy.

- (*string*) – JSON representation of a valid policy.

get_identity_verification_attributes (***kwargs*)

Given a list of identities (email addresses and/or domains), returns the verification status and (for domain identities) the verification token for each identity.

This action is throttled at one request per second and can only get verification attributes for up to 100 identities at a time.

Request Syntax

```
response = client.get_identity_verification_attributes(
    Identities=[
        'string',
    ]
)
```

Parameters **Identities** (*list*) – [REQUIRED]

A list of identities.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'VerificationAttributes': {
        'string': {
            'VerificationStatus': 'Pending'|'Success'|'Failed'|'TemporaryFailure'|
            'VerificationToken': 'string'
        }
    }
}
```

Response Structure

- (*dict*) –

Represents the verification attributes for a list of identities.

– **VerificationAttributes** (*dict*) –

A map of Identities to IdentityVerificationAttributes objects.

* (*string*) –

· (*dict*) –

Represents the verification attributes of a single identity.

- **VerificationStatus** (*string*) –

The verification status of the identity: “Pending”, “Success”, “Failed”, or “TemporaryFailure”.

- **VerificationToken** (*string*) –

The verification token for a domain identity. Null for email address identities.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you’d normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type L[botocore.paginate.Paginator]

Returns A paginator object.

get_send_quota()

Returns the user's current sending limits.

This action is throttled at one request per second.

Request Syntax

```
response = client.get_send_quota()
```

Return type dict

Returns

Response Syntax

```
{
    'Max24HourSend': 123.0,
    'MaxSendRate': 123.0,
    'SentLast24Hours': 123.0
}
```

Response Structure

- (*dict*) –

Represents the user's current activity limits returned from a successful GetSendQuota request.

- **Max24HourSend** (*float*) –

The maximum number of emails the user is allowed to send in a 24-hour interval. A value of -1 signifies an unlimited quota.

- **MaxSendRate** (*float*) –

The maximum number of emails that Amazon SES can accept from the user's account per second.

Note: The rate at which Amazon SES accepts the user's messages might be less than the maximum send rate.

- **SentLast24Hours** (*float*) –

The number of emails sent during the previous 24 hours.

get_send_statistics()

Returns the user's sending statistics. The result is a list of data points, representing the last two weeks of sending activity.

Each data point in the list contains statistics for a 15-minute interval.

This action is throttled at one request per second.

Request Syntax

```
response = client.get_send_statistics()
```

Return type dict

Returns

Response Syntax

```
{
    'SendDataPoints': [
        {
            'Timestamp': datetime(2015, 1, 1),
            'DeliveryAttempts': 123,
            'Bounces': 123,
            'Complaints': 123,
            'Rejects': 123
        },
    ]
}
```

Response Structure

- *(dict)* –

Represents a list of `SendDataPoint` items returned from a successful `GetSendStatistics` request. This list contains aggregated data from the previous two weeks of sending activity.

- **SendDataPoints** (*list*) –

A list of data points, each of which represents 15 minutes of activity.

- * *(dict)* –

Represents sending statistics data. Each `SendDataPoint` contains statistics for a 15-minute period of sending activity.

- **Timestamp** (*datetime*) –

Time of the data point.

- **DeliveryAttempts** (*integer*) –

Number of emails that have been enqueued for sending.

- **Bounces** (*integer*) –

Number of emails that have bounced.

- **Complaints** (*integer*) –

Number of unwanted emails that were rejected by recipients.

- **Rejects** (*integer*) –

Number of emails rejected by Amazon SES.

get_waiter (*waiter_name*)

list_identities (***kwargs*)

Returns a list containing all of the identities (email addresses and domains) for a specific AWS Account, regardless of verification status.

This action is throttled at one request per second.

Request Syntax

```
response = client.list_identities(
    IdentityType='EmailAddress'|'Domain',
    NextToken='string',
    MaxItems=123
)
```

Parameters

- **IdentityType** (*string*) – The type of the identities to list. Possible values are “EmailAddress” and “Domain”. If this parameter is omitted, then all identities will be listed.

- **NextToken** (*string*) – The token to use for pagination.
- **MaxItems** (*integer*) – The maximum number of identities per page. Possible values are 1-1000 inclusive.

Return type dict

Returns

Response Syntax

```
{
  'Identities': [
    'string',
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents a list of all verified identities for the AWS Account.

- **Identities** (*list*) –

A list of identities.

- * (*string*) –

- **NextToken** (*string*) –

The token used for pagination.

list_identity_policies (***kwargs*)

Returns a list of sending authorization policies that are attached to the given identity (email address or domain). This API returns only a list. If you want the actual policy content, you can use `GetIdentityPolicies`.

Note: This API is for the identity owner only. If you have not verified the identity, this API will return an error.

Sending authorization is a feature that enables an identity owner to authorize other senders to use its identities. For information about using sending authorization, see the [Amazon SES Developer Guide](#).

This action is throttled at one request per second.

Request Syntax

```
response = client.list_identity_policies(
    Identity='string'
)
```

Parameters **Identity** (*string*) – [REQUIRED]

The identity that is associated with the policy for which the policies will be listed. You can specify an identity by using its name or by using its Amazon Resource Name (ARN). Examples: `user@example.com`, `example.com`, `arn:aws:ses:us-east-1:123456789012:identity/example.com`.

To successfully call this API, you must own the identity.

Return type dict

Returns

Response Syntax

```
{
    'PolicyNames': [
        'string',
    ]
}
```

Response Structure

- (*dict*) –

Represents a list of policy names returned from a successful `ListIdentityPolicies` request.

- **PolicyNames** (*list*) –

A list of names of policies that apply to the specified identity.

* (*string*) – name of the policy.

list_verified_email_addresses()

Returns a list containing all of the email addresses that have been verified.

Warning: The `ListVerifiedEmailAddresses` action is deprecated as of the May 15, 2012 release of Domain Verification. The `ListIdentities` action is now preferred.

This action is throttled at one request per second.

Request Syntax

```
response = client.list_verified_email_addresses()
```

Return type dict

Returns**Response Syntax**

```
{
    'VerifiedEmailAddresses': [
        'string',
    ]
}
```

Response Structure

- (*dict*) –

Represents a list of all the email addresses verified for the current user.

- **VerifiedEmailAddresses** (*list*) –

A list of email addresses that have been verified.

* (*string*) –

put_identity_policy(kwargs)**

Adds or updates a sending authorization policy for the specified identity (email address or domain).

Note: This API is for the identity owner only. If you have not verified the identity, this API will return an error.

Sending authorization is a feature that enables an identity owner to authorize other senders to use its identities. For information about using sending authorization, see the [Amazon SES Developer Guide](#).

This action is throttled at one request per second.

Request Syntax

```
response = client.put_identity_policy(
    Identity='string',
    PolicyName='string',
    Policy='string'
)
```

Parameters

- **Identity** (*string*) – [REQUIRED]

The identity to which the policy will apply. You can specify an identity by using its name or by using its Amazon Resource Name (ARN). Examples: `user@example.com` , `example.com` , `arn:aws:ses:us-east-1:123456789012:identity/example.com` .

To successfully call this API, you must own the identity.

- **PolicyName** (*string*) – [REQUIRED]

The name of the policy.

The policy name cannot exceed 64 characters and can only include alphanumeric characters, dashes, and underscores.

- **Policy** (*string*) – [REQUIRED]

The text of the policy in JSON format. The policy cannot exceed 4 KB.

For information about the syntax of sending authorization policies, see the [Amazon SES Developer Guide](#) .

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –

An empty element. Receiving this element indicates that the request completed successfully.

send_email (***kwargs*)

Composes an email message based on input data, and then immediately queues the message for sending.

There are several important points to know about `SendEmail` :

- You can only send email from verified email addresses and domains; otherwise, you will get an “Email address not verified” error. If your account is still in the Amazon SES sandbox, you must also verify every recipient email address except for the recipients provided by the Amazon SES mailbox simulator. For more information, go to the [Amazon SES Developer Guide](#) .
- The total size of the message cannot exceed 10 MB. This includes any attachments that are part of the message.
- Amazon SES has a limit on the total number of recipients per message. The combined number of To:, CC: and BCC: email addresses cannot exceed 50. If you need to send an email message to a larger audience, you can divide your recipient list into groups of 50 or fewer, and then call Amazon SES repeatedly to send the message to each group.
- For every message that you send, the total number of recipients (To:, CC: and BCC:) is counted against your sending quota - the maximum number of emails you can send in a 24-hour period. For information about your sending quota, go to the [Amazon SES Developer Guide](#) .

Request Syntax

```
response = client.send_email(  
    Source='string',  
    Destination={  
        'ToAddresses': [  
            'string',  
        ],  
        'CcAddresses': [  
            'string',  
        ],  
        'BccAddresses': [  
            'string',  
        ]  
    },  
    Message={  
        'Subject': {  
            'Data': 'string',  
            'Charset': 'string'  
        },  
        'Body': {  
            'Text': {  
                'Data': 'string',  
                'Charset': 'string'  
            },  
            'Html': {  
                'Data': 'string',  
                'Charset': 'string'  
            }  
        }  
    },  
    ReplyToAddresses=[  
        'string',  
    ],  
    ReturnPath='string',  
    SourceArn='string',  
    ReturnPathArn='string'  
)
```

Parameters

- **Source** (*string*) – [REQUIRED]

The email address that is sending the email. This email address must be either individually verified with Amazon SES, or from a domain that has been verified with Amazon SES. For information about verifying identities, see the [Amazon SES Developer Guide](#).

If you are sending on behalf of another user and have been permitted to do so by a sending authorization policy, then you must also specify the `SourceArn` parameter. For more information about sending authorization, see the [Amazon SES Developer Guide](#).

In all cases, the email address must be 7-bit ASCII. If the text must contain any other characters, then you must use MIME encoded-word syntax (RFC 2047) instead of a literal string. MIME encoded-word syntax uses the following form: `=?charset?encoding?encoded-text?=`. For more information, see [RFC 2047](#).

- **Destination** (*dict*) – [REQUIRED]

The destination for this email, composed of To:, CC:, and BCC: fields.

- **ToAddresses** (*list*) –

The To: field(s) of the message.

* (*string*) –

- **CcAddresses** (*list*) –

The CC: field(s) of the message.

* (*string*) –

- **BccAddresses** (*list*) –

The BCC: field(s) of the message.

* (*string*) –

- **Message** (*dict*) – [REQUIRED]

The message to be sent.

- **Subject** (*dict*) – [REQUIRED]

The subject of the message: A short summary of the content, which will appear in the recipient's inbox.

* **Data** (*string*) – [REQUIRED]

The textual data of the content.

* **Charset** (*string*) –

The character set of the content.

- **Body** (*dict*) – [REQUIRED]

The message body.

* **Text** (*dict*) –

The content of the message, in text format. Use this for text-based email clients, or clients on high-latency networks (such as mobile devices).

· **Data** (*string*) – [REQUIRED]

The textual data of the content.

· **Charset** (*string*) –

The character set of the content.

* **Html** (*dict*) –

The content of the message, in HTML format. Use this for email clients that can process HTML. You can include clickable links, formatted text, and much more in an HTML message.

· **Data** (*string*) – [REQUIRED]

The textual data of the content.

· **Charset** (*string*) –

The character set of the content.

- **ReplyToAddresses** (*list*) – The reply-to email address(es) for the message. If the recipient replies to the message, each reply-to address will receive the reply.
 - (*string*) –
- **ReturnPath** (*string*) – The email address to which bounces and complaints are to be forwarded when feedback forwarding is enabled. If the message cannot be delivered to the recipient, then an error message will be returned from the recipient's ISP; this message will then be forwarded to the email address specified by the `ReturnPath` parameter. The `ReturnPath` parameter is never overwritten. This email address must be either individually verified with Amazon SES, or from a domain that has been verified with Amazon SES.

- **SourceArn** (*string*) – This parameter is used only for sending authorization. It is the ARN of the identity that is associated with the sending authorization policy that permits you to send for the email address specified in the `Source` parameter.

For example, if the owner of `example.com` (which has ARN `arn:aws:ses:us-east-1:123456789012:identity/example.com`) attaches a policy to it that authorizes you to send from `user@example.com`, then you would specify the `SourceArn` to be `arn:aws:ses:us-east-1:123456789012:identity/example.com`, and the `Source` to be `user@example.com`.

For more information about sending authorization, see the [Amazon SES Developer Guide](#).

- **ReturnPathArn** (*string*) – This parameter is used only for sending authorization. It is the ARN of the identity that is associated with the sending authorization policy that permits you to use the email address specified in the `ReturnPath` parameter.

For example, if the owner of `example.com` (which has ARN `arn:aws:ses:us-east-1:123456789012:identity/example.com`) attaches a policy to it that authorizes you to use `feedback@example.com`, then you would specify the `ReturnPathArn` to be `arn:aws:ses:us-east-1:123456789012:identity/example.com`, and the `ReturnPath` to be `feedback@example.com`.

For more information about sending authorization, see the [Amazon SES Developer Guide](#).

Return type dict

Returns

Response Syntax

```
{
    'MessageId': 'string'
}
```

Response Structure

- (*dict*) –

Represents a unique message ID returned from a successful `SendEmail` request.

- **MessageId** (*string*) –

The unique message identifier returned from the `SendEmail` action.

send_raw_email (***kwargs*)

Sends an email message, with header and content specified by the client. The `SendRawEmail` action is useful for sending multipart MIME emails. The raw text of the message must comply with Internet email standards; otherwise, the message cannot be sent.

There are several important points to know about `SendRawEmail`:

- You can only send email from verified email addresses and domains; otherwise, you will get an “Email address not verified” error. If your account is still in the Amazon SES sandbox, you must also verify every recipient email address except for the recipients provided by the Amazon SES mailbox simulator. For more information, go to the [Amazon SES Developer Guide](#).
- The total size of the message cannot exceed 10 MB. This includes any attachments that are part of the message.

- Amazon SES has a limit on the total number of recipients per message. The combined number of To:, CC: and BCC: email addresses cannot exceed 50. If you need to send an email message to a larger audience, you can divide your recipient list into groups of 50 or fewer, and then call Amazon SES repeatedly to send the message to each group.
- The To:, CC:, and BCC: headers in the raw message can contain a group list. Note that each recipient in a group list counts towards the 50-recipient limit.
- For every message that you send, the total number of recipients (To:, CC: and BCC:) is counted against your sending quota - the maximum number of emails you can send in a 24-hour period. For information about your sending quota, go to the [Amazon SES Developer Guide](#).
- If you are using sending authorization to send on behalf of another user, `SendRawEmail` enables you to specify the cross-account identity for the email's "Source," "From," and "Return-Path" parameters in one of two ways: you can pass optional parameters `SourceArn`, `FromArn`, and/or `ReturnPathArn` to the API, or you can include the following X-headers in the header of your raw email: * X-SES-SOURCE-ARN
- X-SES-FROM-ARN
- X-SES-RETURN-PATH-ARN

Warning: Do not include these X-headers in the DKIM signature, because they are removed by Amazon SES before sending the email.

For the most common sending authorization use case, we recommend that you specify the `SourceIdentityArn` and do not specify either the `FromIdentityArn` or `ReturnPathIdentityArn`. (The same note applies to the corresponding X-headers.) If you only specify the `SourceIdentityArn`, Amazon SES will simply set the "From" address and the "Return Path" address to the identity specified in `SourceIdentityArn`. For more information about sending authorization, see the [Amazon SES Developer Guide](#).

Request Syntax

```
response = client.send_raw_email(
    Source='string',
    Destinations=[
        'string',
    ],
    RawMessage={
        'Data': b'bytes'
    },
    FromArn='string',
    SourceArn='string',
    ReturnPathArn='string'
)
```

Parameters

- **Source** (*string*) – The identity's email address. If you do not provide a value for this parameter, you must specify a "From" address in the raw text of the message. (You can also specify both.)

By default, the string must be 7-bit ASCII. If the text must contain any other characters, then you must use MIME encoded-word syntax (RFC 2047) instead of a literal string. MIME encoded-word syntax uses the following form: `=?charset?encoding?encoded-text?=`. For more information, see [RFC 2047](#).

Note: If you specify the `Source` parameter and have feedback forwarding enabled, then bounces and complaints will be sent to this email address. This takes precedence over any *Return-Path* header that you might include in the raw text of the message.

- **Destinations** (*list*) – A list of destinations for the message, consisting of To:, CC:, and BCC: addresses.
 - (*string*) –
- **RawMessage** (*dict*) – **[REQUIRED]**

The raw text of the message. The client is responsible for ensuring the following:

- Message must contain a header and a body, separated by a blank line.
- All required header fields must be present.
- Each part of a multipart MIME message must be formatted properly.
- MIME content types must be among those supported by Amazon SES. For more information, go to the [Amazon SES Developer Guide](#).
- Content must be base64-encoded, if MIME requires it.

- **Data** (*bytes*) – **[REQUIRED]**

The raw data of the message. The client must ensure that the message format complies with Internet email standards regarding email header fields, MIME types, MIME encoding, and base64 encoding (if necessary).

The To:, CC:, and BCC: headers in the raw message can contain a group list.

If you are using `SendRawEmail` with sending authorization, you can include X-headers in the raw message to specify the “Source,” “From,” and “Return-Path” addresses. For more information, see the documentation for `SendRawEmail`.

Warning: Do not include these X-headers in the DKIM signature, because they are removed by Amazon SES before sending the email.

For more information, go to the [Amazon SES Developer Guide](#).

- **FromArn** (*string*) – This parameter is used only for sending authorization. It is the ARN of the identity that is associated with the sending authorization policy that permits you to specify a particular “From” address in the header of the raw email.

Instead of using this parameter, you can use the X-header `X-SES-FROM-ARN` in the raw message of the email. If you use both the `FromArn` parameter and the corresponding X-header, Amazon SES uses the value of the `FromArn` parameter.

Note: For information about when to use this parameter, see the description of `SendRawEmail` in this guide, or see the [Amazon SES Developer Guide](#).

- **SourceArn** (*string*) – This parameter is used only for sending authorization. It is the ARN of the identity that is associated with the sending authorization policy that permits you to send for the email address specified in the `Source` parameter.

For example, if the owner of `example.com` (which has ARN `arn:aws:ses:us-east-1:123456789012:identity/example.com`) attaches a policy to it that authorizes you to send from `user@example.com`, then you would specify the `SourceArn` to be `arn:aws:ses:us-east-1:123456789012:identity/example.com`, and the `Source` to be `user@example.com`.

Instead of using this parameter, you can use the X-header

X-SES-SOURCE-ARN in the raw message of the email. If you use both the `SourceArn` parameter and the corresponding X-header, Amazon SES uses the value of the `SourceArn` parameter.

Note: For information about when to use this parameter, see the description of `SendRawEmail` in this guide, or see the [Amazon SES Developer Guide](#).

- **ReturnPathArn** (*string*) – This parameter is used only for sending authorization. It is the ARN of the identity that is associated with the sending authorization policy that permits you to use the email address specified in the `ReturnPath` parameter.

For example, if the owner of `example.com` (which has ARN `arn:aws:ses:us-east-1:123456789012:identity/example.com`) attaches a policy to it that authorizes you to use `feedback@example.com`, then you would specify the `ReturnPathArn` to be `arn:aws:ses:us-east-1:123456789012:identity/example.com`, and the `ReturnPath` to be `feedback@example.com`.

Instead of using this parameter, you can use the X-header `X-SES-RETURN-PATH-ARN` in the raw message of the email. If you use both the `ReturnPathArn` parameter and the corresponding X-header, Amazon SES uses the value of the `ReturnPathArn` parameter.

Note: For information about when to use this parameter, see the description of `SendRawEmail` in this guide, or see the [Amazon SES Developer Guide](#).

Return type dict

Returns

Response Syntax

```
{
    'MessageId': 'string'
}
```

Response Structure

- (*dict*) –

Represents a unique message ID returned from a successful `SendRawEmail` request.

- **MessageId** (*string*) –

The unique message identifier returned from the `SendRawEmail` action.

set_identity_dkim_enabled (***kwargs*)

Enables or disables Easy DKIM signing of email sent from an identity:

- If Easy DKIM signing is enabled for a domain name identity (e.g., `example.com`), then Amazon SES will DKIM-sign all email sent by addresses under that domain name (e.g., `user@example.com`).
- If Easy DKIM signing is enabled for an email address, then Amazon SES will DKIM-sign all email sent by that email address.

For email addresses (e.g., `user@example.com`), you can only enable Easy DKIM signing if the corresponding domain (e.g., `example.com`) has been set up for Easy DKIM using the AWS Console or the `VerifyDomainDkim` action.

This action is throttled at one request per second.

For more information about Easy DKIM signing, go to the [Amazon SES Developer Guide](#).

Request Syntax

```
response = client.set_identity_dkim_enabled(  
    Identity='string',  
    DkimEnabled=True|False  
)
```

Parameters

- **Identity** (*string*) – [REQUIRED]
The identity for which DKIM signing should be enabled or disabled.
- **DkimEnabled** (*boolean*) – [REQUIRED]
Sets whether DKIM signing is enabled for an identity. Set to `true` to enable DKIM signing for this identity; `false` to disable it.

Return type dict

Returns

Response Syntax

```
{}
```

Response Structure

- (*dict*) –
An empty element. Receiving this element indicates that the request completed successfully.

set_identity_feedback_forwarding_enabled (***kwargs*)

Given an identity (email address or domain), enables or disables whether Amazon SES forwards bounce and complaint notifications as email. Feedback forwarding can only be disabled when Amazon Simple Notification Service (Amazon SNS) topics are specified for both bounces and complaints.

Note: Feedback forwarding does not apply to delivery notifications. Delivery notifications are only available through Amazon SNS.

This action is throttled at one request per second.

For more information about using notifications with Amazon SES, see the [Amazon SES Developer Guide](#).

Request Syntax

```
response = client.set_identity_feedback_forwarding_enabled(  
    Identity='string',  
    ForwardingEnabled=True|False  
)
```

Parameters

- **Identity** (*string*) – [REQUIRED]
The identity for which to set bounce and complaint notification forwarding. Examples: `user@example.com`, `example.com`.
- **ForwardingEnabled** (*boolean*) – [REQUIRED]
Sets whether Amazon SES will forward bounce and complaint notifications as email. `true` specifies that Amazon SES will forward bounce and complaint notifications as email, in addition to any Amazon SNS topic publishing otherwise specified. `false` specifies that Amazon SES will publish bounce and complaint

notifications only through Amazon SNS. This value can only be set to `false` when Amazon SNS topics are set for both `Bounce` and `Complaint` notification types.

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –

An empty element. Receiving this element indicates that the request completed successfully.

`set_identity_notification_topic` (**kwargs)

Given an identity (email address or domain), sets the Amazon Simple Notification Service (Amazon SNS) topic to which Amazon SES will publish bounce, complaint, and/or delivery notifications for emails sent with that identity as the `Source`.

Note: Unless feedback forwarding is enabled, you must specify Amazon SNS topics for bounce and complaint notifications. For more information, see `SetIdentityFeedbackForwardingEnabled`.

This action is throttled at one request per second.

For more information about feedback notification, see the [Amazon SES Developer Guide](#).

Request Syntax

```
response = client.set_identity_notification_topic(
    Identity='string',
    NotificationType='Bounce'|'Complaint'|'Delivery',
    SnsTopic='string'
)
```

Parameters

- **Identity** (*string*) – [REQUIRED]

The identity for which the Amazon SNS topic will be set. You can specify an identity by using its name or by using its Amazon Resource Name (ARN). Examples: `user@example.com`, `example.com`, `arn:aws:ses:us-east-1:123456789012:identity/example.com`.

- **NotificationType** (*string*) – [REQUIRED]

The type of notifications that will be published to the specified Amazon SNS topic.

- **SnsTopic** (*string*) – The Amazon Resource Name (ARN) of the Amazon SNS topic. If the parameter is omitted from the request or a null value is passed, `SnsTopic` is cleared and publishing is disabled.

Return type dict

Returns

Response Syntax

```
{ }
```

Response Structure

- (*dict*) –

An empty element. Receiving this element indicates that the request completed successfully.

verify_domain_dkim (***kwargs*)

Returns a set of DKIM tokens for a domain. DKIM *tokens* are character strings that represent your domain's identity. Using these tokens, you will need to create DNS CNAME records that point to DKIM public keys hosted by Amazon SES. Amazon Web Services will eventually detect that you have updated your DNS records; this detection process may take up to 72 hours. Upon successful detection, Amazon SES will be able to DKIM-sign email originating from that domain.

This action is throttled at one request per second.

To enable or disable Easy DKIM signing for a domain, use the `SetIdentityDkimEnabled` action.

For more information about creating DNS records using DKIM tokens, go to the [Amazon SES Developer Guide](#).

Request Syntax

```
response = client.verify_domain_dkim(
    Domain='string'
)
```

Parameters **Domain** (*string*) – [REQUIRED]

The name of the domain to be verified for Easy DKIM signing.

Return type dict

Returns**Response Syntax**

```
{
    'DkimTokens': [
        'string',
    ]
}
```

Response Structure

- (*dict*) –

Represents the DNS records that must be published in the domain name's DNS to complete DKIM setup.

- **DkimTokens** (*list*) –

A set of character strings that represent the domain's identity. If the identity is an email address, the tokens represent the domain of that address.

Using these tokens, you will need to create DNS CNAME records that point to DKIM public keys hosted by Amazon SES. Amazon Web Services will eventually detect that you have updated your DNS records; this detection process may take up to 72 hours. Upon successful detection, Amazon SES will be able to DKIM-sign emails originating from that domain.

For more information about creating DNS records using DKIM tokens, go to the [Amazon SES Developer Guide](#).

- * (*string*) –

verify_domain_identity (**kwargs)

Verifies a domain.

This action is throttled at one request per second.

Request Syntax

```
response = client.verify_domain_identity(
    Domain='string'
)
```

Parameters **Domain** (*string*) – [REQUIRED]

The domain to be verified.

Return type dict

Returns

Response Syntax

```
{
    'VerificationToken': 'string'
}
```

Response Structure

- (*dict*) –

Represents a token used for domain ownership verification.

- **VerificationToken** (*string*) –

A TXT record that must be placed in the DNS settings for the domain, in order to complete domain verification.

verify_email_address (**kwargs)

Verifies an email address. This action causes a confirmation email message to be sent to the specified address.

Warning: The VerifyEmailAddress action is deprecated as of the May 15, 2012 release of Domain Verification. The VerifyEmailIdentity action is now preferred.

This action is throttled at one request per second.

Request Syntax

```
response = client.verify_email_address(
    EmailAddress='string'
)
```

Parameters **EmailAddress** (*string*) – [REQUIRED]

The email address to be verified.

Returns None

verify_email_identity (**kwargs)

Verifies an email address. This action causes a confirmation email message to be sent to the specified address.

This action is throttled at one request per second.

Request Syntax

```
response = client.verify_email_identity(
    EmailAddress='string'
)
```

Parameters **EmailAddress** (*string*) – [REQUIRED]

The email address to be verified.

Return type dict

Returns

Response Syntax

```
{}
```

Response Structure

- (*dict*) –

An empty element. Receiving this element indicates that the request completed successfully.

Paginators

The available paginators are:

- `SES.Paginator.ListIdentities`

class `SES.Paginator.ListIdentities`

```
paginator = client.get_paginator('list_identities')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `SES.Client.list_identities()`.

Request Syntax

```
response_iterator = paginator.paginate(
    IdentityType='EmailAddress'|'Domain',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **IdentityType** (*string*) – The type of the identities to list. Possible values are “EmailAddress” and “Domain”. If this parameter is omitted, then all identities will be listed.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –
The size of each page.
- **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
    'Identities': [
        'string',
    ],
}
```

Response Structure

- (*dict*) –
Represents a list of all verified identities for the AWS Account.
 - **Identities** (*list*) –
A list of identities.
 - * (*string*) –

Waiters

The available waiters are:

- `SES.Waiter.IdentityExists`

class `SES.Waiter.IdentityExists`

```
waiter = client.get_waiter('identity_exists')
```

wait (***kwargs*)

Polls `SES.Client.get_identity_verification_attributes()` every 3 seconds until a successful state is reached. An error is returned after 20 failed checks.

Request Syntax

```
waiter.wait(
    Identities=[
        'string',
    ]
)
```

Parameters **Identities** (*list*) – [REQUIRED]

A list of identities.

- (*string*) –

Returns None

SNS

Table of Contents

- SNS
 - Client
 - Paginators
 - Service Resource
 - PlatformApplication
 - PlatformEndpoint
 - Subscription
 - Topic

Client

class `SNS.Client`

A low-level client representing Amazon Simple Notification Service (SNS):

```
import boto3
```

```
client = boto3.client('sns')
```

These are the available methods:

- `add_permission()`
- `can_paginate()`
- `confirm_subscription()`
- `create_platform_application()`
- `create_platform_endpoint()`
- `create_topic()`
- `delete_endpoint()`
- `delete_platform_application()`
- `delete_topic()`
- `generate_presigned_url()`
- `get_endpoint_attributes()`
- `get_paginator()`
- `get_platform_application_attributes()`
- `get_subscription_attributes()`
- `get_topic_attributes()`
- `get_waiter()`
- `list_endpoints_by_platform_application()`
- `list_platform_applications()`
- `list_subscriptions()`
- `list_subscriptions_by_topic()`
- `list_topics()`
- `publish()`
- `remove_permission()`
- `set_endpoint_attributes()`
- `set_platform_application_attributes()`
- `set_subscription_attributes()`
- `set_topic_attributes()`
- `subscribe()`
- `unsubscribe()`

add_permission (***kwargs*)

Adds a statement to a topic's access control policy, granting access for the specified AWS accounts to the specified actions.

Request Syntax

```
response = client.add_permission(
    TopicArn='string',
    Label='string',
    AWSAccountId=[
        'string',
    ],
    ActionName=[
        'string',
    ]
)
```

Parameters

- **TopicArn** (*string*) – [REQUIRED]

The ARN of the topic whose access control policy you wish to modify.

- **Label** (*string*) – [REQUIRED]

A unique identifier for the new policy statement.

- **AWSAccountId** (*list*) – [REQUIRED]

The AWS account IDs of the users (principals) who will be given access to the specified actions. The users must have AWS accounts, but do not need to be signed up for this service.

– (*string*) –

- **ActionName** (*list*) – [REQUIRED]

The action you want to allow for the specified principal(s).

Valid values: any Amazon SNS action name.

– (*string*) –

Returns None

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

confirm_subscription (***kwargs*)

Verifies an endpoint owner's intent to receive messages by validating the token sent to the endpoint by an earlier `Subscribe` action. If the token is valid, the action creates a new subscription and returns its Amazon Resource Name (ARN). This call requires an AWS signature only when the `AuthenticateOnUnsubscribe` flag is set to "true".

Request Syntax

```
response = client.confirm_subscription(
    TopicArn='string',
    Token='string',
)
```

```
AuthenticateOnUnsubscribe='string'  
)
```

Parameters

- **TopicArn** (*string*) – [REQUIRED]
The ARN of the topic for which you wish to confirm a subscription.
- **Token** (*string*) – [REQUIRED]
Short-lived token sent to an endpoint during the `Subscribe` action.
- **AuthenticateOnUnsubscribe** (*string*) – Disallows unauthenticated unsubscribes of the subscription. If the value of this parameter is `true` and the request has an AWS signature, then only the topic owner and the subscription owner can unsubscribe the endpoint. The unsubscribe action requires AWS authentication.

Return type dict**Returns****Response Syntax**

```
{  
    'SubscriptionArn': 'string'  
}
```

Response Structure

- (*dict*) – Response for `ConfirmSubscriptions` action.
 - **SubscriptionArn** (*string*) –
The ARN of the created subscription.

create_platform_application (***kwargs*)

Creates a platform application object for one of the supported push notification services, such as APNS and GCM, to which devices and mobile apps may register. You must specify `PlatformPrincipal` and `PlatformCredential` attributes when using the `CreatePlatformApplication` action. The `PlatformPrincipal` is received from the notification service. For APNS/APNS_SANDBOX, `PlatformPrincipal` is “SSL certificate”. For GCM, `PlatformPrincipal` is not applicable. For ADM, `PlatformPrincipal` is “client id”. The `PlatformCredential` is also received from the notification service. For APNS/APNS_SANDBOX, `PlatformCredential` is “private key”. For GCM, `PlatformCredential` is “API key”. For ADM, `PlatformCredential` is “client secret”. The `PlatformApplicationArn` that is returned when using `CreatePlatformApplication` is then used as an attribute for the `CreatePlatformEndpoint` action. For more information, see [Using Amazon SNS Mobile Push Notifications](#).

Request Syntax

```
response = client.create_platform_application(  
    Name='string',  
    Platform='string',  
    Attributes={  
        'string': 'string'  
    }  
)
```

Parameters

- **Name** (*string*) – [REQUIRED]
Application names must be made up of only uppercase and lowercase ASCII letters, numbers, underscores, hyphens, and periods, and must be between 1 and 256 characters long.

- **Platform** (*string*) – [REQUIRED]

The following platforms are supported: ADM (Amazon Device Messaging), APNS (Apple Push Notification Service), APNS_SANDBOX, and GCM (Google Cloud Messaging).

- **Attributes** (*dict*) – [REQUIRED]

For a list of attributes, see [SetPlatformApplicationAttributes](#)

- (*string*) –
- * (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'PlatformApplicationArn': 'string'
}
```

Response Structure

- (*dict*) –

Response from CreatePlatformApplication action.

- **PlatformApplicationArn** (*string*) –

PlatformApplicationArn is returned.

create_platform_endpoint (**kwargs)

Creates an endpoint for a device and mobile app on one of the supported push notification services, such as GCM and APNS. `CreatePlatformEndpoint` requires the `PlatformApplicationArn` that is returned from `CreatePlatformApplication`. The `EndpointArn` that is returned when using `CreatePlatformEndpoint` can then be used by the `Publish` action to send a message to a mobile app or by the `Subscribe` action for subscription to a topic. The `CreatePlatformEndpoint` action is idempotent, so if the requester already owns an endpoint with the same device token and attributes, that endpoint's ARN is returned without creating a new endpoint. For more information, see [Using Amazon SNS Mobile Push Notifications](#).

When using `CreatePlatformEndpoint` with Baidu, two attributes must be provided: `ChannelId` and `UserId`. The token field must also contain the `ChannelId`. For more information, see [Creating an Amazon SNS Endpoint for Baidu](#).

Request Syntax

```
response = client.create_platform_endpoint(
    PlatformApplicationArn='string',
    Token='string',
    CustomUserData='string',
    Attributes={
        'string': 'string'
    }
)
```

Parameters

- **PlatformApplicationArn** (*string*) – [REQUIRED]

PlatformApplicationArn returned from `CreatePlatformApplication` is used to create a an endpoint.

- **Token** (*string*) – [REQUIRED]

Unique identifier created by the notification service for an app on a device. The specific name for Token will vary, depending on which notification service is being used. For example, when using APNS as the notification service, you need the device token. Alternatively, when using GCM or ADM, the device token equivalent is called the registration ID.

- **CustomUserData** (*string*) – Arbitrary user data to associate with the endpoint. Amazon SNS does not use this data. The data must be in UTF-8 format and less than 2KB.
- **Attributes** (*dict*) – For a list of attributes, see [SetEndpointAttributes](#).
 - (*string*) –
 - * (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'EndpointArn': 'string'
}
```

Response Structure

- (*dict*) –

Response from CreateEndpoint action.

- **EndpointArn** (*string*) –

EndpointArn returned from CreateEndpoint action.

create_topic (***kwargs*)

Creates a topic to which notifications can be published. Users can create at most 3000 topics. For more information, see <http://aws.amazon.com/sns>. This action is idempotent, so if the requester already owns a topic with the specified name, that topic's ARN is returned without creating a new topic.

Request Syntax

```
response = client.create_topic(
    Name='string'
)
```

Parameters **Name** (*string*) – **[REQUIRED]**

The name of the topic you want to create.

Constraints: Topic names must be made up of only uppercase and lowercase ASCII letters, numbers, underscores, and hyphens, and must be between 1 and 256 characters long.

Return type dict

Returns

Response Syntax

```
{
    'TopicArn': 'string'
}
```

Response Structure

- (*dict*) –

Response from CreateTopic action.

– **TopicArn** (*string*) –

The Amazon Resource Name (ARN) assigned to the created topic.

delete_endpoint (***kwargs*)

Deletes the endpoint from Amazon SNS. This action is idempotent. For more information, see [Using Amazon SNS Mobile Push Notifications](#).

Request Syntax

```
response = client.delete_endpoint(
    EndpointArn='string'
)
```

Parameters **EndpointArn** (*string*) – [REQUIRED]

EndpointArn of endpoint to delete.

Returns None

delete_platform_application (***kwargs*)

Deletes a platform application object for one of the supported push notification services, such as APNS and GCM. For more information, see [Using Amazon SNS Mobile Push Notifications](#).

Request Syntax

```
response = client.delete_platform_application(
    PlatformApplicationArn='string'
)
```

Parameters **PlatformApplicationArn** (*string*) – [REQUIRED]

PlatformApplicationArn of platform application object to delete.

Returns None

delete_topic (***kwargs*)

Deletes a topic and all its subscriptions. Deleting a topic might prevent some messages previously sent to the topic from being delivered to subscribers. This action is idempotent, so deleting a topic that does not exist does not result in an error.

Request Syntax

```
response = client.delete_topic(
    TopicArn='string'
)
```

Parameters **TopicArn** (*string*) – [REQUIRED]

The ARN of the topic you want to delete.

Returns None

generate_presigned_url (*ClientMethod, Params=None, ExpiresIn=3600, HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_endpoint_attributes (***kwargs*)

Retrieves the endpoint attributes for a device on one of the supported push notification services, such as GCM and APNS. For more information, see [Using Amazon SNS Mobile Push Notifications](#).

Request Syntax

```
response = client.get_endpoint_attributes(
    EndpointArn='string'
)
```

Parameters **EndpointArn** (*string*) – **[REQUIRED]**

EndpointArn for GetEndpointAttributes input.

Return type dict

Returns**Response Syntax**

```
{
    'Attributes': {
        'string': 'string'
    }
}
```

Response Structure

- (*dict*) –

Response from GetEndpointAttributes of the EndpointArn.

- **Attributes** (*dict*) –

Attributes include the following:

- * **CustomUserData** – arbitrary user data to associate with the endpoint. Amazon SNS does not use this data. The data must be in UTF-8 format and less than 2KB.
- * **Enabled** – flag that enables/disables delivery to the endpoint. Amazon SNS will set this to false when a notification service indicates to Amazon SNS that the endpoint is invalid. Users can set it back to true, typically after updating Token.
- * **Token** – device token, also referred to as a registration id, for an app and mobile device. This is returned from the notification service when an app and mobile device are registered with the notification service.
- * (*string*) –
 - (*string*) –

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type L{botocore.paginate.Paginator}

Returns A paginator object.

get_platform_application_attributes (***kwargs*)

Retrieves the attributes of the platform application object for the supported push notification services, such as APNS and GCM. For more information, see [Using Amazon SNS Mobile Push Notifications](#).

Request Syntax

```
response = client.get_platform_application_attributes(
    PlatformApplicationArn='string'
)
```

Parameters **PlatformApplicationArn** (*string*) – [REQUIRED]

PlatformApplicationArn for GetPlatformApplicationAttributesInput.

Return type dict

Returns**Response Syntax**

```
{
    'Attributes': {
        'string': 'string'
    }
}
```

Response Structure

- (*dict*) –

Response for GetPlatformApplicationAttributes action.

- **Attributes** (*dict*) –

Attributes include the following:

- * **EventEndpointCreated** – Topic ARN to which EndpointCreated event notifications should be sent.
- * **EventEndpointDeleted** – Topic ARN to which EndpointDeleted event notifications should be sent.
- * **EventEndpointUpdated** – Topic ARN to which EndpointUpdate event notifications should be sent.
- * **EventDeliveryFailure** – Topic ARN to which DeliveryFailure event notifications should be sent upon Direct Publish delivery failure (permanent) to one of the application’s endpoints.

- * (*string*) –
 - (*string*) –

get_subscription_attributes (***kwargs*)

Returns all of the properties of a subscription.

Request Syntax

```
response = client.get_subscription_attributes(
    SubscriptionArn='string'
)
```

Parameters **SubscriptionArn** (*string*) – [REQUIRED]

The ARN of the subscription whose properties you want to get.

Return type dict

Returns**Response Syntax**

```
{
    'Attributes': {
        'string': 'string'
    }
}
```

Response Structure

- (*dict*) –

Response for GetSubscriptionAttributes action.

- **Attributes** (*dict*) –

A map of the subscription’s attributes. Attributes in this map include the following:

- * **SubscriptionArn** – the subscription’s ARN
- * **TopicArn** – the topic ARN that the subscription is associated with
- * **Owner** – the AWS account ID of the subscription’s owner
- * **ConfirmationWasAuthenticated** – **true** if the subscription confirmation request was authenticated
- * **DeliveryPolicy** – the JSON serialization of the subscription’s delivery policy
- * **EffectiveDeliveryPolicy** – the JSON serialization of the effective delivery policy that takes into account the topic delivery policy and account system defaults
- * (*string*) –
 - (*string*) –

get_topic_attributes (***kwargs*)

Returns all of the properties of a topic. Topic properties returned might differ based on the authorization of the user.

Request Syntax

```
response = client.get_topic_attributes(
    TopicArn='string'
)
```

Parameters **TopicArn** (*string*) – **[REQUIRED]**

The ARN of the topic whose properties you want to get.

Return type *dict*

Returns**Response Syntax**

```
{
    'Attributes': {
        'string': 'string'
    }
}
```

Response Structure

- (*dict*) –

Response for GetTopicAttributes action.

- **Attributes** (*dict*) –

A map of the topic’s attributes. Attributes in this map include the following:

- * `TopicArn` – the topic’s ARN
- * `Owner` – the AWS account ID of the topic’s owner
- * `Policy` – the JSON serialization of the topic’s access control policy
- * `DisplayName` – the human-readable name used in the “From” field for notifications to email and email-json endpoints
- * `SubscriptionsPending` – the number of subscriptions pending confirmation on this topic
- * `SubscriptionsConfirmed` – the number of confirmed subscriptions on this topic
- * `SubscriptionsDeleted` – the number of deleted subscriptions on this topic
- * `DeliveryPolicy` – the JSON serialization of the topic’s delivery policy
- * `EffectiveDeliveryPolicy` – the JSON serialization of the effective delivery policy that takes into account system defaults
- * `(string)` –
 · `(string)` –

get_waiter (*waiter_name*)

list_endpoints_by_platform_application (***kwargs*)

Lists the endpoints and endpoint attributes for devices in a supported push notification service, such as GCM and APNS. The results for `ListEndpointsByPlatformApplication` are paginated and return a limited list of endpoints, up to 100. If additional records are available after the first page results, then a `NextToken` string will be returned. To receive the next page, you call `ListEndpointsByPlatformApplication` again using the `NextToken` string received from the previous call. When there are no more records to return, `NextToken` will be null. For more information, see [Using Amazon SNS Mobile Push Notifications](#).

Request Syntax

```
response = client.list_endpoints_by_platform_application(
    PlatformApplicationArn='string',
    NextToken='string'
)
```

Parameters

- **PlatformApplicationArn** (*string*) – [REQUIRED]

PlatformApplicationArn for ListEndpointsByPlatformApplicationInput action.

- **NextToken** (*string*) – NextToken string is used when calling ListEndpointsByPlatformApplication action to retrieve additional records that are available after the first page results.

Return type dict

Returns

Response Syntax

```
{
  'Endpoints': [
    {
      'EndpointArn': 'string',
      'Attributes': {
```

```
        'string': 'string'
    },
    ],
    'NextToken': 'string'
}
```

Response Structure

- *(dict)* –

Response for ListEndpointsByPlatformApplication action.

- **Endpoints** (*list*) –

Endpoints returned for ListEndpointsByPlatformApplication action.

- * *(dict)* –

Endpoint for mobile app and device.

- **EndpointArn** (*string*) –

EndpointArn for mobile app and device.

- **Attributes** (*dict*) –

Attributes for endpoint.

- (*string*) –

- (*string*) –

- **NextToken** (*string*) –

NextToken string is returned when calling ListEndpointsByPlatformApplication action if additional records are available after the first page results.

list_platform_applications (***kwargs*)

Lists the platform application objects for the supported push notification services, such as APNS and GCM. The results for ListPlatformApplications are paginated and return a limited list of applications, up to 100. If additional records are available after the first page results, then a NextToken string will be returned. To receive the next page, you call ListPlatformApplications using the NextToken string received from the previous call. When there are no more records to return, NextToken will be null. For more information, see [Using Amazon SNS Mobile Push Notifications](#) .

Request Syntax

```
response = client.list_platform_applications(
    NextToken='string'
)
```

Parameters **NextToken** (*string*) – NextToken string is used when calling ListPlatformApplications action to retrieve additional records that are available after the first page results.

Return type dict

Returns

Response Syntax

```
{
    'PlatformApplications': [
        {
            'PlatformApplicationArn': 'string',
            'Attributes': {
                'string': 'string'
            }
        }
    ]
}
```



```

        }
    },
    ],
    'NextToken': 'string'
}

```

Response Structure

- *(dict)* –

Response for ListPlatformApplications action.

- **PlatformApplications** (*list*) –

Platform applications returned when calling ListPlatformApplications action.

- * *(dict)* –

Platform application object.

- **PlatformApplicationArn** (*string*) –

PlatformApplicationArn for platform application object.

- **Attributes** (*dict*) –

Attributes for platform application object.

- (*string*) –

- (*string*) –

- **NextToken** (*string*) –

NextToken string is returned when calling ListPlatformApplications action if additional records are available after the first page results.

list_subscriptions (***kwargs*)

Returns a list of the requester's subscriptions. Each call returns a limited list of subscriptions, up to 100. If there are more subscriptions, a NextToken is also returned. Use the NextToken parameter in a new ListSubscriptions call to get further results.

Request Syntax

```

response = client.list_subscriptions(
    NextToken='string'
)

```

Parameters **NextToken** (*string*) – Token returned by the previous ListSubscriptions request.

Return type dict

Returns

Response Syntax

```

{
    'Subscriptions': [
        {
            'SubscriptionArn': 'string',
            'Owner': 'string',
            'Protocol': 'string',
            'Endpoint': 'string',
            'TopicArn': 'string'
        },
    ],
    'NextToken': 'string'
}

```

Response Structure

- *(dict)* –

Response for ListSubscriptions action

- **Subscriptions** (*list*) –

A list of subscriptions.

- * *(dict)* –

A wrapper type for the attributes of an Amazon SNS subscription.

- **SubscriptionArn** (*string*) –

The subscription's ARN.

- **Owner** (*string*) –

The subscription's owner.

- **Protocol** (*string*) –

The subscription's protocol.

- **Endpoint** (*string*) –

The subscription's endpoint (format depends on the protocol).

- **TopicArn** (*string*) –

The ARN of the subscription's topic.

- **NextToken** (*string*) –

Token to pass along to the next ListSubscriptions request. This element is returned if there are more subscriptions to retrieve.

list_subscriptions_by_topic (***kwargs*)

Returns a list of the subscriptions to a specific topic. Each call returns a limited list of subscriptions, up to 100. If there are more subscriptions, a NextToken is also returned. Use the NextToken parameter in a new ListSubscriptionsByTopic call to get further results.

Request Syntax

```
response = client.list_subscriptions_by_topic(
    TopicArn='string',
    NextToken='string'
)
```

Parameters

- **TopicArn** (*string*) – **[REQUIRED]**

The ARN of the topic for which you wish to find subscriptions.

- **NextToken** (*string*) – Token returned by the previous ListSubscriptionsByTopic request.

Return type dict

Returns**Response Syntax**

```
{
  'Subscriptions': [
    {
      'SubscriptionArn': 'string',
      'Owner': 'string',
      'Protocol': 'string',
      'Endpoint': 'string',
      'TopicArn': 'string'
    }
  ]
}
```

```

        },
    ],
    'NextToken': 'string'
}

```

Response Structure

- *(dict)* –

Response for ListSubscriptionsByTopic action.

- **Subscriptions** (*list*) –

A list of subscriptions.

- * *(dict)* –

A wrapper type for the attributes of an Amazon SNS subscription.

- **SubscriptionArn** (*string*) –

The subscription's ARN.

- **Owner** (*string*) –

The subscription's owner.

- **Protocol** (*string*) –

The subscription's protocol.

- **Endpoint** (*string*) –

The subscription's endpoint (format depends on the protocol).

- **TopicArn** (*string*) –

The ARN of the subscription's topic.

- **NextToken** (*string*) –

Token to pass along to the next ListSubscriptionsByTopic request. This element is returned if there are more subscriptions to retrieve.

list_topics (**kwargs)

Returns a list of the requester's topics. Each call returns a limited list of topics, up to 100. If there are more topics, a NextToken is also returned. Use the NextToken parameter in a new ListTopics call to get further results.

Request Syntax

```

response = client.list_topics(
    NextToken='string'
)

```

Parameters **NextToken** (*string*) – Token returned by the previous ListTopics request.

Return type dict

Returns

Response Syntax

```

{
    'Topics': [
        {
            'TopicArn': 'string'
        },
    ],
    'NextToken': 'string'
}

```

Response Structure

- (*dict*) –

Response for ListTopics action.

- **Topics** (*list*) –

A list of topic ARNs.

- * (*dict*) –

A wrapper type for the topic's Amazon Resource Name (ARN). To retrieve a topic's attributes, use `GetTopicAttributes`.

- **TopicArn** (*string*) –

The topic's ARN.

- **NextToken** (*string*) –

Token to pass along to the next `ListTopics` request. This element is returned if there are additional topics to retrieve.

publish (***kwargs*)

Sends a message to all of a topic's subscribed endpoints. When a `messageId` is returned, the message has been saved and Amazon SNS will attempt to deliver it to the topic's subscribers shortly. The format of the outgoing message to each subscribed endpoint depends on the notification protocol selected.

To use the `Publish` action for sending a message to a mobile endpoint, such as an app on a Kindle device or mobile phone, you must specify the `EndpointArn`. The `EndpointArn` is returned when making a call with the `CreatePlatformEndpoint` action. The second example below shows a request and response for publishing to a mobile endpoint.

Request Syntax

```
response = client.publish(
    TopicArn='string',
    TargetArn='string',
    Message='string',
    Subject='string',
    MessageStructure='string',
    MessageAttributes={
        'string': {
            'DataType': 'string',
            'StringValue': 'string',
            'BinaryValue': b'bytes'
        }
    }
)
```

Parameters

- **TopicArn** (*string*) – The topic you want to publish to.
- **TargetArn** (*string*) – Either `TopicArn` or `EndpointArn`, but not both.
- **Message** (*string*) – **[REQUIRED]**

The message you want to send to the topic.

If you want to send the same message to all transport protocols, include the text of the message as a `String` value.

If you want to send different messages for each transport protocol, set the value of the `MessageStructure` parameter to `json` and use a JSON object for the `Message` parameter. See the `Examples` section for the format of the JSON object.

Constraints: Messages must be UTF-8 encoded strings at most 256 KB in size (262144 bytes, not 262144 characters).

JSON-specific constraints:

- Keys in the JSON object that correspond to supported transport protocols must have simple JSON string values.
 - The values will be parsed (unescaped) before they are used in outgoing messages.
 - Outbound notifications are JSON encoded (meaning that the characters will be reescaped for sending).
 - Values have a minimum length of 0 (the empty string, “”, is allowed).
 - Values have a maximum length bounded by the overall message size (so, including multiple protocols may limit message sizes).
 - Non-string values will cause the key to be ignored.
 - Keys that do not correspond to supported transport protocols are ignored.
 - Duplicate keys are not allowed.
 - Failure to parse or validate any key or value in the message will cause the `Publish` call to return an error (no partial delivery).
- **Subject** (*string*) – Optional parameter to be used as the “Subject” line when the message is delivered to email endpoints. This field will also be included, if present, in the standard JSON messages delivered to other endpoints.

Constraints: Subjects must be ASCII text that begins with a letter, number, or punctuation mark; must not include line breaks or control characters; and must be less than 100 characters long.

- **MessageStructure** (*string*) – Set `MessageStructure` to `json` if you want to send a different message for each protocol. For example, using one `publish` action, you can send a short message to your SMS subscribers and a longer message to your email subscribers. If you set `MessageStructure` to `json`, the value of the `Message` parameter must:
 - be a syntactically valid JSON object; and
 - contain at least a top-level JSON key of “default” with a value that is a string.

You can define other top-level keys that define the message you want to send to a specific transport protocol (e.g., “http”).

For information about sending different messages for each protocol using the AWS Management Console, go to [Create Different Messages for Each Protocol](#) in the *Amazon Simple Notification Service Getting Started Guide*.

Valid value: `json`

- **MessageAttributes** (*dict*) – Message attributes for `Publish` action.
 - (*string*) –
 - * (*dict*) –

The user-specified message attribute value. For string data types, the value attribute has the same restrictions on the content as the message body. For more information, see [Publish](#).

Name, type, and value must not be empty or null. In addition, the message body should not be empty or null. All parts of the message attribute, including name, type, and value, are included in the message size restriction, which is currently 256 KB (262,144 bytes). For more information, see [Using Amazon SNS Message Attributes](#).

- **DataType** (*string*) – [REQUIRED]

Amazon SNS supports the following logical data types: String, Number, and Binary. For more information, see [Message Attribute Data Types](#) .

- **StringValue** (*string*) –

Strings are Unicode with UTF8 binary encoding. For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters .

- **BinaryValue** (*bytes*) –

Binary type attributes can store any binary data, for example, compressed data, encrypted data, or images.

Return type dict

Returns

Response Syntax

```
{
    'MessageId': 'string'
}
```

Response Structure

- (*dict*) –

Response for Publish action.

- **MessageId** (*string*) –

Unique identifier assigned to the published message.

Length Constraint: Maximum 100 characters

remove_permission (***kwargs*)

Removes a statement from a topic's access control policy.

Request Syntax

```
response = client.remove_permission(
    TopicArn='string',
    Label='string'
)
```

Parameters

- **TopicArn** (*string*) – **[REQUIRED]**

The ARN of the topic whose access control policy you wish to modify.

- **Label** (*string*) – **[REQUIRED]**

The unique label of the statement you want to remove.

Returns None

set_endpoint_attributes (***kwargs*)

Sets the attributes for an endpoint for a device on one of the supported push notification services, such as GCM and APNS. For more information, see [Using Amazon SNS Mobile Push Notifications](#) .

Request Syntax

```
response = client.set_endpoint_attributes(
    EndpointArn='string',
    Attributes={
        'string': 'string'
    }
)
```

```
    }
)
```

Parameters

- **EndpointArn** (*string*) – [REQUIRED]

EndpointArn used for SetEndpointAttributes action.

- **Attributes** (*dict*) – [REQUIRED]

A map of the endpoint attributes. Attributes in this map include the following:

- CustomUserData – arbitrary user data to associate with the endpoint. Amazon SNS does not use this data. The data must be in UTF-8 format and less than 2KB.
- Enabled – flag that enables/disables delivery to the endpoint. Amazon SNS will set this to false when a notification service indicates to Amazon SNS that the endpoint is invalid. Users can set it back to true, typically after updating Token.
- Token – device token, also referred to as a registration id, for an app and mobile device. This is returned from the notification service when an app and mobile device are registered with the notification service.
- (*string*) –
 * (*string*) –

Returns None

set_platform_application_attributes (***kwargs*)

Sets the attributes of the platform application object for the supported push notification services, such as APNS and GCM. For more information, see [Using Amazon SNS Mobile Push Notifications](#).

Request Syntax

```
response = client.set_platform_application_attributes(
    PlatformApplicationArn='string',
    Attributes={
        'string': 'string'
    }
)
```

Parameters

- **PlatformApplicationArn** (*string*) – [REQUIRED]

PlatformApplicationArn for SetPlatformApplicationAttributes action.

- **Attributes** (*dict*) – [REQUIRED]

A map of the platform application attributes. Attributes in this map include the following:

- PlatformCredential – The credential received from the notification service. For APNS/APNS_SANDBOX, PlatformCredential is “private key”. For GCM, PlatformCredential is “API key”. For ADM, PlatformCredential is “client secret”.
- PlatformPrincipal – The principal received from the notification service. For APNS/APNS_SANDBOX, PlatformPrincipal is “SSL certificate”. For GCM, PlatformPrincipal is not applicable. For ADM, PlatformPrincipal is “client id”.
- EventEndpointCreated – Topic ARN to which EndpointCreated event notifications should be sent.
- EventEndpointDeleted – Topic ARN to which EndpointDeleted event notifications should be sent.

- EventEndpointUpdated – Topic ARN to which EndpointUpdate event notifications should be sent.
- EventDeliveryFailure – Topic ARN to which DeliveryFailure event notifications should be sent upon Direct Publish delivery failure (permanent) to one of the application’s endpoints.
- (*string*) –
 - * (*string*) –

Returns None

set_subscription_attributes (***kwargs*)

Allows a subscription owner to set an attribute of the topic to a new value.

Request Syntax

```
response = client.set_subscription_attributes(
    SubscriptionArn='string',
    AttributeName='string',
    AttributeValue='string'
)
```

Parameters

- **SubscriptionArn** (*string*) – **[REQUIRED]**
The ARN of the subscription to modify.
- **AttributeName** (*string*) – **[REQUIRED]**
The name of the attribute you want to set. Only a subset of the subscriptions attributes are mutable.

Valid values: `DeliveryPolicy` | `RawMessageDelivery`
- **AttributeValue** (*string*) – The new value for the attribute in JSON format.

Returns None

set_topic_attributes (***kwargs*)

Allows a topic owner to set an attribute of the topic to a new value.

Request Syntax

```
response = client.set_topic_attributes(
    TopicArn='string',
    AttributeName='string',
    AttributeValue='string'
)
```

Parameters

- **TopicArn** (*string*) – **[REQUIRED]**
The ARN of the topic to modify.
- **AttributeName** (*string*) – **[REQUIRED]**
The name of the attribute you want to set. Only a subset of the topic’s attributes are mutable.

Valid values: `Policy` | `DisplayName` | `DeliveryPolicy`
- **AttributeValue** (*string*) – The new value for the attribute.

Returns None

subscribe (***kwargs*)

Prepares to subscribe an endpoint by sending the endpoint a confirmation message. To actually create a

subscription, the endpoint owner must call the `ConfirmSubscription` action with the token from the confirmation message. Confirmation tokens are valid for three days.

Request Syntax

```
response = client.subscribe(
    TopicArn='string',
    Protocol='string',
    Endpoint='string'
)
```

Parameters

- **TopicArn** (*string*) – [REQUIRED]

The ARN of the topic you want to subscribe to.

- **Protocol** (*string*) – [REQUIRED]

The protocol you want to use. Supported protocols include:

- `http` – delivery of JSON-encoded message via HTTP POST
- `https` – delivery of JSON-encoded message via HTTPS POST
- `email` – delivery of message via SMTP
- `email-json` – delivery of JSON-encoded message via SMTP
- `sms` – delivery of message via SMS
- `sqs` – delivery of JSON-encoded message to an Amazon SQS queue
- `application` – delivery of JSON-encoded message to an EndpointArn for a mobile app and device.

- **Endpoint** (*string*) – The endpoint that you want to receive notifications. Endpoints vary by protocol:

- For the `http` protocol, the endpoint is an URL beginning with “`http://`”
- For the `https` protocol, the endpoint is a URL beginning with “`https://`”
- For the `email` protocol, the endpoint is an email address
- For the `email-json` protocol, the endpoint is an email address
- For the `sms` protocol, the endpoint is a phone number of an SMS-enabled device
- For the `sqs` protocol, the endpoint is the ARN of an Amazon SQS queue
- For the `application` protocol, the endpoint is the EndpointArn of a mobile app and device.

Return type dict

Returns

Response Syntax

```
{
    'SubscriptionArn': 'string'
}
```

Response Structure

- (*dict*) – Response for Subscribe action.
 - **SubscriptionArn** (*string*) –

The ARN of the subscription, if the service was able to create a subscription immediately (without requiring endpoint owner confirmation).

unsubscribe (***kwargs*)

Deletes a subscription. If the subscription requires authentication for deletion, only the owner of the subscription or the topic’s owner can unsubscribe, and an AWS signature is required. If the `Unsubscribe` call does not require authentication and the requester is not the subscription owner, a final cancellation

message is delivered to the endpoint, so that the endpoint owner can easily resubscribe to the topic if the Unsubscribe request was unintended.

Request Syntax

```
response = client.unsubscribe(
    SubscriptionArn='string'
)
```

Parameters **SubscriptionArn** (*string*) – [REQUIRED]

The ARN of the subscription to be deleted.

Returns None

Paginators

The available paginators are:

- `SNS.Paginator.ListEndpointsByPlatformApplication`
- `SNS.Paginator.ListPlatformApplications`
- `SNS.Paginator.ListSubscriptions`
- `SNS.Paginator.ListSubscriptionsByTopic`
- `SNS.Paginator.ListTopics`

class `SNS.Paginator.ListEndpointsByPlatformApplication`

```
paginator = client.get_paginator('list_endpoints_by_platform_application')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `SNS.Client.list_endpoints_by_platform_application()`.

Request Syntax

```
response_iterator = paginator.paginate(
    PlatformApplicationArn='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **PlatformApplicationArn** (*string*) – [REQUIRED]

PlatformApplicationArn for ListEndpointsByPlatformApplicationInput action.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Endpoints': [
    {
      'EndpointArn': 'string',
      'Attributes': {
        'string': 'string'
      }
    },
  ],
}
```

Response Structure

- (*dict*) –

Response for `ListEndpointsByPlatformApplication` action.

- **Endpoints** (*list*) –

Endpoints returned for `ListEndpointsByPlatformApplication` action.

- * (*dict*) –

Endpoint for mobile app and device.

- **EndpointArn** (*string*) –

EndpointArn for mobile app and device.

- **Attributes** (*dict*) –

Attributes for endpoint.

- (*string*) –

- (*string*) –

class `SNS.Paginator.ListPlatformApplications`

```
paginator = client.get_paginator('list_platform_applications')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `SNS.Client.list_platform_applications()`.

Request Syntax

```
response_iterator = paginator.paginate(
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
```

```
}  
)
```

Parameters **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{  
    'PlatformApplications': [  
        {  
            'PlatformApplicationArn': 'string',  
            'Attributes': {  
                'string': 'string'  
            }  
        },  
    ],  
}
```

Response Structure

- (*dict*) –

Response for ListPlatformApplications action.

- **PlatformApplications** (*list*) –

Platform applications returned when calling ListPlatformApplications action.

- * (*dict*) –

Platform application object.

- **PlatformApplicationArn** (*string*) –

PlatformApplicationArn for platform application object.

- **Attributes** (*dict*) –

Attributes for platform application object.

- (*string*) –

- (*string*) –

class `SNS.Paginator.ListSubscriptions`

```
paginator = client.get_paginator('list_subscriptions')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `SNS.Client.list_subscriptions()`.

Request Syntax

```
response_iterator = paginator.paginate(
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns**Response Syntax**

```
{
    'Subscriptions': [
        {
            'SubscriptionArn': 'string',
            'Owner': 'string',
            'Protocol': 'string',
            'Endpoint': 'string',
            'TopicArn': 'string'
        },
    ],
}
```

Response Structure

- (*dict*) –

Response for ListSubscriptions action

- **Subscriptions** (*list*) –

A list of subscriptions.

- * (*dict*) –

A wrapper type for the attributes of an Amazon SNS subscription.

- **SubscriptionArn** (*string*) –

The subscription's ARN.

- **Owner** (*string*) –
The subscription's owner.
- **Protocol** (*string*) –
The subscription's protocol.
- **Endpoint** (*string*) –
The subscription's endpoint (format depends on the protocol).
- **TopicArn** (*string*) –
The ARN of the subscription's topic.

class SNS.Paginator.**ListSubscriptionsByTopic**

```
paginator = client.get_paginator('list_subscriptions_by_topic')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `SNS.Client.list_subscriptions_by_topic()`.

Request Syntax

```
response_iterator = paginator.paginate(
    TopicArn='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **TopicArn** (*string*) – **[REQUIRED]**
The ARN of the topic for which you wish to find subscriptions.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
    'Subscriptions': [
        {
            'SubscriptionArn': 'string',
```

```

        'Owner': 'string',
        'Protocol': 'string',
        'Endpoint': 'string',
        'TopicArn': 'string'
    },
],
}

```

Response Structure

- (*dict*) –

Response for ListSubscriptionsByTopic action.

- **Subscriptions** (*list*) –

A list of subscriptions.

- * (*dict*) –

A wrapper type for the attributes of an Amazon SNS subscription.

- **SubscriptionArn** (*string*) –

The subscription's ARN.

- **Owner** (*string*) –

The subscription's owner.

- **Protocol** (*string*) –

The subscription's protocol.

- **Endpoint** (*string*) –

The subscription's endpoint (format depends on the protocol).

- **TopicArn** (*string*) –

The ARN of the subscription's topic.

class SNS.Paginator.**ListTopics**

```
paginator = client.get_paginator('list_topics')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `SNS.Client.list_topics()`.

Request Syntax

```

response_iterator = paginator.paginate(
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)

```

Parameters **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –
The size of each page.
- **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
    'Topics': [
        {
            'TopicArn': 'string'
        },
    ],
}
```

Response Structure

- (*dict*) –
Response for ListTopics action.
 - **Topics** (*list*) –
A list of topic ARNs.
 - * (*dict*) –
A wrapper type for the topic's Amazon Resource Name (ARN). To retrieve a topic's attributes, use `GetTopicAttributes`.
 - **TopicArn** (*string*) –
The topic's ARN.

Service Resource

`class SNS.ServiceResource`

A resource representing Amazon Simple Notification Service (SNS):

```
import boto3
```

```
sns = boto3.resource('sns')
```

These are the resource's available actions:

- `create_platform_application()`
- `create_topic()`

These are the resource's available sub-resources:

- `PlatformApplication()`
- `PlatformEndpoint()`
- `Subscription()`
- `Topic()`

These are the resource's available collections:

- `platform_applications`
- `subscriptions`
- `topics`

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

`create_platform_application` (**kwargs)

Creates a platform application object for one of the supported push notification services, such as APNS and GCM, to which devices and mobile apps may register. You must specify PlatformPrincipal and PlatformCredential attributes when using the `CreatePlatformApplication` action. The PlatformPrincipal is received from the notification service. For APNS/APNS_SANDBOX, PlatformPrincipal is “SSL certificate”. For GCM, PlatformPrincipal is not applicable. For ADM, PlatformPrincipal is “client id”. The PlatformCredential is also received from the notification service. For APNS/APNS_SANDBOX, PlatformCredential is “private key”. For GCM, PlatformCredential is “API key”. For ADM, PlatformCredential is “client secret”. The PlatformApplicationArn that is returned when using `CreatePlatformApplication` is then used as an attribute for the `CreatePlatformEndpoint` action. For more information, see [Using Amazon SNS Mobile Push Notifications](#).

Request Syntax

```
platform_application = sns.create_platform_application(
    Name='string',
    Platform='string',
    Attributes={
        'string': 'string'
    }
)
```

Parameters

- **Name** (*string*) – [REQUIRED]

Application names must be made up of only uppercase and lowercase ASCII letters, numbers, underscores, hyphens, and periods, and must be between 1 and 256 characters long.

- **Platform** (*string*) – [REQUIRED]

The following platforms are supported: ADM (Amazon Device Messaging), APNS (Apple Push Notification Service), APNS_SANDBOX, and GCM (Google Cloud Messaging).

- **Attributes** (*dict*) – [REQUIRED]

For a list of attributes, see [SetPlatformApplicationAttributes](#)

– (*string*) –
* (*string*) –

Return type `sns.PlatformApplication`

Returns PlatformApplication resource

`create_topic` (**kwargs)

Creates a topic to which notifications can be published. Users can create at most 3000 topics. For more information, see <http://aws.amazon.com/sns>. This action is idempotent, so if the requester already owns a topic with the specified name, that topic’s ARN is returned without creating a new topic.

Request Syntax

```
topic = sns.create_topic(
    Name='string'
)
```

Parameters `Name (string)` – [REQUIRED]

The name of the topic you want to create.

Constraints: Topic names must be made up of only uppercase and lowercase ASCII letters, numbers, underscores, and hyphens, and must be between 1 and 256 characters long.

Return type `sns.Topic`**Returns** Topic resource

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

PlatformApplication (`arn`)

Creates a PlatformApplication resource.:

```
platform_application = sns.PlatformApplication('arn')
```

Parameters `arn (string)` – The PlatformApplication's arn identifier. This **must** be set.**Return type** `SNS.PlatformApplication`**Returns** A PlatformApplication resource**PlatformEndpoint** (`arn`)

Creates a PlatformEndpoint resource.:

```
platform_endpoint = sns.PlatformEndpoint('arn')
```

Parameters `arn (string)` – The PlatformEndpoint's arn identifier. This **must** be set.**Return type** `SNS.PlatformEndpoint`**Returns** A PlatformEndpoint resource**Subscription** (`arn`)

Creates a Subscription resource.:

```
subscription = sns.Subscription('arn')
```

Parameters `arn (string)` – The Subscription's arn identifier. This **must** be set.**Return type** `SNS.Subscription`**Returns** A Subscription resource**Topic** (`arn`)

Creates a Topic resource.:

```
topic = sns.Topic('arn')
```

Parameters `arn (string)` – The Topic's arn identifier. This **must** be set.**Return type** `SNS.Topic`**Returns** A Topic resource

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

`platform_applications`

all()

Creates an iterable of all PlatformApplication resources in the collection.

Request Syntax

```
platform_application_iterator = sns.platform_applications.all()
```

Return type list(sns.PlatformApplication)

Returns A list of PlatformApplication resources

filter(kwargs)**

Creates an iterable of all PlatformApplication resources in the collection filtered by kwargs passed to method.

Request Syntax

```
platform_application_iterator = sns.platform_applications.filter(
    NextToken='string'
)
```

Parameters **NextToken** (*string*) – NextToken string is used when calling ListPlatformApplications action to retrieve additional records that are available after the first page results.

Return type list(sns.PlatformApplication)

Returns A list of PlatformApplication resources

limit(kwargs)**

Creates an iterable up to a specified amount of PlatformApplication resources in the collection.

Request Syntax

```
platform_application_iterator = sns.platform_applications.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type list(sns.PlatformApplication)

Returns A list of PlatformApplication resources

page_size(kwargs)**

Creates an iterable of all PlatformApplication resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
platform_application_iterator = sns.platform_applications.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type list(sns.PlatformApplication)

Returns A list of PlatformApplication resources

subscriptions

all()

Creates an iterable of all Subscription resources in the collection.

Request Syntax

```
subscription_iterator = sns.subscriptions.all()
```

Return type list(sns.Subscription)

Returns A list of Subscription resources

filter (**kwargs)

Creates an iterable of all Subscription resources in the collection filtered by kwargs passed to method.

Request Syntax

```
subscription_iterator = sns.subscriptions.filter(  
    NextToken='string'  
)
```

Parameters **NextToken** (*string*) – Token returned by the previous ListSubscriptions request.

Return type list(sns.Subscription)

Returns A list of Subscription resources

limit (**kwargs)

Creates an iterable up to a specified amount of Subscription resources in the collection.

Request Syntax

```
subscription_iterator = sns.subscriptions.limit(  
    count=123  
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type list(sns.Subscription)

Returns A list of Subscription resources

page_size (**kwargs)

Creates an iterable of all Subscription resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
subscription_iterator = sns.subscriptions.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type list(sns.Subscription)

Returns A list of Subscription resources

topics

all ()

Creates an iterable of all Topic resources in the collection.

Request Syntax

```
topic_iterator = sns.topics.all()
```

Return type list(sns.Topic)

Returns A list of Topic resources

filter (***kwargs*)

Creates an iterable of all Topic resources in the collection filtered by kwargs passed to method.

Request Syntax

```
topic_iterator = sns.topics.filter(
    NextToken='string'
)
```

Parameters **NextToken** (*string*) – Token returned by the previous `ListTopics` request.

Return type `list(sns.Topic)`

Returns A list of Topic resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Topic resources in the collection.

Request Syntax

```
topic_iterator = sns.topics.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(sns.Topic)`

Returns A list of Topic resources

page_size (***kwargs*)

Creates an iterable of all Topic resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
topic_iterator = sns.topics.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(sns.Topic)`

Returns A list of Topic resources

PlatformApplication

class `SNS.PlatformApplication` (*arn*)

A resource representing an Amazon Simple Notification Service (SNS) PlatformApplication:

```
import boto3
```

```
sns = boto3.resource('sns')
platform_application = sns.PlatformApplication('arn')
```

Parameters **arn** (*string*) – The PlatformApplication’s arn identifier. This **must** be set.

These are the resource’s available identifiers:

- `arn`

These are the resource’s available attributes:

- `attributes`

These are the resource's available actions:

- `create_platform_endpoint()`
- `delete()`
- `load()`
- `reload()`
- `set_attributes()`

These are the resource's available collections:

- `endpoints`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

arn

(*string*) The PlatformApplication's arn identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

attributes

(*dict*)

Attributes include the following:

- `EventEndpointCreated` – Topic ARN to which EndpointCreated event notifications should be sent.
- `EventEndpointDeleted` – Topic ARN to which EndpointDeleted event notifications should be sent.
- `EventEndpointUpdated` – Topic ARN to which EndpointUpdate event notifications should be sent.
- `EventDeliveryFailure` – Topic ARN to which DeliveryFailure event notifications should be sent upon Direct Publish delivery failure (permanent) to one of the application's endpoints.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

create_platform_endpoint (***kwargs*)

Creates an endpoint for a device and mobile app on one of the supported push notification services, such as GCM and APNS. `CreatePlatformEndpoint` requires the `PlatformApplicationArn` that is returned from `CreatePlatformApplication`. The `EndpointArn` that is returned when using `CreatePlatformEndpoint` can then be used by the `Publish` action to send a message to a mobile app or by the `Subscribe` action for subscription to a topic. The `CreatePlatformEndpoint` action is idempotent, so if the requester already owns an endpoint with the same device token and attributes, that endpoint's ARN is returned without creating a new endpoint. For more information, see [Using Amazon SNS Mobile Push Notifications](#).

When using `CreatePlatformEndpoint` with Baidu, two attributes must be provided: `ChannelId` and `UserId`. The token field must also contain the `ChannelId`. For more information, see [Creating an Amazon SNS Endpoint for Baidu](#).

Request Syntax

```
platform_endpoint = platform_application.create_platform_endpoint(  
    Token='string',  
    CustomUserData='string',  
    Attributes={
```

```

        'string': 'string'
    }
)

```

Parameters

- **Token** (*string*) – [REQUIRED]

Unique identifier created by the notification service for an app on a device. The specific name for Token will vary, depending on which notification service is being used. For example, when using APNS as the notification service, you need the device token. Alternatively, when using GCM or ADM, the device token equivalent is called the registration ID.

- **CustomUserData** (*string*) – Arbitrary user data to associate with the endpoint. Amazon SNS does not use this data. The data must be in UTF-8 format and less than 2KB.
- **Attributes** (*dict*) – For a list of attributes, see [SetEndpointAttributes](#) .
 - (*string*) –
 - * (*string*) –

Return type `sns.PlatformEndpoint`

Returns PlatformEndpoint resource

delete()

Deletes a platform application object for one of the supported push notification services, such as APNS and GCM. For more information, see [Using Amazon SNS Mobile Push Notifications](#) .

Request Syntax

```
response = platform_application.delete()
```

Returns None

load()

Calls `sns.Client.get_platform_application_attributes()` to update the attributes of the PlatformApplication resource

Request Syntax

```
platform_application.load()
```

Returns None

reload()

Calls `sns.Client.get_platform_application_attributes()` to update the attributes of the PlatformApplication resource

Request Syntax

```
platform_application.reload()
```

Returns None

set_attributes(kwargs)**

Sets the attributes of the platform application object for the supported push notification services, such as APNS and GCM. For more information, see [Using Amazon SNS Mobile Push Notifications](#) .

Request Syntax

```
response = platform_application.set_attributes(  
    Attributes={  
        'string': 'string'  
    }  
)
```

Parameters *Attributes* (*dict*) – [REQUIRED]

A map of the platform application attributes. Attributes in this map include the following:

- *PlatformCredential* – The credential received from the notification service. For APNS/APNS_SANDBOX, *PlatformCredential* is “private key”. For GCM, *PlatformCredential* is “API key”. For ADM, *PlatformCredential* is “client secret”.
- *PlatformPrincipal* – The principal received from the notification service. For APNS/APNS_SANDBOX, *PlatformPrincipal* is “SSL certificate”. For GCM, *PlatformPrincipal* is not applicable. For ADM, *PlatformPrincipal* is “client id”.
- *EventEndpointCreated* – Topic ARN to which *EndpointCreated* event notifications should be sent.
- *EventEndpointDeleted* – Topic ARN to which *EndpointDeleted* event notifications should be sent.
- *EventEndpointUpdated* – Topic ARN to which *EndpointUpdate* event notifications should be sent.
- *EventDeliveryFailure* – Topic ARN to which *DeliveryFailure* event notifications should be sent upon Direct Publish delivery failure (permanent) to one of the application’s endpoints.
- (*string*) –
 – (*string*) –

Returns None

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

endpoints**all()**

Creates an iterable of all *PlatformEndpoint* resources in the collection.

Request Syntax

```
platform_endpoint_iterator = platform_application.endpoints.all()
```

Return type `list(sns.PlatformEndpoint)`

Returns A list of *PlatformEndpoint* resources

filter(kwargs)**

Creates an iterable of all *PlatformEndpoint* resources in the collection filtered by kwargs passed to method.

Request Syntax

```
platform_endpoint_iterator = platform_application.endpoints.filter(  
    NextToken='string'  
)
```


Parameters **NextToken** (*string*) – NextToken string is used when calling ListEndpointsByPlatformApplication action to retrieve additional records that are available after the first page results.

Return type list(sns.PlatformEndpoint)

Returns A list of PlatformEndpoint resources

limit (***kwargs*)

Creates an iterable up to a specified amount of PlatformEndpoint resources in the collection.

Request Syntax

```
platform_endpoint_iterator = platform_application.endpoints.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type list(sns.PlatformEndpoint)

Returns A list of PlatformEndpoint resources

page_size (***kwargs*)

Creates an iterable of all PlatformEndpoint resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
platform_endpoint_iterator = platform_application.endpoints.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type list(sns.PlatformEndpoint)

Returns A list of PlatformEndpoint resources

PlatformEndpoint

class SNS.**PlatformEndpoint** (*arn*)

A resource representing an Amazon Simple Notification Service (SNS) PlatformEndpoint:

```
import boto3
```

```
sns = boto3.resource('sns')
platform_endpoint = sns.PlatformEndpoint('arn')
```

Parameters **arn** (*string*) – The PlatformEndpoint’s arn identifier. This **must** be set.

These are the resource’s available identifiers:

- `arn`

These are the resource’s available attributes:

- `attributes`

These are the resource’s available actions:

- `delete()`
- `load()`
- `publish()`
- `reload()`
- `set_attributes()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

arn

(*string*) The PlatformEndpoint's arn identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

attributes

(*dict*)

Attributes include the following:

- `CustomUserData` – arbitrary user data to associate with the endpoint. Amazon SNS does not use this data. The data must be in UTF-8 format and less than 2KB.
- `Enabled` – flag that enables/disables delivery to the endpoint. Amazon SNS will set this to false when a notification service indicates to Amazon SNS that the endpoint is invalid. Users can set it back to true, typically after updating Token.
- `Token` – device token, also referred to as a registration id, for an app and mobile device. This is returned from the notification service when an app and mobile device are registered with the notification service.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

delete()

Deletes the endpoint from Amazon SNS. This action is idempotent. For more information, see [Using Amazon SNS Mobile Push Notifications](#).

Request Syntax

```
response = platform_endpoint.delete()
```

Returns None

load()

Calls `sns.Client.get_endpoint_attributes()` to update the attributes of the PlatformEndpoint resource

Request Syntax

```
platform_endpoint.load()
```

Returns None

publish(kwargs)**

Sends a message to all of a topic's subscribed endpoints. When a `messageId` is returned, the message has been saved and Amazon SNS will attempt to deliver it to the topic's subscribers shortly. The format of the outgoing message to each subscribed endpoint depends on the notification protocol selected.

To use the `Publish` action for sending a message to a mobile endpoint, such as an app on a Kindle device or mobile phone, you must specify the `EndpointArn`. The `EndpointArn` is returned when making a call with the `CreatePlatformEndpoint` action. The second example below shows a request and response for publishing to a mobile endpoint.

Request Syntax

```

response = platform_endpoint.publish(
    TopicArn='string',
    Message='string',
    Subject='string',
    MessageStructure='string',
    MessageAttributes={
        'string': {
            'DataType': 'string',
            'StringValue': 'string',
            'BinaryValue': b'bytes'
        }
    }
)

```

Parameters

- **TopicArn** (*string*) – The topic you want to publish to.
- **Message** (*string*) – [REQUIRED]

The message you want to send to the topic.

If you want to send the same message to all transport protocols, include the text of the message as a String value.

If you want to send different messages for each transport protocol, set the value of the `MessageStructure` parameter to `json` and use a JSON object for the `Message` parameter. See the Examples section for the format of the JSON object.

Constraints: Messages must be UTF-8 encoded strings at most 256 KB in size (262144 bytes, not 262144 characters).

JSON-specific constraints:

- Keys in the JSON object that correspond to supported transport protocols must have simple JSON string values.
- The values will be parsed (unescaped) before they are used in outgoing messages.
- Outbound notifications are JSON encoded (meaning that the characters will be reescaped for sending).
- Values have a minimum length of 0 (the empty string, “”, is allowed).
- Values have a maximum length bounded by the overall message size (so, including multiple protocols may limit message sizes).
- Non-string values will cause the key to be ignored.
- Keys that do not correspond to supported transport protocols are ignored.
- Duplicate keys are not allowed.
- Failure to parse or validate any key or value in the message will cause the `Publish` call to return an error (no partial delivery).
- **Subject** (*string*) – Optional parameter to be used as the “Subject” line when the message is delivered to email endpoints. This field will also be included, if present, in the standard JSON messages delivered to other endpoints.

Constraints: Subjects must be ASCII text that begins with a letter, number, or punctuation mark; must not include line breaks or control characters; and must be less than 100 characters long.

- **MessageStructure** (*string*) – Set `MessageStructure` to `json` if you want to send a different message for each protocol. For example, using one `publish` action, you can send a short message to your SMS subscribers and a longer message to your email subscribers. If you set `MessageStructure` to `json`,

the value of the `Message` parameter must:

- be a syntactically valid JSON object; and
- contain at least a top-level JSON key of “default” with a value that is a string.

You can define other top-level keys that define the message you want to send to a specific transport protocol (e.g., “http”).

For information about sending different messages for each protocol using the AWS Management Console, go to [Create Different Messages for Each Protocol](#) in the *Amazon Simple Notification Service Getting Started Guide* .

Valid value: `json`

- **MessageAttributes** (*dict*) – Message attributes for Publish action.

- (*string*) –
- * (*dict*) –

The user-specified message attribute value. For string data types, the value attribute has the same restrictions on the content as the message body. For more information, see [Publish](#) .

Name, type, and value must not be empty or null. In addition, the message body should not be empty or null. All parts of the message attribute, including name, type, and value, are included in the message size restriction, which is currently 256 KB (262,144 bytes). For more information, see [Using Amazon SNS Message Attributes](#) .

- **DataType** (*string*) – [REQUIRED]

Amazon SNS supports the following logical data types: String, Number, and Binary. For more information, see [Message Attribute Data Types](#) .

- **StringValue** (*string*) –

Strings are Unicode with UTF8 binary encoding. For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters .

- **BinaryValue** (*bytes*) –

Binary type attributes can store any binary data, for example, compressed data, encrypted data, or images.

Return type `dict`

Returns

Response Syntax

```
{
    'MessageId': 'string'
}
```

Response Structure

- (*dict*) –

Response for Publish action.

- **MessageId** (*string*) –

Unique identifier assigned to the published message.

Length Constraint: Maximum 100 characters

reload()

Calls `sns.Client.get_endpoint_attributes()` to update the attributes of the PlatformEndpoint resource

Request Syntax

```
platform_endpoint.reload()
```

Returns None

set_attributes(kwargs)**

Sets the attributes for an endpoint for a device on one of the supported push notification services, such as GCM and APNS. For more information, see [Using Amazon SNS Mobile Push Notifications](#).

Request Syntax

```
response = platform_endpoint.set_attributes(
    Attributes={
        'string': 'string'
    }
)
```

Parameters Attributes (dict) – [REQUIRED]

A map of the endpoint attributes. Attributes in this map include the following:

- `CustomUserData` – arbitrary user data to associate with the endpoint. Amazon SNS does not use this data. The data must be in UTF-8 format and less than 2KB.
- `Enabled` – flag that enables/disables delivery to the endpoint. Amazon SNS will set this to false when a notification service indicates to Amazon SNS that the endpoint is invalid. Users can set it back to true, typically after updating Token.
- `Token` – device token, also referred to as a registration id, for an app and mobile device. This is returned from the notification service when an app and mobile device are registered with the notification service.
- *(string)* –
– *(string)* –

Returns None

Subscription**class SNS.Subscription(arn)**

A resource representing an Amazon Simple Notification Service (SNS) Subscription:

```
import boto3
```

```
sns = boto3.resource('sns')
subscription = sns.Subscription('arn')
```

Parameters arn (string) – The Subscription’s arn identifier. This **must** be set.

These are the resource’s available identifiers:

- `arn`

These are the resource’s available attributes:

- `attributes`

These are the resource’s available actions:

- `delete()`
- `load()`
- `reload()`
- `set_attributes()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

arn

(*string*) The Subscription's arn identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

attributes

(*dict*)

A map of the subscription's attributes. Attributes in this map include the following:

- `SubscriptionArn` – the subscription's ARN
- `TopicArn` – the topic ARN that the subscription is associated with
- `Owner` – the AWS account ID of the subscription's owner
- `ConfirmationWasAuthenticated` – true if the subscription confirmation request was authenticated
- `DeliveryPolicy` – the JSON serialization of the subscription's delivery policy
- `EffectiveDeliveryPolicy` – the JSON serialization of the effective delivery policy that takes into account the topic delivery policy and account system defaults

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

delete()

Deletes a subscription. If the subscription requires authentication for deletion, only the owner of the subscription or the topic's owner can unsubscribe, and an AWS signature is required. If the `Unsubscribe` call does not require authentication and the requester is not the subscription owner, a final cancellation message is delivered to the endpoint, so that the endpoint owner can easily resubscribe to the topic if the `Unsubscribe` request was unintended.

Request Syntax

```
response = subscription.delete()
```

Returns None

load()

Calls `sns.Client.get_subscription_attributes()` to update the attributes of the Subscription resource

Request Syntax

```
subscription.load()
```

Returns None

reload()

Calls `sns.Client.get_subscription_attributes()` to update the attributes of the Subscription resource

Request Syntax

```
subscription.reload()
```

Returns None

set_attributes (***kwargs*)

Allows a subscription owner to set an attribute of the topic to a new value.

Request Syntax

```
response = subscription.set_attributes(
    AttributeName='string',
    AttributeValue='string'
)
```

Parameters

- **AttributeName** (*string*) – [REQUIRED]

The name of the attribute you want to set. Only a subset of the subscriptions attributes are mutable.

Valid values: `DeliveryPolicy` | `RawMessageDelivery`

- **AttributeValue** (*string*) – The new value for the attribute in JSON format.

Returns None**Topic**

class `SNS.Topic` (*arn*)

A resource representing an Amazon Simple Notification Service (SNS) Topic:

```
import boto3
```

```
sns = boto3.resource('sns')
topic = sns.Topic('arn')
```

Parameters **arn** (*string*) – The Topic’s arn identifier. This **must** be set.

These are the resource’s available identifiers:

- `arn`

These are the resource’s available attributes:

- `attributes`

These are the resource’s available actions:

- `add_permission()`
- `confirm_subscription()`
- `delete()`
- `load()`
- `publish()`
- `reload()`
- `remove_permission()`
- `set_attributes()`
- `subscribe()`

These are the resource’s available collections:

- `subscriptions`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

arn

(*string*) The Topic's arn identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

attributes

(*dict*)

A map of the topic's attributes. Attributes in this map include the following:

- **TopicArn** – the topic's ARN
- **Owner** – the AWS account ID of the topic's owner
- **Policy** – the JSON serialization of the topic's access control policy
- **DisplayName** – the human-readable name used in the "From" field for notifications to email and email-json endpoints
- **SubscriptionsPending** – the number of subscriptions pending confirmation on this topic
- **SubscriptionsConfirmed** – the number of confirmed subscriptions on this topic
- **SubscriptionsDeleted** – the number of deleted subscriptions on this topic
- **DeliveryPolicy** – the JSON serialization of the topic's delivery policy
- **EffectiveDeliveryPolicy** – the JSON serialization of the effective delivery policy that takes into account system defaults

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

add_permission (***kwargs*)

Adds a statement to a topic's access control policy, granting access for the specified AWS accounts to the specified actions.

Request Syntax

```
response = topic.add_permission(  
    Label='string',  
    AWSAccountId=[  
        'string',  
    ],  
    ActionName=[  
        'string',  
    ]  
)
```

Parameters

- **Label** (*string*) – **[REQUIRED]**

A unique identifier for the new policy statement.

- **AWSAccountId** (*list*) – **[REQUIRED]**

The AWS account IDs of the users (principals) who will be given access to the specified actions. The users must have AWS accounts, but do not need to be signed up for this service.

– (*string*) –

- **ActionName** (*list*) – [REQUIRED]

The action you want to allow for the specified principal(s).

Valid values: any Amazon SNS action name.

– (*string*) –

Returns None

confirm_subscription (***kwargs*)

Verifies an endpoint owner’s intent to receive messages by validating the token sent to the endpoint by an earlier `Subscribe` action. If the token is valid, the action creates a new subscription and returns its Amazon Resource Name (ARN). This call requires an AWS signature only when the `AuthenticateOnUnsubscribe` flag is set to “true”.

Request Syntax

```
subscription = topic.confirm_subscription(
    Token='string',
    AuthenticateOnUnsubscribe='string'
)
```

Parameters

- **Token** (*string*) – [REQUIRED]

Short-lived token sent to an endpoint during the `Subscribe` action.

- **AuthenticateOnUnsubscribe** (*string*) – Disallows unauthenticated unsubscribes of the subscription. If the value of this parameter is `true` and the request has an AWS signature, then only the topic owner and the subscription owner can unsubscribe the endpoint. The unsubscribe action requires AWS authentication.

Return type `sns.Subscription`

Returns Subscription resource

delete ()

Deletes a topic and all its subscriptions. Deleting a topic might prevent some messages previously sent to the topic from being delivered to subscribers. This action is idempotent, so deleting a topic that does not exist does not result in an error.

Request Syntax

```
response = topic.delete()
```

Returns None

load ()

Calls `sns.Client.get_topic_attributes()` to update the attributes of the Topic resource

Request Syntax

```
topic.load()
```

Returns None

publish (***kwargs*)

Sends a message to all of a topic’s subscribed endpoints. When a `messageId` is returned, the message has been saved and Amazon SNS will attempt to deliver it to the topic’s subscribers shortly. The format of the outgoing message to each subscribed endpoint depends on the notification protocol selected.

To use the `Publish` action for sending a message to a mobile endpoint, such as an app on a Kindle device or mobile phone, you must specify the `EndpointArn`. The `EndpointArn` is returned when making

a call with the `CreatePlatformEndpoint` action. The second example below shows a request and response for publishing to a mobile endpoint.

Request Syntax

```
response = topic.publish(  
    TargetArn='string',  
    Message='string',  
    Subject='string',  
    MessageStructure='string',  
    MessageAttributes={  
        'string': {  
            'DataType': 'string',  
            'StringValue': 'string',  
            'BinaryValue': b'bytes'  
        }  
    }  
)
```

Parameters

- **TargetArn** (*string*) – Either TopicArn or EndpointArn, but not both.
- **Message** (*string*) – **[REQUIRED]**

The message you want to send to the topic.

If you want to send the same message to all transport protocols, include the text of the message as a String value.

If you want to send different messages for each transport protocol, set the value of the `MessageStructure` parameter to `json` and use a JSON object for the `Message` parameter. See the Examples section for the format of the JSON object.

Constraints: Messages must be UTF-8 encoded strings at most 256 KB in size (262144 bytes, not 262144 characters).

JSON-specific constraints:

- Keys in the JSON object that correspond to supported transport protocols must have simple JSON string values.
 - The values will be parsed (unescaped) before they are used in outgoing messages.
 - Outbound notifications are JSON encoded (meaning that the characters will be reescaped for sending).
 - Values have a minimum length of 0 (the empty string, “”, is allowed).
 - Values have a maximum length bounded by the overall message size (so, including multiple protocols may limit message sizes).
 - Non-string values will cause the key to be ignored.
 - Keys that do not correspond to supported transport protocols are ignored.
 - Duplicate keys are not allowed.
 - Failure to parse or validate any key or value in the message will cause the `Publish` call to return an error (no partial delivery).
- **Subject** (*string*) – Optional parameter to be used as the “Subject” line when the message is delivered to email endpoints. This field will also be included, if present, in the standard JSON messages delivered to other endpoints.

Constraints: Subjects must be ASCII text that begins with a letter, number, or punctuation mark; must not include line breaks or control characters; and must be less than 100 characters long.

- **MessageStructure** (*string*) – Set `MessageStructure` to `json` if you want to send a different message for each protocol. For example, using one publish action, you can send a short message to your SMS subscribers and a longer message to your email subscribers. If you set `MessageStructure` to `json`, the value of the `Message` parameter must:
 - be a syntactically valid JSON object; and
 - contain at least a top-level JSON key of “default” with a value that is a string.

You can define other top-level keys that define the message you want to send to a specific transport protocol (e.g., “http”).

For information about sending different messages for each protocol using the AWS Management Console, go to [Create Different Messages for Each Protocol](#) in the *Amazon Simple Notification Service Getting Started Guide*.

Valid value: `json`

- **MessageAttributes** (*dict*) – Message attributes for Publish action.
 - (*string*) –
 - * (*dict*) –

The user-specified message attribute value. For string data types, the value attribute has the same restrictions on the content as the message body. For more information, see [Publish](#).

Name, type, and value must not be empty or null. In addition, the message body should not be empty or null. All parts of the message attribute, including name, type, and value, are included in the message size restriction, which is currently 256 KB (262,144 bytes). For more information, see [Using Amazon SNS Message Attributes](#).

- **DataType** (*string*) – [REQUIRED]

Amazon SNS supports the following logical data types: String, Number, and Binary. For more information, see [Message Attribute Data Types](#).

- **StringValue** (*string*) –

Strings are Unicode with UTF8 binary encoding. For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters.

- **BinaryValue** (*bytes*) –

Binary type attributes can store any binary data, for example, compressed data, encrypted data, or images.

Return type `dict`
Returns

Response Syntax

```
{
    'MessageId': 'string'
}
```

Response Structure

- (*dict*) –
 Response for Publish action.

– **MessageId** (*string*) –

Unique identifier assigned to the published message.

Length Constraint: Maximum 100 characters

reload()

Calls `sns.Client.get_topic_attributes()` to update the attributes of the Topic resource

Request Syntax

```
topic.reload()
```

Returns None

remove_permission (***kwargs*)

Removes a statement from a topic's access control policy.

Request Syntax

```
response = topic.remove_permission(  
    Label='string'  
)
```

Parameters **Label** (*string*) – **[REQUIRED]**

The unique label of the statement you want to remove.

Returns None

set_attributes (***kwargs*)

Allows a topic owner to set an attribute of the topic to a new value.

Request Syntax

```
response = topic.set_attributes(  
    AttributeName='string',  
    AttributeValue='string'  
)
```

Parameters

- **AttributeName** (*string*) – **[REQUIRED]**

The name of the attribute you want to set. Only a subset of the topic's attributes are mutable.

Valid values: `Policy` | `DisplayName` | `DeliveryPolicy`

- **AttributeValue** (*string*) – The new value for the attribute.

Returns None

subscribe (***kwargs*)

Prepares to subscribe an endpoint by sending the endpoint a confirmation message. To actually create a subscription, the endpoint owner must call the `ConfirmSubscription` action with the token from the confirmation message. Confirmation tokens are valid for three days.

Request Syntax

```
subscription = topic.subscribe(  
    Protocol='string',  
    Endpoint='string'  
)
```

Parameters

- **Protocol** (*string*) – [REQUIRED]

The protocol you want to use. Supported protocols include:

- `http` – delivery of JSON-encoded message via HTTP POST
- `https` – delivery of JSON-encoded message via HTTPS POST
- `email` – delivery of message via SMTP
- `email-json` – delivery of JSON-encoded message via SMTP
- `sms` – delivery of message via SMS
- `sqs` – delivery of JSON-encoded message to an Amazon SQS queue
- `application` – delivery of JSON-encoded message to an EndpointArn for a mobile app and device.

- **Endpoint** (*string*) – The endpoint that you want to receive notifications. Endpoints vary by protocol:

- For the `http` protocol, the endpoint is an URL beginning with “`http://`”
- For the `https` protocol, the endpoint is a URL beginning with “`https://`”
- For the `email` protocol, the endpoint is an email address
- For the `email-json` protocol, the endpoint is an email address
- For the `sms` protocol, the endpoint is a phone number of an SMS-enabled device
- For the `sqs` protocol, the endpoint is the ARN of an Amazon SQS queue
- For the `application` protocol, the endpoint is the EndpointArn of a mobile app and device.

Return type `sns.Subscription`

Returns Subscription resource

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

subscriptions

`all()`

Creates an iterable of all Subscription resources in the collection.

Request Syntax

```
subscription_iterator = topic.subscriptions.all()
```

Return type `list(sns.Subscription)`

Returns A list of Subscription resources

`filter(**kwargs)`

Creates an iterable of all Subscription resources in the collection filtered by kwargs passed to method.

Request Syntax

```
subscription_iterator = topic.subscriptions.filter(
    NextToken='string'
)
```

Parameters `NextToken` (*string*) – Token returned by the previous `ListSubscriptionsByTopic` request.

Return type `list(sns.Subscription)`

Returns A list of Subscription resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Subscription resources in the collection.

Request Syntax

```
subscription_iterator = topic.subscriptions.limit(  
    count=123  
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(sns.Subscription)`

Returns A list of Subscription resources

page_size (***kwargs*)

Creates an iterable of all Subscription resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
subscription_iterator = topic.subscriptions.page_size(  
    count=123  
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(sns.Subscription)`

Returns A list of Subscription resources

SQS

Table of Contents

- [SQS](#)
 - [Client](#)
 - [Service Resource](#)
 - [Message](#)
 - [Queue](#)

Client

class `SQS.Client`

A low-level client representing Amazon Simple Queue Service (SQS):

```
import boto3  
  
client = boto3.client('sqs')
```

These are the available methods:

- `add_permission()`
- `can_paginate()`
- `change_message_visibility()`
- `change_message_visibility_batch()`
- `create_queue()`
- `delete_message()`

- `delete_message_batch()`
- `delete_queue()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_queue_attributes()`
- `get_queue_url()`
- `get_waiter()`
- `list_dead_letter_source_queues()`
- `list_queues()`
- `purge_queue()`
- `receive_message()`
- `remove_permission()`
- `send_message()`
- `send_message_batch()`
- `set_queue_attributes()`

add_permission (***kwargs*)

Adds a permission to a queue for a specific [principal](#) . This allows for sharing access to the queue.

When you create a queue, you have full control access rights for the queue. Only you (as owner of the queue) can grant or deny permissions to the queue. For more information about these permissions, see [Shared Queues in the Amazon SQS Developer Guide](#) .

Note: `AddPermission` writes an Amazon SQS-generated policy. If you want to write your own policy, use `SetQueueAttributes` to upload your policy. For more information about writing your own policy, see [Using The Access Policy Language in the Amazon SQS Developer Guide](#) .

Note: Some API actions take lists of parameters. These lists are specified using the `param.n` notation. Values of `n` are integers starting from 1. For example, a parameter list with two elements looks like this:

```
Attribute.1=this
```

```
Attribute.2=that
```

Request Syntax

```
response = client.add_permission(
    QueueUrl='string',
    Label='string',
    AWSAccountIds=[
        'string',
    ],
    Actions=[
        'string',
    ]
)
```

Parameters

- **QueueUrl** (*string*) – [REQUIRED]

The URL of the Amazon SQS queue to take action on.

- **Label** (*string*) – [REQUIRED]

The unique identification of the permission you're setting (e.g., `AliceSendMessage`). Constraints: Maximum 80 characters; alphanumeric characters, hyphens (-), and underscores (_) are allowed.

- **AWSAccountIds** (*list*) – [REQUIRED]

The AWS account number of the [principal](#) who will be given permission. The principal must have an AWS account, but does not need to be signed up for Amazon SQS. For information about locating the AWS account identification, see [Your AWS Identifiers](#) in the *Amazon SQS Developer Guide*.

– (string) –

- **Actions** (list) – **[REQUIRED]**

The action the client wants to allow for the specified principal. The following are valid values: * | SendMessage | ReceiveMessage | DeleteMessage | ChangeMessageVisibility | GetQueueAttributes | GetQueueUrl . For more information about these actions, see [Understanding Permissions](#) in the *Amazon SQS Developer Guide*.

Specifying `SendMessage`, `DeleteMessage`, or `ChangeMessageVisibility` for the `ActionName.n` also grants permissions for the corresponding batch versions of those actions: `SendMessageBatch`, `DeleteMessageBatch`, and `ChangeMessageVisibilityBatch`.

– (string) –

Returns None

can_paginate (operation_name)

Check if an operation can be paginated.

Parameters `operation_name` (string) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

change_message_visibility (**kwargs)

Changes the visibility timeout of a specified message in a queue to a new value. The maximum allowed timeout value you can set the value to is 12 hours. This means you can't extend the timeout of a message in an existing queue to more than a total visibility timeout of 12 hours. (For more information visibility timeout, see [Visibility Timeout](#) in the *Amazon SQS Developer Guide*.)

For example, let's say you have a message and its default message visibility timeout is 30 minutes. You could call `ChangeMessageVisibility` with a value of two hours and the effective timeout would be two hours and 30 minutes. When that time comes near you could again extend the time out by calling `ChangeMessageVisibility`, but this time the maximum allowed timeout would be 9 hours and 30 minutes.

Note: There is a 120,000 limit for the number of inflight messages per queue. Messages are inflight after they have been received from the queue by a consuming component, but have not yet been deleted from the queue. If you reach the 120,000 limit, you will receive an `OverLimit` error message from Amazon SQS. To help avoid reaching the limit, you should delete the messages from the queue after they have been processed. You can also increase the number of queues you use to process the messages.

Warning: If you attempt to set the `VisibilityTimeout` to an amount more than the maximum time left, Amazon SQS returns an error. It will not automatically recalculate and increase the timeout to the maximum time remaining.

Warning: Unlike with a queue, when you change the visibility timeout for a specific message, that timeout value is applied immediately but is not saved in memory for that message. If you don't delete a message after it is received, the visibility timeout for the message the next time it is received reverts to the original timeout value, not the value you set with the `ChangeMessageVisibility` action.

Request Syntax

```
response = client.change_message_visibility(
    QueueUrl='string',
    ReceiptHandle='string',
    VisibilityTimeout=123
)
```

Parameters

- **QueueUrl** (*string*) – [REQUIRED]
The URL of the Amazon SQS queue to take action on.
- **ReceiptHandle** (*string*) – [REQUIRED]
The receipt handle associated with the message whose visibility timeout should be changed. This parameter is returned by the `ReceiveMessage` action.
- **VisibilityTimeout** (*integer*) – [REQUIRED]
The new value (in seconds - from 0 to 43200 - maximum 12 hours) for the message's visibility timeout.

Returns None

`change_message_visibility_batch` (***kwargs*)

Changes the visibility timeout of multiple messages. This is a batch version of `ChangeMessageVisibility`. The result of the action on each message is reported individually in the response. You can send up to 10 `ChangeMessageVisibility` requests with each `ChangeMessageVisibilityBatch` action.

Warning: Because the batch request can result in a combination of successful and unsuccessful actions, you should check for batch errors even when the call returns an HTTP status code of 200.

Note: Some API actions take lists of parameters. These lists are specified using the `param.n` notation. Values of `n` are integers starting from 1. For example, a parameter list with two elements looks like this:

```
Attribute.1=this
```

```
Attribute.2=that
```

Request Syntax

```
response = client.change_message_visibility_batch(
    QueueUrl='string',
    Entries=[
        {
            'Id': 'string',
            'ReceiptHandle': 'string',
            'VisibilityTimeout': 123
        },
    ]
)
```

Parameters

- **QueueUrl** (*string*) – [REQUIRED]

The URL of the Amazon SQS queue to take action on.

- **Entries** (*list*) – [REQUIRED]

A list of receipt handles of the messages for which the visibility timeout must be changed.

– (*dict*) –

Encloses a receipt handle and an entry id for each message in `ChangeMessageVisibilityBatch`.

Warning: All of the following parameters are list parameters that must be prefixed with `ChangeMessageVisibilityBatchRequestEntry.n`, where `n` is an integer value starting with 1. For example, a parameter list for this action might look like this:

`ChangeMessageVisibilityBatchRequestEntry.1.Id=change_visibility_m`

`ChangeMessageVisibilityBatchRequestEntry.1.ReceiptHandle=Your_Rece`

`ChangeMessageVisibilityBatchRequestEntry.1.VisibilityTimeout=45`

- * **Id** (*string*) – [REQUIRED]

An identifier for this particular receipt handle. This is used to communicate the result. Note that the `Id`s of a batch request need to be unique within the request.

- * **ReceiptHandle** (*string*) – [REQUIRED]

A receipt handle.

- * **VisibilityTimeout** (*integer*) –

The new value (in seconds) for the message's visibility timeout.

Return type dict

Returns

Response Syntax

```
{
  'Successful': [
    {
      'Id': 'string'
    },
  ],
  'Failed': [
    {
      'Id': 'string',
      'SenderFault': True|False,
      'Code': 'string',
      'Message': 'string'
    },
  ]
}
```

Response Structure

- (*dict*) –

For each message in the batch, the response contains a `ChangeMessageVisibilityBatchResultEntry` tag if the message succeeds or a `BatchResultErrorEntry` tag if the message fails.

– **Successful** (*list*) –

A list of `ChangeMessageVisibilityBatchResultEntry` items.

* (*dict*) –

Encloses the id of an entry in `ChangeMessageVisibilityBatch`.

· **Id** (*string*) –

Represents a message whose visibility timeout has been changed successfully.

– **Failed** (*list*) –

A list of `BatchResultErrorEntry` items.

* (*dict*) –

This is used in the responses of batch API to give a detailed description of the result of an action on each entry in the request.

· **Id** (*string*) –

The id of an entry in a batch request.

· **SenderFault** (*boolean*) –

Whether the error happened due to the sender's fault.

· **Code** (*string*) –

An error code representing why the action failed on this entry.

· **Message** (*string*) –

A message explaining why the action failed on this entry.

create_queue (***kwargs*)

Creates a new queue, or returns the URL of an existing one. When you request `CreateQueue`, you provide a name for the queue. To successfully create a new queue, you must provide a name that is unique within the scope of your own queues.

Note: If you delete a queue, you must wait at least 60 seconds before creating a queue with the same name.

You may pass one or more attributes in the request. If you do not provide a value for any attribute, the queue will have the default value for that attribute. Permitted attributes are the same that can be set using `SetQueueAttributes`.

Note: Use `GetQueueUrl` to get a queue's URL. `GetQueueUrl` requires only the `QueueName` parameter.

If you provide the name of an existing queue, along with the exact names and values of all the queue's attributes, `CreateQueue` returns the queue URL for the existing queue. If the queue name, attribute names, or attribute values do not match an existing queue, `CreateQueue` returns an error.

Note: Some API actions take lists of parameters. These lists are specified using the `param.n` notation. Values of `n` are integers starting from 1. For example, a parameter list with two elements looks like this:

`Attribute.1=this`

`Attribute.2=that`

Request Syntax

```
response = client.create_queue(  
    QueueName='string',  
    Attributes={  
        'string': 'string'  
    }  
)
```

Parameters

- **QueueName** (*string*) – **[REQUIRED]**

The name for the queue to be created.

- **Attributes** (*dict*) – A map of attributes with their corresponding values.

The following lists the names, descriptions, and values of the special request parameters the CreateQueue action uses:

- **DelaySeconds** - The time in seconds that the delivery of all messages in the queue will be delayed. An integer from 0 to 900 (15 minutes). The default for this attribute is 0 (zero).
- **MaximumMessageSize** - The limit of how many bytes a message can contain before Amazon SQS rejects it. An integer from 1024 bytes (1 KiB) up to 262144 bytes (256 KiB). The default for this attribute is 262144 (256 KiB).
- **MessageRetentionPeriod** - The number of seconds Amazon SQS retains a message. Integer representing seconds, from 60 (1 minute) to 1209600 (14 days). The default for this attribute is 345600 (4 days).
- **Policy** - The queue's policy. A valid AWS policy. For more information about policy structure, see [Overview of AWS IAM Policies](#) in the *Amazon IAM User Guide* .
- **ReceiveMessageWaitTimeSeconds** - The time for which a `ReceiveMessage` call will wait for a message to arrive. An integer from 0 to 20 (seconds). The default for this attribute is 0.
- **VisibilityTimeout** - The visibility timeout for the queue. An integer from 0 to 43200 (12 hours). The default for this attribute is 30. For more information about visibility timeout, see [Visibility Timeout](#) in the *Amazon SQS Developer Guide* .

- (*string*) –

The name of a queue attribute.

- * (*string*) –

The value of a queue attribute.

Return type dict

Returns

Response Syntax

```
{  
    'QueueUrl': 'string'  
}
```

Response Structure

- (*dict*) –

Returns the QueueUrl element of the created queue.

- **QueueUrl** (*string*) –

The URL for the created Amazon SQS queue.

delete_message (***kwargs*)

Deletes the specified message from the specified queue. You specify the message by using the message's receipt handle and not the message ID you received when you sent the message. Even if the message is locked by another reader due to the visibility timeout setting, it is still deleted from the queue. If you leave a message in the queue for longer than the queue's configured retention period, Amazon SQS automatically deletes it.

Note: The receipt handle is associated with a specific instance of receiving the message. If you receive a message more than once, the receipt handle you get each time you receive the message is different. When you request `DeleteMessage`, if you don't provide the most recently received receipt handle for the message, the request will still succeed, but the message might not be deleted.

Warning: It is possible you will receive a message even after you have deleted it. This might happen on rare occasions if one of the servers storing a copy of the message is unavailable when you request to delete the message. The copy remains on the server and might be returned to you again on a subsequent receive request. You should create your system to be idempotent so that receiving a particular message more than once is not a problem.

Request Syntax

```
response = client.delete_message(
    QueueUrl='string',
    ReceiptHandle='string'
)
```

Parameters

- **QueueUrl** (*string*) – [REQUIRED]
The URL of the Amazon SQS queue to take action on.
- **ReceiptHandle** (*string*) – [REQUIRED]
The receipt handle associated with the message to delete.

Returns None

delete_message_batch (***kwargs*)

Deletes up to ten messages from the specified queue. This is a batch version of `DeleteMessage`. The result of the delete action on each message is reported individually in the response.

Warning: Because the batch request can result in a combination of successful and unsuccessful actions, you should check for batch errors even when the call returns an HTTP status code of 200.

Note: Some API actions take lists of parameters. These lists are specified using the `param.n` notation. Values of `n` are integers starting from 1. For example, a parameter list with two elements looks like this:

```
Attribute.1=this
```

```
Attribute.2=that
```

Request Syntax

```
response = client.delete_message_batch(
    QueueUrl='string',
    Entries=[
        {
            'Id': 'string',
```

```
        'ReceiptHandle': 'string'
    },
]
)
```

Parameters

- **QueueUrl** (*string*) – [REQUIRED]

The URL of the Amazon SQS queue to take action on.

- **Entries** (*list*) – [REQUIRED]

A list of receipt handles for the messages to be deleted.

– (*dict*) –

Encloses a receipt handle and an identifier for it.

- * **Id** (*string*) – [REQUIRED]

An identifier for this particular receipt handle. This is used to communicate the result. Note that the `Id`s of a batch request need to be unique within the request.

- * **ReceiptHandle** (*string*) – [REQUIRED]

A receipt handle.

Return type dict

Returns

Response Syntax

```
{
  'Successful': [
    {
      'Id': 'string'
    },
  ],
  'Failed': [
    {
      'Id': 'string',
      'SenderFault': True|False,
      'Code': 'string',
      'Message': 'string'
    },
  ],
}
```

Response Structure

- (*dict*) –

For each message in the batch, the response contains a `DeleteMessageBatchResultEntry` tag if the message is deleted or a `BatchResultErrorEntry` tag if the message cannot be deleted.

– **Successful** (*list*) –

A list of `DeleteMessageBatchResultEntry` items.

* (*dict*) –

Encloses the id an entry in `DeleteMessageBatch`.

- **Id** (*string*) –

Represents a successfully deleted message.

– **Failed** (*list*) –

A list of BatchResultErrorEntry items.

* (*dict*) –

This is used in the responses of batch API to give a detailed description of the result of an action on each entry in the request.

· **Id** (*string*) –

The id of an entry in a batch request.

· **SenderFault** (*boolean*) –

Whether the error happened due to the sender's fault.

· **Code** (*string*) –

An error code representing why the action failed on this entry.

· **Message** (*string*) –

A message explaining why the action failed on this entry.

delete_queue (***kwargs*)

Deletes the queue specified by the **queue URL**, regardless of whether the queue is empty. If the specified queue does not exist, Amazon SQS returns a successful response.

Warning: Use `DeleteQueue` with care; once you delete your queue, any messages in the queue are no longer available.

When you delete a queue, the deletion process takes up to 60 seconds. Requests you send involving that queue during the 60 seconds might succeed. For example, a `SendMessage` request might succeed, but after the 60 seconds, the queue and that message you sent no longer exist. Also, when you delete a queue, you must wait at least 60 seconds before creating a queue with the same name.

We reserve the right to delete queues that have had no activity for more than 30 days. For more information, see [How Amazon SQS Queues Work](#) in the *Amazon SQS Developer Guide*.

Request Syntax

```
response = client.delete_queue(
    QueueUrl='string'
)
```

Parameters **QueueUrl** (*string*) – [REQUIRED]

The URL of the Amazon SQS queue to take action on.

Returns `None`

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters `operation_name` (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises `OperationNotPageableError` Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_queue_attributes (***kwargs*)

Gets attributes for the specified queue. The following attributes are supported:

- `All` - returns all values.
- `ApproximateNumberOfMessages` - returns the approximate number of visible messages in a queue. For more information, see [Resources Required to Process Messages](#) in the *Amazon SQS Developer Guide*.
- `ApproximateNumberOfMessagesNotVisible` - returns the approximate number of messages that are not timed-out and not deleted. For more information, see [Resources Required to Process Messages](#) in the *Amazon SQS Developer Guide*.
- `VisibilityTimeout` - returns the visibility timeout for the queue. For more information about visibility timeout, see [Visibility Timeout](#) in the *Amazon SQS Developer Guide*.
- `CreatedTimestamp` - returns the time when the queue was created (epoch time in seconds).
- `LastModifiedTimestamp` - returns the time when the queue was last changed (epoch time in seconds).
- `Policy` - returns the queue's policy.
- `MaximumMessageSize` - returns the limit of how many bytes a message can contain before Amazon SQS rejects it.
- `MessageRetentionPeriod` - returns the number of seconds Amazon SQS retains a message.
- `QueueArn` - returns the queue's Amazon resource name (ARN).
- `ApproximateNumberOfMessagesDelayed` - returns the approximate number of messages that are pending to be added to the queue.
- `DelaySeconds` - returns the default delay on the queue in seconds.
- `ReceiveMessageWaitTimeSeconds` - returns the time for which a `ReceiveMessage` call will wait for a message to arrive.
- `RedrivePolicy` - returns the parameters for dead letter queue functionality of the source queue. For more information about `RedrivePolicy` and dead letter queues, see [Using Amazon SQS Dead Letter Queues](#) in the *Amazon SQS Developer Guide*.

Note: Going forward, new attributes might be added. If you are writing code that calls this action, we recommend that you structure your code so that it can handle new attributes gracefully.

Note: Some API actions take lists of parameters. These lists are specified using the `param.n` notation. Values of `n` are integers starting from 1. For example, a parameter list with two elements looks like this:

`Attribute.1=this`

`Attribute.2=that`

Request Syntax

```
response = client.get_queue_attributes(
    QueueUrl='string',
    AttributeNames=[
        'Policy'|'VisibilityTimeout'|'MaximumMessageSize'|'MessageRetentionPeriod'|'ApproximateNumberOfMessagesDelayed'|'DelaySeconds'|'ReceiveMessageWaitTimeSeconds'|'RedrivePolicy'
    ]
)
```


)

Parameters

- **QueueUrl** (*string*) – [REQUIRED]
The URL of the Amazon SQS queue to take action on.
- **AttributeNames** (*list*) – A list of attributes to retrieve information for.
– (*string*) –

Return type dict

Returns

Response Syntax

```
{
  'Attributes': {
    'string': 'string'
  }
}
```

Response Structure

- (*dict*) – A list of returned queue attributes.
– **Attributes** (*dict*) –
A map of attributes to the respective values.
* (*string*) –
The name of a queue attribute.
· (*string*) –
The value of a queue attribute.

get_queue_url (***kwargs*)

Returns the URL of an existing queue. This action provides a simple way to retrieve the URL of an Amazon SQS queue.

To access a queue that belongs to another AWS account, use the `QueueOwnerAWSAccountId` parameter to specify the account ID of the queue's owner. The queue's owner must grant you permission to access the queue. For more information about shared queue access, see [AddPermission](#) or go to [Shared Queues](#) in the *Amazon SQS Developer Guide*.

Request Syntax

```
response = client.get_queue_url(
    QueueName='string',
    QueueOwnerAWSAccountId='string'
)
```

Parameters

- **QueueName** (*string*) – [REQUIRED]
The name of the queue whose URL must be fetched. Maximum 80 characters; alphanumeric characters, hyphens (-), and underscores (_) are allowed.
- **QueueOwnerAWSAccountId** (*string*) – The AWS account ID of the account that created the queue.

Return type dict

Returns

Response Syntax

```
{
    'QueueUrl': 'string'
}
```

Response Structure

- (*dict*) –

For more information, see [Responses](#) in the *Amazon SQS Developer Guide* .

- **QueueUrl** (*string*) –

The URL for the queue.

get_waiter (*waiter_name*)

list_dead_letter_source_queues (***kwargs*)

Returns a list of your queues that have the RedrivePolicy queue attribute configured with a dead letter queue.

For more information about using dead letter queues, see [Using Amazon SQS Dead Letter Queues](#) .

Request Syntax

```
response = client.list_dead_letter_source_queues(
    QueueUrl='string'
)
```

Parameters **QueueUrl** (*string*) – **[REQUIRED]** The queue URL of a dead letter queue.

Return type dict

Returns**Response Syntax**

```
{
    'queueUrls': [
        'string',
    ]
}
```

Response Structure

- (*dict*) – A list of your dead letter source queues.
 - **queueUrls** (*list*) – A list of source queue URLs that have the RedrivePolicy queue attribute configured with a dead letter queue.
 - * (*string*) –

list_queues (***kwargs*)

Returns a list of your queues. The maximum number of queues that can be returned is 1000. If you specify a value for the optional `QueueNamePrefix` parameter, only queues with a name beginning with the specified value are returned.

Request Syntax

```
response = client.list_queues(
    QueueNamePrefix='string'
)
```

Parameters **QueueNamePrefix** (*string*) – A string to use for filtering the list results. Only those queues whose name begins with the specified string are returned.

Return type dict

Returns**Response Syntax**

```
{
    'QueueUrls': [
        'string',
    ]
}
```

Response Structure

- (*dict*) – A list of your queues.
 - **QueueUrls** (*list*) –
 - A list of queue URLs, up to 1000 entries.
 - * (*string*) –

purge_queue (***kwargs*)

Deletes the messages in a queue specified by the **queue URL** .

Warning: When you use the `PurgeQueue` API, the deleted messages in the queue cannot be retrieved.

When you purge a queue, the message deletion process takes up to 60 seconds. All messages sent to the queue before calling `PurgeQueue` will be deleted; messages sent to the queue while it is being purged may be deleted. While the queue is being purged, messages sent to the queue before `PurgeQueue` was called may be received, but will be deleted within the next minute.

Request Syntax

```
response = client.purge_queue(
    QueueUrl='string'
)
```

Parameters **QueueUrl** (*string*) – [REQUIRED]

The queue URL of the queue to delete the messages from when using the `PurgeQueue` API.

Returns None

receive_message (***kwargs*)

Retrieves one or more messages, with a maximum limit of 10 messages, from the specified queue. Long poll support is enabled by using the `WaitTimeSeconds` parameter. For more information, see [Amazon SQS Long Poll](#) in the *Amazon SQS Developer Guide* .

Short poll is the default behavior where a weighted random set of machines is sampled on a `ReceiveMessage` call. This means only the messages on the sampled machines are returned. If the number of messages in the queue is small (less than 1000), it is likely you will get fewer messages than you requested per `ReceiveMessage` call. If the number of messages in the queue is extremely small, you might not receive any messages in a particular `ReceiveMessage` response; in which case you should repeat the request.

For each message returned, the response includes the following:

- Message body
- MD5 digest of the message body. For information about MD5, go to <http://www.faqs.org/rfcs/rfc1321.html> .
- Message ID you received when you sent the message to the queue.
- Receipt handle.
- Message attributes.

- MD5 digest of the message attributes.

The receipt handle is the identifier you must provide when deleting the message. For more information, see [Queue and Message Identifiers](#) in the *Amazon SQS Developer Guide* .

You can provide the `VisibilityTimeout` parameter in your request, which will be applied to the messages that Amazon SQS returns in the response. If you do not include the parameter, the overall visibility timeout for the queue is used for the returned messages. For more information, see [Visibility Timeout](#) in the *Amazon SQS Developer Guide* .

Note: Going forward, new attributes might be added. If you are writing code that calls this action, we recommend that you structure your code so that it can handle new attributes gracefully.

Request Syntax

```
response = client.receive_message(  
    QueueUrl='string',  
    AttributeNames=[  
        'Policy'|'VisibilityTimeout'|'MaximumMessageSize'|'MessageRetentionPeriod'|'ApproximateFirstReceiveTimestamp',  
    ],  
    MessageAttributeNames=[  
        'string',  
    ],  
    MaxNumberOfMessages=123,  
    VisibilityTimeout=123,  
    WaitTimeSeconds=123  
)
```

Parameters

- **QueueUrl** (*string*) – **[REQUIRED]**

The URL of the Amazon SQS queue to take action on.

- **AttributeNames** (*list*) – A list of attributes that need to be returned along with each message.

The following lists the names and descriptions of the attributes that can be returned:

- All - returns all values.
- `ApproximateFirstReceiveTimestamp` - returns the time when the message was first received from the queue (epoch time in milliseconds).
- `ApproximateReceiveCount` - returns the number of times a message has been received from the queue but not deleted.
- `SenderId` - returns the AWS account number (or the IP address, if anonymous access is allowed) of the sender.
- `SentTimestamp` - returns the time when the message was sent to the queue (epoch time in milliseconds).
- (*string*) –
- **MessageAttributeNames** (*list*) – The name of the message attribute, where *N* is the index. The message attribute name can contain the following characters: A-Z, a-z, 0-9, underscore (_), hyphen (-), and period (.). The name must not start or end with a period, and it should not have successive periods. The name is case sensitive and must be unique among all attribute names for the message. The name can be up to 256 characters long. The name cannot start with “AWS.” or “Amazon.” (or any variations in casing), because these prefixes are reserved for use by Amazon Web Services.

When using `ReceiveMessage`, you can send a list of attribute names to receive, or you can return all of the attributes by specifying “All” or “.” in your request. You can also use “foo.*” to return all message attributes starting with the “foo” prefix.

– (string) –

- **MaxNumberOfMessages** (*integer*) – The maximum number of messages to return. Amazon SQS never returns more messages than this value but may return fewer. Values can be from 1 to 10. Default is 1.

All of the messages are not necessarily returned.

- **VisibilityTimeout** (*integer*) – The duration (in seconds) that the received messages are hidden from subsequent retrieve requests after being retrieved by a `ReceiveMessage` request.
- **WaitTimeSeconds** (*integer*) – The duration (in seconds) for which the call will wait for a message to arrive in the queue before returning. If a message is available, the call will return sooner than `WaitTimeSeconds`.

Return type dict

Returns

Response Syntax

```
{
  'Messages': [
    {
      'MessageId': 'string',
      'ReceiptHandle': 'string',
      'MD5OfBody': 'string',
      'Body': 'string',
      'Attributes': {
        'string': 'string'
      },
      'MD5OfMessageAttributes': 'string',
      'MessageAttributes': {
        'string': {
          'StringValue': 'string',
          'BinaryValue': b'bytes',
          'StringListValues': [
            'string',
          ],
          'BinaryListValues': [
            b'bytes',
          ],
          'DataType': 'string'
        }
      }
    },
  ],
}
```

Response Structure

- (*dict*) – A list of received messages.
- **Messages** (*list*) –

A list of messages.

* (*dict*) –

An Amazon SQS message.

· **MessageId** (*string*) –

A unique identifier for the message. Message IDs are considered unique across all AWS accounts for an extended period of time.

- **ReceiptHandle** (*string*) –

An identifier associated with the act of receiving the message. A new receipt handle is returned every time you receive a message. When deleting a message, you provide the last received receipt handle to delete the message.

- **MD5OfBody** (*string*) –

An MD5 digest of the non-URL-encoded message body string.

- **Body** (*string*) –

The message's contents (not URL-encoded).

- **Attributes** (*dict*) –

`SenderId` , `SentTimestamp` ,
`ApproximateReceiveCount` , `and/or`
`ApproximateFirstReceiveTimestamp`
.
`SentTimestamp` `and`
`ApproximateFirstReceiveTimestamp` are each
returned as an integer representing the epoch time in
milliseconds.

- (*string*) –

The name of a queue attribute.

- (*string*) –

The value of a queue attribute.

- **MD5OfMessageAttributes** (*string*) –

An MD5 digest of the non-URL-encoded message attribute string. This can be used to verify that Amazon SQS received the message correctly. Amazon SQS first URL decodes the message before creating the MD5 digest. For information about MD5, go to <http://www.faqs.org/rfcs/rfc1321.html> .

- **MessageAttributes** (*dict*) –

Each message attribute consists of a Name, Type, and Value. For more information, see [Message Attribute Items](#) .

- (*string*) –

- (*dict*) –

The user-specified message attribute value. For string data types, the value attribute has the same restrictions on the content as the message body. For more information, see [SendMessage](#) .

Name, type, and value must not be empty or null. In addition, the message body should not be empty or null. All parts of the message attribute, including name, type, and value, are included in the message size restriction, which is currently 256 KB (262,144 bytes).

- **StringValue** (*string*) –

Strings are Unicode with UTF8 binary encoding. For a list of code values, see

http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters

- **BinaryValue** (*bytes*) –
Binary type attributes can store any binary data, for example, compressed data, encrypted data, or images.
- **StringListValues** (*list*) –
Not implemented. Reserved for future use.
- (*string*) –
- **BinaryListValues** (*list*) –
Not implemented. Reserved for future use.
- (*bytes*) –
- **DataType** (*string*) –
Amazon SQS supports the following logical data types: String, Number, and Binary. In addition, you can append your own custom labels. For more information, see [Message Attribute Data Types](#).

remove_permission (***kwargs*)

Revokes any permissions in the queue policy that matches the specified `Label` parameter. Only the owner of the queue can remove permissions.

Request Syntax

```
response = client.remove_permission(
    QueueUrl='string',
    Label='string'
)
```

Parameters

- **QueueUrl** (*string*) – **[REQUIRED]**
The URL of the Amazon SQS queue to take action on.
- **Label** (*string*) – **[REQUIRED]**
The identification of the permission to remove. This is the label added with the `AddPermission` action.

Returns None

send_message (***kwargs*)

Delivers a message to the specified queue. With Amazon SQS, you now have the ability to send large payload messages that are up to 256KB (262,144 bytes) in size. To send large payloads, you must use an AWS SDK that supports SigV4 signing. To verify whether SigV4 is supported for an AWS SDK, check the SDK release notes.

Warning: The following list shows the characters (in Unicode) allowed in your message, according to the W3C XML specification. For more information, go to <http://www.w3.org/TR/REC-xml/#charsets>. If you send any characters not included in the list, your request will be rejected.
 #x9 | #xA | #xD | [#x20 to #xD7FF] | [#xE000 to #xFFFF] | [#x10000 to #x10FFFF]

Request Syntax

```
response = client.send_message(
    QueueUrl='string',
    MessageBody='string',
    DelaySeconds=123,
```

```
MessageAttributes={
    'string': {
        'StringValue': 'string',
        'BinaryValue': b'bytes',
        'StringListValues': [
            'string',
        ],
        'BinaryListValues': [
            b'bytes',
        ],
        'DataType': 'string'
    }
}
```

Parameters

- **QueueUrl** (*string*) – [REQUIRED]

The URL of the Amazon SQS queue to take action on.

- **MessageBody** (*string*) – [REQUIRED]

The message to send. String maximum 256 KB in size. For a list of allowed characters, see the preceding important note.

- **DelaySeconds** (*integer*) – The number of seconds (0 to 900 - 15 minutes) to delay a specific message. Messages with a positive `DelaySeconds` value become available for processing after the delay time is finished. If you don't specify a value, the default value for the queue applies.
- **MessageAttributes** (*dict*) – Each message attribute consists of a Name, Type, and Value. For more information, see [Message Attribute Items](#).
 - (*string*) –
 - * (*dict*) –

The user-specified message attribute value. For string data types, the value attribute has the same restrictions on the content as the message body. For more information, see [SendMessage](#).

Name, type, and value must not be empty or null. In addition, the message body should not be empty or null. All parts of the message attribute, including name, type, and value, are included in the message size restriction, which is currently 256 KB (262,144 bytes).

- **StringValue** (*string*) –

Strings are Unicode with UTF8 binary encoding. For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters.

- **BinaryValue** (*bytes*) –

Binary type attributes can store any binary data, for example, compressed data, encrypted data, or images.

- **StringListValues** (*list*) –

Not implemented. Reserved for future use.

- (*string*) –

- **BinaryListValues** (*list*) –

Not implemented. Reserved for future use.

- (*bytes*) –

- **DataType** (*string*) – [REQUIRED]

Amazon SQS supports the following logical data types: String, Number, and Binary. In addition, you can append your own custom labels. For more information, see [Message Attribute Data Types](#).

Return type dict

Returns

Response Syntax

```
{
  'MD5OfMessageBody': 'string',
  'MD5OfMessageAttributes': 'string',
  'MessageId': 'string'
}
```

Response Structure

- (*dict*) –

The MD5OfMessageBody and MessageId elements.

- **MD5OfMessageBody** (*string*) –

An MD5 digest of the non-URL-encoded message body string. This can be used to verify that Amazon SQS received the message correctly. Amazon SQS first URL decodes the message before creating the MD5 digest. For information about MD5, go to <http://www.faqs.org/rfcs/rfc1321.html>.

- **MD5OfMessageAttributes** (*string*) –

An MD5 digest of the non-URL-encoded message attribute string. This can be used to verify that Amazon SQS received the message correctly. Amazon SQS first URL decodes the message before creating the MD5 digest. For information about MD5, go to <http://www.faqs.org/rfcs/rfc1321.html>.

- **MessageId** (*string*) –

An element containing the message ID of the message sent to the queue. For more information, see [Queue and Message Identifiers](#) in the *Amazon SQS Developer Guide*.

send_message_batch (***kwargs*)

Delivers up to ten messages to the specified queue. This is a batch version of `SendMessage`. The result of the send action on each message is reported individually in the response. The maximum allowed individual message size is 256 KB (262,144 bytes).

The maximum total payload size (i.e., the sum of all a batch's individual message lengths) is also 256 KB (262,144 bytes).

If the `DelaySeconds` parameter is not specified for an entry, the default for the queue is used.

Warning: The following list shows the characters (in Unicode) that are allowed in your message, according to the W3C XML specification. For more information, go to <http://www.faqs.org/rfcs/rfc1321.html>. If you send any characters that are not included in the list, your request will be rejected.

#x9 | #xA | #xD | [#x20 to #xD7FF] | [#xE000 to #xFFFD] | [#x10000 to #x10FFFF]

Warning: Because the batch request can result in a combination of successful and unsuccessful actions, you should check for batch errors even when the call returns an HTTP status code of 200.

Note: Some API actions take lists of parameters. These lists are specified using the `param.n` notation. Values of `n` are integers starting from 1. For example, a parameter list with two elements looks like this:

`Attribute.1=this`

`Attribute.2=that`

Request Syntax

```
response = client.send_message_batch(
    QueueUrl='string',
    Entries=[
        {
            'Id': 'string',
            'MessageBody': 'string',
            'DelaySeconds': 123,
            'MessageAttributes': {
                'string': {
                    'StringValue': 'string',
                    'BinaryValue': b'bytes',
                    'StringListValues': [
                        'string',
                    ],
                    'BinaryListValues': [
                        b'bytes',
                    ],
                    'DataType': 'string'
                }
            }
        },
    ]
)
```

Parameters

- **QueueUrl** (*string*) – [REQUIRED]

The URL of the Amazon SQS queue to take action on.

- **Entries** (*list*) – [REQUIRED]

A list of `SendMessageBatchRequestEntry` items.

– (*dict*) –

Contains the details of a single Amazon SQS message along with a `Id`.

- * **Id** (*string*) – [REQUIRED]

An identifier for the message in this batch. This is used to communicate the result. Note that the `Id`s of a batch request need to be unique within the request.

- * **MessageBody** (*string*) – [REQUIRED]

Body of the message.

- * **DelaySeconds** (*integer*) –

The number of seconds for which the message has to be delayed.

- * **MessageAttributes** (*dict*) –

Each message attribute consists of a Name, Type, and Value. For more information, see [Message Attribute Items](#).

- (*string*) –
- (*dict*) –

The user-specified message attribute value. For string data types, the value attribute has the same restrictions on the content as the message body. For more information, see [SendMessage](#).

Name, type, and value must not be empty or null. In addition, the message body should not be empty or null. All parts of the message attribute, including name, type, and value, are included in the message size restriction, which is currently 256 KB (262,144 bytes).

- **StringValue** (*string*) –

Strings are Unicode with UTF8 binary encoding. For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters.

- **BinaryValue** (*bytes*) –

Binary type attributes can store any binary data, for example, compressed data, encrypted data, or images.

- **StringListValues** (*list*) –

Not implemented. Reserved for future use.

- (*string*) –

- **BinaryListValues** (*list*) –

Not implemented. Reserved for future use.

- (*bytes*) –

- **DataType** (*string*) – [REQUIRED]

Amazon SQS supports the following logical data types: String, Number, and Binary. In addition, you can append your own custom labels. For more information, see [Message Attribute Data Types](#).

Return type dict

Returns

Response Syntax

```
{
  'Successful': [
    {
      'Id': 'string',
      'MessageId': 'string',
      'MD5OfMessageBody': 'string',
      'MD5OfMessageAttributes': 'string'
    },
  ],
  'Failed': [
    {
      'Id': 'string',
      'SenderFault': True|False,
      'Code': 'string',
      'Message': 'string'
    },
  ],
}
```

```
    },  
  ]  
}
```

Response Structure

- (*dict*) –

For each message in the batch, the response contains a `SendMessageBatchResultEntry` tag if the message succeeds or a `BatchResultErrorEntry` tag if the message fails.

- **Successful** (*list*) –

A list of `SendMessageBatchResultEntry` items.

- * (*dict*) –

Encloses a message ID for successfully enqueued message of a `SendMessageBatch`.

- **Id** (*string*) –

An identifier for the message in this batch.

- **MessageId** (*string*) –

An identifier for the message.

- **MD5OfMessageBody** (*string*) –

An MD5 digest of the non-URL-encoded message body string. This can be used to verify that Amazon SQS received the message correctly. Amazon SQS first URL decodes the message before creating the MD5 digest. For information about MD5, go to <http://www.faqs.org/rfcs/rfc1321.html>.

- **MD5OfMessageAttributes** (*string*) –

An MD5 digest of the non-URL-encoded message attribute string. This can be used to verify that Amazon SQS received the message batch correctly. Amazon SQS first URL decodes the message before creating the MD5 digest. For information about MD5, go to <http://www.faqs.org/rfcs/rfc1321.html>.

- **Failed** (*list*) –

A list of `BatchResultErrorEntry` items with the error detail about each message that could not be enqueued.

- * (*dict*) –

This is used in the responses of batch API to give a detailed description of the result of an action on each entry in the request.

- **Id** (*string*) –

The id of an entry in a batch request.

- **SenderFault** (*boolean*) –

Whether the error happened due to the sender's fault.

- **Code** (*string*) –

An error code representing why the action failed on this entry.

- **Message** (*string*) –

A message explaining why the action failed on this entry.

set_queue_attributes (***kwargs*)

Sets the value of one or more queue attributes. When you change a queue's attributes, the change can take

up to 60 seconds for most of the attributes to propagate throughout the SQS system. Changes made to the `MessageRetentionPeriod` attribute can take up to 15 minutes.

Note: Going forward, new attributes might be added. If you are writing code that calls this action, we recommend that you structure your code so that it can handle new attributes gracefully.

Request Syntax

```
response = client.set_queue_attributes(
    QueueUrl='string',
    Attributes={
        'string': 'string'
    }
)
```

Parameters

- **QueueUrl** (*string*) – **[REQUIRED]**

The URL of the Amazon SQS queue to take action on.

- **Attributes** (*dict*) – **[REQUIRED]**

A map of attributes to set.

The following lists the names, descriptions, and values of the special request parameters the `SetQueueAttributes` action uses:

- `DelaySeconds` - The time in seconds that the delivery of all messages in the queue will be delayed. An integer from 0 to 900 (15 minutes). The default for this attribute is 0 (zero).
- `MaximumMessageSize` - The limit of how many bytes a message can contain before Amazon SQS rejects it. An integer from 1024 bytes (1 KiB) up to 262144 bytes (256 KiB). The default for this attribute is 262144 (256 KiB).
- `MessageRetentionPeriod` - The number of seconds Amazon SQS retains a message. Integer representing seconds, from 60 (1 minute) to 1209600 (14 days). The default for this attribute is 345600 (4 days).
- `Policy` - The queue's policy. A valid AWS policy. For more information about policy structure, see [Overview of AWS IAM Policies](#) in the *Amazon IAM User Guide*.
- `ReceiveMessageWaitTimeSeconds` - The time for which a `ReceiveMessage` call will wait for a message to arrive. An integer from 0 to 20 (seconds). The default for this attribute is 0.
- `VisibilityTimeout` - The visibility timeout for the queue. An integer from 0 to 43200 (12 hours). The default for this attribute is 30. For more information about visibility timeout, see [Visibility Timeout](#) in the *Amazon SQS Developer Guide*.
- `RedrivePolicy` - The parameters for dead letter queue functionality of the source queue. For more information about `RedrivePolicy` and dead letter queues, see [Using Amazon SQS Dead Letter Queues](#) in the *Amazon SQS Developer Guide*.
- (*string*) –
The name of a queue attribute.
- * (*string*) –
The value of a queue attribute.

Returns None

Service Resource

`class SQS.ServiceResource`

A resource representing Amazon Simple Queue Service (SQS):

```
import boto3

sqs = boto3.resource('sqs')
```

These are the resource's available actions:

- `create_queue()`
- `get_queue_by_name()`

These are the resource's available sub-resources:

- `Message()`
- `Queue()`

These are the resource's available collections:

- `queues`

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

`create_queue(**kwargs)`

Creates a new queue, or returns the URL of an existing one. When you request `CreateQueue`, you provide a name for the queue. To successfully create a new queue, you must provide a name that is unique within the scope of your own queues.

Note: If you delete a queue, you must wait at least 60 seconds before creating a queue with the same name.

You may pass one or more attributes in the request. If you do not provide a value for any attribute, the queue will have the default value for that attribute. Permitted attributes are the same that can be set using `SetQueueAttributes`.

Note: Use `GetQueueUrl` to get a queue's URL. `GetQueueUrl` requires only the `QueueName` parameter.

If you provide the name of an existing queue, along with the exact names and values of all the queue's attributes, `CreateQueue` returns the queue URL for the existing queue. If the queue name, attribute names, or attribute values do not match an existing queue, `CreateQueue` returns an error.

Note: Some API actions take lists of parameters. These lists are specified using the `param.n` notation. Values of `n` are integers starting from 1. For example, a parameter list with two elements looks like this:

```
Attribute.1=this
Attribute.2=that
```

Request Syntax

```
queue = sqs.create_queue(
    QueueName='string',
    Attributes={
        'string': 'string'
    }
)
```

Parameters

- **QueueName** (*string*) – [REQUIRED]

The name for the queue to be created.

- **Attributes** (*dict*) – A map of attributes with their corresponding values.

The following lists the names, descriptions, and values of the special request parameters the `CreateQueue` action uses:

- `DelaySeconds` - The time in seconds that the delivery of all messages in the queue will be delayed. An integer from 0 to 900 (15 minutes). The default for this attribute is 0 (zero).
- `MaximumMessageSize` - The limit of how many bytes a message can contain before Amazon SQS rejects it. An integer from 1024 bytes (1 KiB) up to 262144 bytes (256 KiB). The default for this attribute is 262144 (256 KiB).
- `MessageRetentionPeriod` - The number of seconds Amazon SQS retains a message. Integer representing seconds, from 60 (1 minute) to 1209600 (14 days). The default for this attribute is 345600 (4 days).
- `Policy` - The queue's policy. A valid AWS policy. For more information about policy structure, see [Overview of AWS IAM Policies](#) in the *Amazon IAM User Guide*.
- `ReceiveMessageWaitTimeSeconds` - The time for which a `ReceiveMessage` call will wait for a message to arrive. An integer from 0 to 20 (seconds). The default for this attribute is 0.
- `VisibilityTimeout` - The visibility timeout for the queue. An integer from 0 to 43200 (12 hours). The default for this attribute is 30. For more information about visibility timeout, see [Visibility Timeout](#) in the *Amazon SQS Developer Guide*.
- (*string*) –
The name of a queue attribute.
* (*string*) –
The value of a queue attribute.

Return type `sqs.Queue`

Returns Queue resource

get_queue_by_name (***kwargs*)

Returns the URL of an existing queue. This action provides a simple way to retrieve the URL of an Amazon SQS queue.

To access a queue that belongs to another AWS account, use the `QueueOwnerAWSAccountId` parameter to specify the account ID of the queue's owner. The queue's owner must grant you permission to access the queue. For more information about shared queue access, see `AddPermission` or go to [Shared Queues](#) in the *Amazon SQS Developer Guide*.

Request Syntax

```
queue = sqs.get_queue_by_name(
    QueueName='string',
    QueueOwnerAWSAccountId='string'
)
```

Parameters

- **QueueName** (*string*) – [REQUIRED]

The name of the queue whose URL must be fetched. Maximum 80 characters; alphanumeric characters, hyphens (-), and underscores (_) are allowed.

- **QueueOwnerAWSAccountId** (*string*) – The AWS account ID of the account that created the queue.

Return type `sqs.Queue`

Returns Queue resource

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Message (*queue_url*, *receipt_handle*)

Creates a Message resource.:

```
message = sqs.Message('queue_url', 'receipt_handle')
```

Parameters

- **queue_url** (*string*) – The Message's queue_url identifier. This **must** be set.
- **receipt_handle** (*string*) – The Message's receipt_handle identifier. This **must** be set.

Return type `SQS.Message`

Returns A Message resource

Queue (*url*)

Creates a Queue resource.:

```
queue = sqs.Queue('url')
```

Parameters **url** (*string*) – The Queue's url identifier. This **must** be set.

Return type `SQS.Queue`

Returns A Queue resource

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

queues

all ()

Creates an iterable of all Queue resources in the collection.

Request Syntax

```
queue_iterator = sqs.queues.all()
```

Return type `list(sqs.Queue)`

Returns A list of Queue resources

filter (***kwargs*)

Creates an iterable of all Queue resources in the collection filtered by kwargs passed to method.

Request Syntax

```
queue_iterator = sqs.queues.filter(  
    QueueNamePrefix='string'  
)
```

Parameters **QueueNamePrefix** (*string*) – A string to use for filtering the list results. Only those queues whose name begins with the specified string are returned.

Return type `list(sqs.Queue)`
Returns A list of Queue resources

limit (***kwargs*)

Creates an iterable up to a specified amount of Queue resources in the collection.

Request Syntax

```
queue_iterator = sqs.queues.limit(
    count=123
)
```

Parameters **count** (*integer*) – The limit to the number of resources in the iterable.

Return type `list(sqs.Queue)`

Returns A list of Queue resources

page_size (***kwargs*)

Creates an iterable of all Queue resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
queue_iterator = sqs.queues.page_size(
    count=123
)
```

Parameters **count** (*integer*) – The number of items returned by each service call

Return type `list(sqs.Queue)`

Returns A list of Queue resources

Message

class `SQS.Message` (*queue_url, receipt_handle*)

A resource representing an Amazon Simple Queue Service (SQS) Message:

```
import boto3
```

```
sqs = boto3.resource('sqs')
message = sqs.Message('queue_url', 'receipt_handle')
```

Parameters

- **queue_url** (*string*) – The Message’s queue_url identifier. This **must** be set.
- **receipt_handle** (*string*) – The Message’s receipt_handle identifier. This **must** be set.

These are the resource’s available identifiers:

- `queue_url`
- `receipt_handle`

These are the resource’s available attributes:

- `attributes`
- `body`
- `md5_of_body`
- `md5_of_message_attributes`
- `message_attributes`
- `message_id`

These are the resource’s available actions:

- `change_visibility()`

- `delete()`

These are the resource's available sub-resources:

- `Queue()`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

queue_url

(*string*) The Message's queue_url identifier. This **must** be set.

receipt_handle

(*string*) The Message's receipt_handle identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

attributes

(*dict*)

SenderId , SentTimestamp , ApproximateReceiveCount , and/or ApproximateFirstReceiveTimestamp . SentTimestamp and ApproximateFirstReceiveTimestamp are each returned as an integer representing the [epoch time](#) in milliseconds.

body

(*string*)

The message's contents (not URL-encoded).

md5_of_body

(*string*)

An MD5 digest of the non-URL-encoded message body string.

md5_of_message_attributes

(*string*)

An MD5 digest of the non-URL-encoded message attribute string. This can be used to verify that Amazon SQS received the message correctly. Amazon SQS first URL decodes the message before creating the MD5 digest. For information about MD5, go to <http://www.faqs.org/rfcs/rfc1321.html> .

message_attributes

(*dict*)

Each message attribute consists of a Name, Type, and Value. For more information, see [Message Attribute Items](#) .

message_id

(*string*)

A unique identifier for the message. Message IDs are considered unique across all AWS accounts for an extended period of time.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

change_visibility (***kwargs*)

Changes the visibility timeout of a specified message in a queue to a new value. The maximum allowed timeout value you can set the value to is 12 hours. This means you can't extend the timeout of a message

in an existing queue to more than a total visibility timeout of 12 hours. (For more information visibility timeout, see [Visibility Timeout](#) in the *Amazon SQS Developer Guide*.)

For example, let's say you have a message and its default message visibility timeout is 30 minutes. You could call `ChangeMessageVisibility` with a value of two hours and the effective timeout would be two hours and 30 minutes. When that time comes near you could again extend the time out by calling `ChangeMessageVisibility`, but this time the maximum allowed timeout would be 9 hours and 30 minutes.

Note: There is a 120,000 limit for the number of inflight messages per queue. Messages are inflight after they have been received from the queue by a consuming component, but have not yet been deleted from the queue. If you reach the 120,000 limit, you will receive an `OverLimit` error message from Amazon SQS. To help avoid reaching the limit, you should delete the messages from the queue after they have been processed. You can also increase the number of queues you use to process the messages.

Warning: If you attempt to set the `VisibilityTimeout` to an amount more than the maximum time left, Amazon SQS returns an error. It will not automatically recalculate and increase the timeout to the maximum time remaining.

Warning: Unlike with a queue, when you change the visibility timeout for a specific message, that timeout value is applied immediately but is not saved in memory for that message. If you don't delete a message after it is received, the visibility timeout for the message the next time it is received reverts to the original timeout value, not the value you set with the `ChangeMessageVisibility` action.

Request Syntax

```
response = message.change_visibility(
    VisibilityTimeout=123
)
```

Parameters `VisibilityTimeout` (*integer*) – [REQUIRED]

The new value (in seconds - from 0 to 43200 - maximum 12 hours) for the message's visibility timeout.

Returns None

`delete()`

Deletes the specified message from the specified queue. You specify the message by using the message's receipt handle and not the message ID you received when you sent the message. Even if the message is locked by another reader due to the visibility timeout setting, it is still deleted from the queue. If you leave a message in the queue for longer than the queue's configured retention period, Amazon SQS automatically deletes it.

Note: The receipt handle is associated with a specific instance of receiving the message. If you receive a message more than once, the receipt handle you get each time you receive the message is different. When you request `DeleteMessage`, if you don't provide the most recently received receipt handle for the message, the request will still succeed, but the message might not be deleted.

Warning: It is possible you will receive a message even after you have deleted it. This might happen on rare occasions if one of the servers storing a copy of the message is unavailable when you request to delete the message. The copy remains on the server and might be returned to you again on a subsequent receive request. You should create your system to be idempotent so that receiving a particular message more than once is not a problem.

Request Syntax

```
response = message.delete()
```

Returns None

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

`Queue()`

Creates a Queue resource.:

```
queue = message.Queue()
```

Return type `SQS.Queue`

Returns A Queue resource

Queue

class `SQS.Queue(url)`

A resource representing an Amazon Simple Queue Service (SQS) Queue:

```
import boto3
```

```
sqs = boto3.resource('sqs')
```

```
queue = sqs.Queue('url')
```

Parameters `url (string)` – The Queue's url identifier. This **must** be set.

These are the resource's available identifiers:

- `url`

These are the resource's available attributes:

- `attributes`

These are the resource's available actions:

- `add_permission()`
- `change_message_visibility_batch()`
- `delete()`
- `delete_messages()`
- `load()`
- `purge()`
- `receive_messages()`
- `reload()`
- `remove_permission()`
- `send_message()`
- `send_messages()`
- `set_attributes()`

These are the resource's available sub-resources:

- `Message()`

These are the resource's available collections:

- `dead_letter_source_queues`

Identifiers

Identifiers are properties of a resource that are set upon instantiation of the resource. For more information about identifiers refer to the [Resources Introduction Guide](#).

url

(*string*) The Queue's url identifier. This **must** be set.

Attributes

Attributes provide access to the properties of a resource. Attributes are lazy-loaded the first time one is accessed via the `load()` method. For more information about attributes refer to the [Resources Introduction Guide](#).

attributes

(*dict*)

A map of attributes to the respective values.

Actions

Actions call operations on resources. They may automatically handle the passing in of arguments set from identifiers and some attributes. For more information about actions refer to the [Resources Introduction Guide](#).

add_permission (***kwargs*)

Adds a permission to a queue for a specific [principal](#) . This allows for sharing access to the queue.

When you create a queue, you have full control access rights for the queue. Only you (as owner of the queue) can grant or deny permissions to the queue. For more information about these permissions, see [Shared Queues](#) in the *Amazon SQS Developer Guide* .

Note: `AddPermission` writes an Amazon SQS-generated policy. If you want to write your own policy, use `SetQueueAttributes` to upload your policy. For more information about writing your own policy, see [Using The Access Policy Language](#) in the *Amazon SQS Developer Guide* .

Note: Some API actions take lists of parameters. These lists are specified using the `param.n` notation. Values of `n` are integers starting from 1. For example, a parameter list with two elements looks like this:

```
Attribute.1=this
```

```
Attribute.2=that
```

Request Syntax

```
response = queue.add_permission(
    Label='string',
    AWSAccountIds=[
        'string',
    ],
    Actions=[
        'string',
    ]
)
```

Parameters

- **Label** (*string*) – **[REQUIRED]**

The unique identification of the permission you're setting (e.g., `AliceSendMessage`). Constraints: Maximum 80 characters; alphanumeric characters, hyphens (-), and underscores (_) are allowed.

- **AWSAccountIds** (*list*) – **[REQUIRED]**

The AWS account number of the [principal](#) who will be given permission. The principal must have an AWS account, but does not need to be signed up for Amazon SQS. For information about locating the AWS account identification, see [Your AWS Identifiers](#) in the *Amazon SQS Developer Guide* .

- (string) –
- **Actions** (list) – [REQUIRED]
The action the client wants to allow for the specified principal. The following are valid values: * | SendMessage | ReceiveMessage | DeleteMessage | ChangeMessageVisibility | GetQueueAttributes | GetQueueUrl . For more information about these actions, see [Understanding Permissions](#) in the *Amazon SQS Developer Guide* .
Specifying SendMessage , DeleteMessage , or ChangeMessageVisibility for the ActionName.n also grants permissions for the corresponding batch versions of those actions: SendMessageBatch , DeleteMessageBatch , and ChangeMessageVisibilityBatch .
- (string) –

Returns None

change_message_visibility_batch (**kwargs)

Changes the visibility timeout of multiple messages. This is a batch version of ChangeMessageVisibility . The result of the action on each message is reported individually in the response. You can send up to 10 ChangeMessageVisibility requests with each ChangeMessageVisibilityBatch action.

Warning: Because the batch request can result in a combination of successful and unsuccessful actions, you should check for batch errors even when the call returns an HTTP status code of 200.

Note: Some API actions take lists of parameters. These lists are specified using the param.n notation. Values of n are integers starting from 1. For example, a parameter list with two elements looks like this:

Attribute.1=this

Attribute.2=that

Request Syntax

```
response = queue.change_message_visibility_batch(
    Entries=[
        {
            'Id': 'string',
            'ReceiptHandle': 'string',
            'VisibilityTimeout': 123
        },
    ],
)
```

Parameters **Entries** (list) – [REQUIRED]

A list of receipt handles of the messages for which the visibility timeout must be changed.

- (dict) –
Encloses a receipt handle and an entry id for each message in ChangeMessageVisibilityBatch .

Warning: All of the following parameters are list parameters that must be prefixed with `ChangeMessageVisibilityBatchRequestEntry.n`, where `n` is an integer value starting with 1. For example, a parameter list for this action might look like this:

```
ChangeMessageVisibilityBatchRequestEntry.1.Id=change_visibility_msg_2
ChangeMessageVisibilityBatchRequestEntry.1.ReceiptHandle=Your_Receipt_
ChangeMessageVisibilityBatchRequestEntry.1.VisibilityTimeout=45
```

– **Id** (*string*) – **[REQUIRED]**

An identifier for this particular receipt handle. This is used to communicate the result. Note that the `Id`s of a batch request need to be unique within the request.

– **ReceiptHandle** (*string*) – **[REQUIRED]**

A receipt handle.

– **VisibilityTimeout** (*integer*) –

The new value (in seconds) for the message's visibility timeout.

Return type dict

Returns

Response Syntax

```
{
  'Successful': [
    {
      'Id': 'string'
    },
  ],
  'Failed': [
    {
      'Id': 'string',
      'SenderFault': True|False,
      'Code': 'string',
      'Message': 'string'
    },
  ],
}
```

Response Structure

- (*dict*) –

For each message in the batch, the response contains a `ChangeMessageVisibilityBatchResultEntry` tag if the message succeeds or a `BatchResultErrorEntry` tag if the message fails.

– **Successful** (*list*) –

A list of `ChangeMessageVisibilityBatchResultEntry` items.

- * (*dict*) –

Encloses the id of an entry in `ChangeMessageVisibilityBatch`.

· **Id** (*string*) –

Represents a message whose visibility timeout has been changed successfully.

– **Failed** (*list*) –

A list of `BatchResultErrorEntry` items.

* (*dict*) –

This is used in the responses of batch API to give a detailed description of the result of an action on each entry in the request.

· **Id** (*string*) –

The id of an entry in a batch request.

· **SenderFault** (*boolean*) –

Whether the error happened due to the sender's fault.

· **Code** (*string*) –

An error code representing why the action failed on this entry.

· **Message** (*string*) –

A message explaining why the action failed on this entry.

delete ()

Deletes the queue specified by the **queue URL**, regardless of whether the queue is empty. If the specified queue does not exist, Amazon SQS returns a successful response.

Warning: Use `DeleteQueue` with care; once you delete your queue, any messages in the queue are no longer available.

When you delete a queue, the deletion process takes up to 60 seconds. Requests you send involving that queue during the 60 seconds might succeed. For example, a `SendMessage` request might succeed, but after the 60 seconds, the queue and that message you sent no longer exist. Also, when you delete a queue, you must wait at least 60 seconds before creating a queue with the same name.

We reserve the right to delete queues that have had no activity for more than 30 days. For more information, see [How Amazon SQS Queues Work](#) in the *Amazon SQS Developer Guide*.

Request Syntax

```
response = queue.delete()
```

Returns None

delete_messages (***kwargs*)

Deletes up to ten messages from the specified queue. This is a batch version of `DeleteMessage`. The result of the delete action on each message is reported individually in the response.

Warning: Because the batch request can result in a combination of successful and unsuccessful actions, you should check for batch errors even when the call returns an HTTP status code of 200.

Note: Some API actions take lists of parameters. These lists are specified using the `param.n` notation. Values of `n` are integers starting from 1. For example, a parameter list with two elements looks like this:

```
Attribute.1=this
```

```
Attribute.2=that
```

Request Syntax


```
response = queue.delete_messages(
    Entries=[
        {
            'Id': 'string',
            'ReceiptHandle': 'string'
        },
    ]
)
```

Parameters *Entries (list)* – **[REQUIRED]**

A list of receipt handles for the messages to be deleted.

- (*dict*) –

Encloses a receipt handle and an identifier for it.

- **Id** (*string*) – **[REQUIRED]**

An identifier for this particular receipt handle. This is used to communicate the result. Note that the `Ids` of a batch request need to be unique within the request.

- **ReceiptHandle** (*string*) – **[REQUIRED]**

A receipt handle.

Return type `dict`

Returns

Response Syntax

```
{
    'Successful': [
        {
            'Id': 'string'
        },
    ],
    'Failed': [
        {
            'Id': 'string',
            'SenderFault': True|False,
            'Code': 'string',
            'Message': 'string'
        },
    ]
}
```

Response Structure

- (*dict*) –

For each message in the batch, the response contains a `DeleteMessageBatchResultEntry` tag if the message is deleted or a `BatchResultErrorEntry` tag if the message cannot be deleted.

- **Successful** (*list*) –

A list of `DeleteMessageBatchResultEntry` items.

- * (*dict*) –

Encloses the id an entry in `DeleteMessageBatch`.

- **Id** (*string*) –

Represents a successfully deleted message.

– **Failed** (*list*) –

A list of `BatchResultErrorEntry` items.

* (*dict*) –

This is used in the responses of batch API to give a detailed description of the result of an action on each entry in the request.

· **Id** (*string*) –

The id of an entry in a batch request.

· **SenderFault** (*boolean*) –

Whether the error happened due to the sender's fault.

· **Code** (*string*) –

An error code representing why the action failed on this entry.

· **Message** (*string*) –

A message explaining why the action failed on this entry.

load()

Calls `sqs.Client.get_queue_attributes()` to update the attributes of the Queue resource

Request Syntax

```
queue.load()
```

Returns None

purge()

Deletes the messages in a queue specified by the **queue URL** .

Warning: When you use the `PurgeQueue` API, the deleted messages in the queue cannot be retrieved.

When you purge a queue, the message deletion process takes up to 60 seconds. All messages sent to the queue before calling `PurgeQueue` will be deleted; messages sent to the queue while it is being purged may be deleted. While the queue is being purged, messages sent to the queue before `PurgeQueue` was called may be received, but will be deleted within the next minute.

Request Syntax

```
response = queue.purge()
```

Returns None

receive_messages (***kwargs*)

Retrieves one or more messages, with a maximum limit of 10 messages, from the specified queue. Long poll support is enabled by using the `WaitTimeSeconds` parameter. For more information, see [Amazon SQS Long Poll](#) in the *Amazon SQS Developer Guide* .

Short poll is the default behavior where a weighted random set of machines is sampled on a `ReceiveMessage` call. This means only the messages on the sampled machines are returned. If the number of messages in the queue is small (less than 1000), it is likely you will get fewer messages than you requested per `ReceiveMessage` call. If the number of messages in the queue is extremely small, you might not receive any messages in a particular `ReceiveMessage` response; in which case you should repeat the request.

For each message returned, the response includes the following:

- Message body

- MD5 digest of the message body. For information about MD5, go to <http://www.faqs.org/rfcs/rfc1321.html>.
- Message ID you received when you sent the message to the queue.
- Receipt handle.
- Message attributes.
- MD5 digest of the message attributes.

The receipt handle is the identifier you must provide when deleting the message. For more information, see [Queue and Message Identifiers](#) in the *Amazon SQS Developer Guide*.

You can provide the `VisibilityTimeout` parameter in your request, which will be applied to the messages that Amazon SQS returns in the response. If you do not include the parameter, the overall visibility timeout for the queue is used for the returned messages. For more information, see [Visibility Timeout](#) in the *Amazon SQS Developer Guide*.

Note: Going forward, new attributes might be added. If you are writing code that calls this action, we recommend that you structure your code so that it can handle new attributes gracefully.

Request Syntax

```
message = queue.receive_messages(
    AttributeNames=[
        'Policy'|'VisibilityTimeout'|'MaximumMessageSize'|'MessageRetentionPeriod'|'ApproximateFirstReceiveTimestamp',
    ],
    MessageAttributeNames=[
        'string',
    ],
    MaxNumberOfMessages=123,
    VisibilityTimeout=123,
    WaitTimeSeconds=123
)
```

Parameters

- **AttributeNames** (*list*) – A list of attributes that need to be returned along with each message.

The following lists the names and descriptions of the attributes that can be returned:

- `All` - returns all values.
- `ApproximateFirstReceiveTimestamp` - returns the time when the message was first received from the queue (epoch time in milliseconds).
- `ApproximateReceiveCount` - returns the number of times a message has been received from the queue but not deleted.
- `SenderId` - returns the AWS account number (or the IP address, if anonymous access is allowed) of the sender.
- `SentTimestamp` - returns the time when the message was sent to the queue (epoch time in milliseconds).
- (*string*) –
- **MessageAttributeNames** (*list*) – The name of the message attribute, where *N* is the index. The message attribute name can contain the following characters: A-Z, a-z, 0-9, underscore (`_`), hyphen (`-`), and period (`.`). The name must not start or end with a period, and it should not have successive periods. The name is case sensitive and must be unique among all attribute names for the message. The name can be up to 256 characters long. The name cannot start with “AWS.” or “Amazon.” (or any variations in casing), because these prefixes are reserved

for use by Amazon Web Services.

When using `ReceiveMessage`, you can send a list of attribute names to receive, or you can return all of the attributes by specifying “All” or “.” in your request. You can also use “foo.” to return all message attributes starting with the “foo” prefix.

– (*string*) –

- **MaxNumberOfMessages** (*integer*) – The maximum number of messages to return. Amazon SQS never returns more messages than this value but may return fewer. Values can be from 1 to 10. Default is 1.

All of the messages are not necessarily returned.

- **VisibilityTimeout** (*integer*) – The duration (in seconds) that the received messages are hidden from subsequent retrieve requests after being retrieved by a `ReceiveMessage` request.
- **WaitTimeSeconds** (*integer*) – The duration (in seconds) for which the call will wait for a message to arrive in the queue before returning. If a message is available, the call will return sooner than `WaitTimeSeconds`.

Return type `list(sqs.Message)`

Returns A list of Message resources

reload()

Calls `sqs.Client.get_queue_attributes()` to update the attributes of the Queue resource

Request Syntax

```
queue.reload()
```

Returns None

remove_permission (***kwargs*)

Revokes any permissions in the queue policy that matches the specified `Label` parameter. Only the owner of the queue can remove permissions.

Request Syntax

```
response = queue.remove_permission(  
    Label='string'  
)
```

Parameters `Label` (*string*) – [REQUIRED]

The identification of the permission to remove. This is the label added with the `AddPermission` action.

Returns None

send_message (***kwargs*)

Delivers a message to the specified queue. With Amazon SQS, you now have the ability to send large payload messages that are up to 256KB (262,144 bytes) in size. To send large payloads, you must use an AWS SDK that supports SigV4 signing. To verify whether SigV4 is supported for an AWS SDK, check the SDK release notes.

Warning: The following list shows the characters (in Unicode) allowed in your message, according to the W3C XML specification. For more information, go to <http://www.w3.org/TR/REC-xml/#charsets>. If you send any characters not included in the list, your request will be rejected.

`#x9 | #xA | #xD | [#x20 to #xD7FF] | [#xE000 to #xFFFD] | [#x10000 to #x10FFFF]`

Request Syntax

```

response = queue.send_message(
    MessageBody='string',
    DelaySeconds=123,
    MessageAttributes={
        'string': {
            'StringValue': 'string',
            'BinaryValue': b'bytes',
            'StringListValues': [
                'string',
            ],
            'BinaryListValues': [
                b'bytes',
            ],
            'DataType': 'string'
        }
    }
)

```

Parameters

- **MessageBody** (*string*) – [REQUIRED]

The message to send. String maximum 256 KB in size. For a list of allowed characters, see the preceding important note.

- **DelaySeconds** (*integer*) – The number of seconds (0 to 900 - 15 minutes) to delay a specific message. Messages with a positive `DelaySeconds` value become available for processing after the delay time is finished. If you don't specify a value, the default value for the queue applies.
- **MessageAttributes** (*dict*) – Each message attribute consists of a Name, Type, and Value. For more information, see [Message Attribute Items](#).

– (*string*) –

* (*dict*) –

The user-specified message attribute value. For string data types, the value attribute has the same restrictions on the content as the message body. For more information, see [SendMessage](#).

Name, type, and value must not be empty or null. In addition, the message body should not be empty or null. All parts of the message attribute, including name, type, and value, are included in the message size restriction, which is currently 256 KB (262,144 bytes).

- **StringValue** (*string*) –

Strings are Unicode with UTF8 binary encoding. For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters.

- **BinaryValue** (*bytes*) –

Binary type attributes can store any binary data, for example, compressed data, encrypted data, or images.

- **StringListValues** (*list*) –

Not implemented. Reserved for future use.

- (*string*) –

- **BinaryListValues** (*list*) –

Not implemented. Reserved for future use.

- (*bytes*) –

- **DataType** (*string*) – [REQUIRED]

Amazon SQS supports the following logical data types: String, Number, and Binary. In addition, you can append your own custom labels. For more information, see [Message Attribute Data Types](#).

Return type dict

Returns

Response Syntax

```
{
  'MD5OfMessageBody': 'string',
  'MD5OfMessageAttributes': 'string',
  'MessageId': 'string'
}
```

Response Structure

- (*dict*) –

The MD5OfMessageBody and MessageId elements.

- **MD5OfMessageBody** (*string*) –

An MD5 digest of the non-URL-encoded message body string. This can be used to verify that Amazon SQS received the message correctly. Amazon SQS first URL decodes the message before creating the MD5 digest. For information about MD5, go to <http://www.faqs.org/rfcs/rfc1321.html>.

- **MD5OfMessageAttributes** (*string*) –

An MD5 digest of the non-URL-encoded message attribute string. This can be used to verify that Amazon SQS received the message correctly. Amazon SQS first URL decodes the message before creating the MD5 digest. For information about MD5, go to <http://www.faqs.org/rfcs/rfc1321.html>.

- **MessageId** (*string*) –

An element containing the message ID of the message sent to the queue. For more information, see [Queue and Message Identifiers](#) in the *Amazon SQS Developer Guide*.

send_messages (***kwargs*)

Delivers up to ten messages to the specified queue. This is a batch version of `SendMessage`. The result of the send action on each message is reported individually in the response. The maximum allowed individual message size is 256 KB (262,144 bytes).

The maximum total payload size (i.e., the sum of all a batch's individual message lengths) is also 256 KB (262,144 bytes).

If the `DelaySeconds` parameter is not specified for an entry, the default for the queue is used.

Warning: The following list shows the characters (in Unicode) that are allowed in your message, according to the W3C XML specification. For more information, go to <http://www.faqs.org/rfcs/rfc1321.html>. If you send any characters that are not included in the list, your request will be rejected.

#x9 | #xA | #xD | [#x20 to #xD7FF] | [#xE000 to #xFFFD] | [#x10000 to #x10FFFF]

Warning: Because the batch request can result in a combination of successful and unsuccessful actions, you should check for batch errors even when the call returns an HTTP status code of 200.

Note: Some API actions take lists of parameters. These lists are specified using the `param.n` notation. Values of `n` are integers starting from 1. For example, a parameter list with two elements looks like this:

```
Attribute.1=this
```

```
Attribute.2=that
```

Request Syntax

```
response = queue.send_messages(
    Entries=[
        {
            'Id': 'string',
            'MessageBody': 'string',
            'DelaySeconds': 123,
            'MessageAttributes': {
                'string': {
                    'StringValue': 'string',
                    'BinaryValue': b'bytes',
                    'StringListValues': [
                        'string',
                    ],
                    'BinaryListValues': [
                        b'bytes',
                    ],
                    'DataType': 'string'
                }
            }
        },
    ]
)
```

Parameters **Entries** (*list*) – [REQUIRED]

A list of `SendMessageBatchRequestEntry` items.

- (*dict*) –

Contains the details of a single Amazon SQS message along with a `Id`.

– **Id** (*string*) – [REQUIRED]

An identifier for the message in this batch. This is used to communicate the result. Note that the `Id`s of a batch request need to be unique within the request.

– **MessageBody** (*string*) – [REQUIRED]

Body of the message.

– **DelaySeconds** (*integer*) –

The number of seconds for which the message has to be delayed.

– **MessageAttributes** (*dict*) –

Each message attribute consists of a Name, Type, and Value. For more information, see [Message Attribute Items](#).

* (*string*) –

· (*dict*) –

The user-specified message attribute value. For string data types, the value attribute has the same restrictions on the content as the message body. For more information, see [SendMessage](#).

Name, type, and value must not be empty or null. In addition, the message body should not be empty or null. All parts of the message attribute, including name, type, and value, are included in the message size restriction, which is currently 256 KB (262,144 bytes).

- **StringValue** (*string*) –

Strings are Unicode with UTF8 binary encoding. For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters.

- **BinaryValue** (*bytes*) –

Binary type attributes can store any binary data, for example, compressed data, encrypted data, or images.

- **StringListValues** (*list*) –

Not implemented. Reserved for future use.

- (*string*) –

- **BinaryListValues** (*list*) –

Not implemented. Reserved for future use.

- (*bytes*) –

- **DataType** (*string*) – [REQUIRED]

Amazon SQS supports the following logical data types: String, Number, and Binary. In addition, you can append your own custom labels. For more information, see [Message Attribute Data Types](#).

Return type dict

Returns

Response Syntax

```
{
  'Successful': [
    {
      'Id': 'string',
      'MessageId': 'string',
      'MD5OfMessageBody': 'string',
      'MD5OfMessageAttributes': 'string'
    },
  ],
  'Failed': [
    {
      'Id': 'string',
      'SenderFault': True|False,
      'Code': 'string',
      'Message': 'string'
    },
  ],
}
```

Response Structure

- (*dict*) –

For each message in the batch, the response contains a `SendMessageBatchResultEntry` tag if the message succeeds or a `BatchResultErrorEntry` tag if the message fails.

- **Successful** (*list*) –

A list of `SendMessageBatchResultEntry` items.

- * (*dict*) –

Encloses a message ID for successfully enqueued message of a `SendMessageBatch`.

- **Id** (*string*) –

An identifier for the message in this batch.

- **MessageId** (*string*) –

An identifier for the message.

- **MD5OfMessageBody** (*string*) –

An MD5 digest of the non-URL-encoded message body string. This can be used to verify that Amazon SQS received the message correctly. Amazon SQS first URL decodes the message before creating the MD5 digest. For information about MD5, go to <http://www.faqs.org/rfcs/rfc1321.html>.

- **MD5OfMessageAttributes** (*string*) –

An MD5 digest of the non-URL-encoded message attribute string. This can be used to verify that Amazon SQS received the message batch correctly. Amazon SQS first URL decodes the message before creating the MD5 digest. For information about MD5, go to <http://www.faqs.org/rfcs/rfc1321.html>.

- **Failed** (*list*) –

A list of `BatchResultErrorEntry` items with the error detail about each message that could not be enqueued.

- * (*dict*) –

This is used in the responses of batch API to give a detailed description of the result of an action on each entry in the request.

- **Id** (*string*) –

The id of an entry in a batch request.

- **SenderFault** (*boolean*) –

Whether the error happened due to the sender's fault.

- **Code** (*string*) –

An error code representing why the action failed on this entry.

- **Message** (*string*) –

A message explaining why the action failed on this entry.

set_attributes (***kwargs*)

Sets the value of one or more queue attributes. When you change a queue's attributes, the change can take up to 60 seconds for most of the attributes to propagate throughout the SQS system. Changes made to the `MessageRetentionPeriod` attribute can take up to 15 minutes.

Note: Going forward, new attributes might be added. If you are writing code that calls this action, we recommend that you structure your code so that it can handle new attributes gracefully.

Request Syntax

```
response = queue.set_attributes(  
    Attributes={  
        'string': 'string'  
    }  
)
```

Parameters `Attributes` (*dict*) – [REQUIRED]

A map of attributes to set.

The following lists the names, descriptions, and values of the special request parameters the `SetQueueAttributes` action uses:

- `DelaySeconds` - The time in seconds that the delivery of all messages in the queue will be delayed. An integer from 0 to 900 (15 minutes). The default for this attribute is 0 (zero).
- `MaximumMessageSize` - The limit of how many bytes a message can contain before Amazon SQS rejects it. An integer from 1024 bytes (1 KiB) up to 262144 bytes (256 KiB). The default for this attribute is 262144 (256 KiB).
- `MessageRetentionPeriod` - The number of seconds Amazon SQS retains a message. Integer representing seconds, from 60 (1 minute) to 1209600 (14 days). The default for this attribute is 345600 (4 days).
- `Policy` - The queue's policy. A valid AWS policy. For more information about policy structure, see [Overview of AWS IAM Policies](#) in the *Amazon IAM User Guide*.
- `ReceiveMessageWaitTimeSeconds` - The time for which a `ReceiveMessage` call will wait for a message to arrive. An integer from 0 to 20 (seconds). The default for this attribute is 0.
- `VisibilityTimeout` - The visibility timeout for the queue. An integer from 0 to 43200 (12 hours). The default for this attribute is 30. For more information about visibility timeout, see [Visibility Timeout](#) in the *Amazon SQS Developer Guide*.
- `RedrivePolicy` - The parameters for dead letter queue functionality of the source queue. For more information about `RedrivePolicy` and dead letter queues, see [Using Amazon SQS Dead Letter Queues](#) in the *Amazon SQS Developer Guide*.
- (*string*) –
The name of a queue attribute.
– (*string*) –

The value of a queue attribute.

Returns `None`

Sub-resources

Sub-resources are methods that create a new instance of a child resource. This resource's identifiers get passed along to the child. For more information about sub-resources refer to the [Resources Introduction Guide](#).

Message (*receipt_handle*)

Creates a Message resource.:

```
message = queue.Message('receipt_handle')
```

Parameters `receipt_handle` (*string*) – The Message's `receipt_handle` identifier. This **must** be set.

Return type `SQS.Message`

Returns A Message resource

Collections

Collections provide an interface to iterate over and manipulate groups of resources. For more information about collections refer to the [Resources Introduction Guide](#).

`dead_letter_source_queues`

`all()`

Creates an iterable of all Queue resources in the collection.

Request Syntax

```
queue_iterator = queue.dead_letter_source_queues.all()
```

Return type `list(sqs.Queue)`

Returns A list of Queue resources

`filter()`

Creates an iterable of all Queue resources in the collection filtered by kwargs passed to method.

Request Syntax

```
queue_iterator = queue.dead_letter_source_queues.filter()
```

Return type `list(sqs.Queue)`

Returns A list of Queue resources

`limit(**kwargs)`

Creates an iterable up to a specified amount of Queue resources in the collection.

Request Syntax

```
queue_iterator = queue.dead_letter_source_queues.limit(
    count=123
)
```

Parameters `count (integer)` – The limit to the number of resources in the iterable.

Return type `list(sqs.Queue)`

Returns A list of Queue resources

`page_size(**kwargs)`

Creates an iterable of all Queue resources in the collection, but limits the number of items returned by each service call by the specified amount.

Request Syntax

```
queue_iterator = queue.dead_letter_source_queues.page_size(
    count=123
)
```

Parameters `count (integer)` – The number of items returned by each service call

Return type `list(sqs.Queue)`

Returns A list of Queue resources

SSM

Table of Contents

- [SSM](#)
 - [Client](#)

Client

class SSM.Client

A low-level client representing Amazon Simple Systems Management Service (SSM):

```
import boto3
```

```
client = boto3.client('ssm')
```

These are the available methods:

- `can_paginate()`
- `create_association()`
- `create_association_batch()`
- `create_document()`
- `delete_association()`
- `delete_document()`
- `describe_association()`
- `describe_document()`
- `generate_presigned_url()`
- `get_document()`
- `get_paginator()`
- `get_waiter()`
- `list_associations()`
- `list_documents()`
- `update_association_status()`

`can_paginate(operation_name)`

Check if an operation can be paginated.

Parameters `operation_name` (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

`create_association(**kwargs)`

Associates the specified configuration document with the specified instance.

When you associate a configuration document with an instance, the configuration agent on the instance processes the configuration document and configures the instance as specified.

If you associate a configuration document with an instance that already has an associated configuration document, we replace the current configuration document with the new configuration document.

Request Syntax

```
response = client.create_association(
    Name='string',
    InstanceId='string'
)
```

Parameters

- **Name** (*string*) – [REQUIRED]
The name of the configuration document.
- **InstanceId** (*string*) – [REQUIRED]
The ID of the instance.

Return type dict

Returns

Response Syntax

```
{
  'AssociationDescription': {
    'Name': 'string',
    'InstanceId': 'string',
    'Date': datetime(2015, 1, 1),
    'Status': {
      'Date': datetime(2015, 1, 1),
      'Name': 'Pending'|'Success'|'Failed',
      'Message': 'string',
      'AdditionalInfo': 'string'
    }
  }
}
```

Response Structure

- (*dict*) –
 - **AssociationDescription** (*dict*) –
Information about the association.
 - * **Name** (*string*) –
The name of the configuration document.
 - * **InstanceId** (*string*) –
The ID of the instance.
 - * **Date** (*datetime*) –
The date when the association was made.
 - * **Status** (*dict*) –
The association status.
 - **Date** (*datetime*) –
The date when the status changed.
 - **Name** (*string*) –
The status.
 - **Message** (*string*) –
The reason for the status.
 - **AdditionalInfo** (*string*) –
A user-defined string.

create_association_batch (***kwargs*)

Associates the specified configuration documents with the specified instances.

When you associate a configuration document with an instance, the configuration agent on the instance processes the configuration document and configures the instance as specified.

If you associate a configuration document with an instance that already has an associated configuration document, we replace the current configuration document with the new configuration document.

Request Syntax

```
response = client.create_association_batch(
    Entries=[
        {
            'Name': 'string',
            'InstanceId': 'string'
        },
    ]
)
```

Parameters *Entries (list)* – **[REQUIRED]**

One or more associations.

- (*dict*) –

Describes the association of a configuration document and an instance.

- **Name** (*string*) –

The name of the configuration document.

- **InstanceId** (*string*) –

The ID of the instance.

Return type dict

Returns**Response Syntax**

```
{
    'Successful': [
        {
            'Name': 'string',
            'InstanceId': 'string',
            'Date': datetime(2015, 1, 1),
            'Status': {
                'Date': datetime(2015, 1, 1),
                'Name': 'Pending'|'Success'|'Failed',
                'Message': 'string',
                'AdditionalInfo': 'string'
            }
        },
    ],
    'Failed': [
        {
            'Entry': {
                'Name': 'string',
                'InstanceId': 'string'
            },
            'Message': 'string',
            'Fault': 'Client'|'Server'|'Unknown'
        },
    ],
}
```

```
]
}
```

Response Structure

- *(dict)* –

- **Successful** (*list*) –

Information about the associations that succeeded.

- * *(dict)* –

Describes an association.

- **Name** (*string*) –

The name of the configuration document.

- **InstanceId** (*string*) –

The ID of the instance.

- **Date** (*datetime*) –

The date when the association was made.

- **Status** (*dict*) –

The association status.

- **Date** (*datetime*) –

The date when the status changed.

- **Name** (*string*) –

The status.

- **Message** (*string*) –

The reason for the status.

- **AdditionalInfo** (*string*) –

A user-defined string.

- **Failed** (*list*) –

Information about the associations that failed.

- * *(dict)* –

Describes a failed association.

- **Entry** (*dict*) –

The association.

- **Name** (*string*) –

The name of the configuration document.

- **InstanceId** (*string*) –

The ID of the instance.

- **Message** (*string*) –

A description of the failure.

- **Fault** (*string*) –

The source of the failure.

create_document (***kwargs*)

Creates a configuration document.

After you create a configuration document, you can use `CreateAssociation` to associate it with one or more running instances.

Request Syntax

```
response = client.create_document(
    Content='string',
    Name='string'
)
```

Parameters

- **Content** (*string*) – [REQUIRED]

A valid JSON file. For more information about the contents of this file, see [Configuration Document](#).

- **Name** (*string*) – [REQUIRED]

A name for the configuration document.

Return type dict

Returns**Response Syntax**

```
{
    'DocumentDescription': {
        'Shal': 'string',
        'Name': 'string',
        'CreateDate': datetime(2015, 1, 1),
        'Status': 'Creating'|'Active'|'Deleting'
    }
}
```

Response Structure

- (*dict*) –
 - **DocumentDescription** (*dict*) –

Information about the configuration document.

- * **Shal** (*string*) –

The SHA1 hash of the document, which you can use for verification purposes.

- * **Name** (*string*) –

The name of the configuration document.

- * **CreateDate** (*datetime*) –

The date when the configuration document was created.

- * **Status** (*string*) –

The status of the configuration document.

delete_association (***kwargs*)

Disassociates the specified configuration document from the specified instance.

When you disassociate a configuration document from an instance, it does not change the configuration of the instance. To change the configuration state of an instance after you disassociate a configuration document, you must create a new configuration document with the desired configuration and associate it with the instance.

Request Syntax

```
response = client.delete_association(
    Name='string',
    InstanceId='string'
)
```


Parameters

- **Name** (*string*) – [REQUIRED]
The name of the configuration document.
- **InstanceId** (*string*) – [REQUIRED]
The ID of the instance.

Return type dict**Returns****Response Syntax**

{ }

Response Structure

- (*dict*) –

delete_document (***kwargs*)

Deletes the specified configuration document.

You must use DeleteAssociation to disassociate all instances that are associated with the configuration document before you can delete it.

Request Syntax

```
response = client.delete_document(
    Name='string'
)
```

Parameters **Name** (*string*) – [REQUIRED]

The name of the configuration document.

Return type dict**Returns****Response Syntax**

{ }

Response Structure

- (*dict*) –

describe_association (***kwargs*)

Describes the associations for the specified configuration document or instance.

Request Syntax

```
response = client.describe_association(
    Name='string',
    InstanceId='string'
)
```

Parameters

- **Name** (*string*) – [REQUIRED]
The name of the configuration document.
- **InstanceId** (*string*) – [REQUIRED]
The ID of the instance.

Return type dict

Returns

Response Syntax

```
{
  'AssociationDescription': {
    'Name': 'string',
    'InstanceId': 'string',
    'Date': datetime(2015, 1, 1),
    'Status': {
      'Date': datetime(2015, 1, 1),
      'Name': 'Pending'|'Success'|'Failed',
      'Message': 'string',
      'AdditionalInfo': 'string'
    }
  }
}
```

Response Structure

- (*dict*) –
 - **AssociationDescription** (*dict*) –
Information about the association.
 - * **Name** (*string*) –
The name of the configuration document.
 - * **InstanceId** (*string*) –
The ID of the instance.
 - * **Date** (*datetime*) –
The date when the association was made.
 - * **Status** (*dict*) –
The association status.
 - **Date** (*datetime*) –
The date when the status changed.
 - **Name** (*string*) –
The status.
 - **Message** (*string*) –
The reason for the status.
 - **AdditionalInfo** (*string*) –
A user-defined string.

describe_document (***kwargs*)

Describes the specified configuration document.

Request Syntax

```
response = client.describe_document(
    Name='string'
)
```

Parameters **Name** (*string*) – [REQUIRED]

The name of the configuration document.

Return type dict

Returns**Response Syntax**

```
{
    'Document': {
        'Sha1': 'string',
        'Name': 'string',
        'CreateDate': datetime(2015, 1, 1),
        'Status': 'Creating'|'Active'|'Deleting'
    }
}
```

Response Structure

- *(dict)* –

- **Document** (*dict*) –

Information about the configuration document.

- * **Sha1** (*string*) –

The SHA1 hash of the document, which you can use for verification purposes.

- * **Name** (*string*) –

The name of the configuration document.

- * **CreateDate** (*datetime*) –

The date when the configuration document was created.

- * **Status** (*string*) –

The status of the configuration document.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to *ClientMethod*.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_document (***kwargs*)

Gets the contents of the specified configuration document.

Request Syntax

```
response = client.get_document(
    Name='string'
)
```

Parameters **Name** (*string*) – **[REQUIRED]**

The name of the configuration document.

Return type dict

Returns**Response Syntax**

```
{
    'Name': 'string',
    'Content': 'string'
}
```

Response Structure

- (*dict*) –
 - **Name** (*string*) –
The name of the configuration document.
 - **Content** (*string*) –
The contents of the configuration document.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_waiter (*waiter_name*)**list_associations** (***kwargs*)

Lists the associations for the specified configuration document or instance.

Request Syntax

```
response = client.list_associations(
    AssociationFilterList=[
        {
            'key': 'InstanceId|Name',
            'value': 'string'
        },
    ],
    MaxResults=123,
    NextToken='string'
)
```

Parameters

- **AssociationFilterList** (*list*) – **[REQUIRED]**
One or more filters. Use a filter to return a more specific list of results.
 - (*dict*) –
Describes a filter.
 - * **key** (*string*) – **[REQUIRED]**
The name of the filter.
 - * **value** (*string*) – **[REQUIRED]**
The filter value.

- **MaxResults** (*integer*) – The maximum number of items to return for this call. The call also returns a token that you can specify in a subsequent call to get the next set of results.
- **NextToken** (*string*) – The token for the next set of items to return. (You received this token from a previous call.)

Return type dict

Returns

Response Syntax

```
{
  'Associations': [
    {
      'Name': 'string',
      'InstanceId': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **Associations** (*list*) –

The associations.

 - * (*dict*) –

Describes an association of a configuration document and an instance.

 - **Name** (*string*) –

The name of the configuration document.
 - **InstanceId** (*string*) –

The ID of the instance.
 - **NextToken** (*string*) –

The token to use when requesting the next set of items. If there are no additional items to return, the string is empty.

list_documents (***kwargs*)

Describes one or more of your configuration documents.

Request Syntax

```
response = client.list_documents(
    DocumentFilterList=[
        {
            'key': 'Name',
            'value': 'string'
        },
    ],
    MaxResults=123,
    NextToken='string'
)
```

Parameters

- **DocumentFilterList** (*list*) – One or more filters. Use a filter to return a more specific list of results.

– (*dict*) –

Describes a filter.

* **key** (*string*) – [REQUIRED]

The name of the filter.

* **value** (*string*) – [REQUIRED]

The value of the filter.

- **MaxResults** (*integer*) – The maximum number of items to return for this call. The call also returns a token that you can specify in a subsequent call to get the next set of results.
- **NextToken** (*string*) – The token for the next set of items to return. (You received this token from a previous call.)

Return type dict

Returns

Response Syntax

```
{
    'DocumentIdentifiers': [
        {
            'Name': 'string'
        },
    ],
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **DocumentIdentifiers** (*list*) –

The names of the configuration documents.

* (*dict*) –

Describes the name of a configuration document.

 - **Name** (*string*) –

The name of the configuration document.
 - **NextToken** (*string*) –

The token to use when requesting the next set of items. If there are no additional items to return, the string is empty.

update_association_status (**kwargs)

Updates the status of the configuration document associated with the specified instance.

Request Syntax

```
response = client.update_association_status(
    Name='string',
    InstanceId='string',
    AssociationStatus={
        'Date': datetime(2015, 1, 1),
        'Name': 'Pending'|'Success'|'Failed',
        'Message': 'string',
        'AdditionalInfo': 'string'
    }
)
```

Parameters

- **Name** (*string*) – [REQUIRED]

The name of the configuration document.

- **InstanceId** (*string*) – [REQUIRED]

The ID of the instance.

- **AssociationStatus** (*dict*) – [REQUIRED]

The association status.

- **Date** (*datetime*) – [REQUIRED]

The date when the status changed.

- **Name** (*string*) – [REQUIRED]

The status.

- **Message** (*string*) – [REQUIRED]

The reason for the status.

- **AdditionalInfo** (*string*) –

A user-defined string.

Return type dict

Returns

Response Syntax

```
{
  'AssociationDescription': {
    'Name': 'string',
    'InstanceId': 'string',
    'Date': datetime(2015, 1, 1),
    'Status': {
      'Date': datetime(2015, 1, 1),
      'Name': 'Pending'|'Success'|'Failed',
      'Message': 'string',
      'AdditionalInfo': 'string'
    }
  }
}
```

Response Structure

- (*dict*) –

- **AssociationDescription** (*dict*) –

Information about the association.

- * **Name** (*string*) –

The name of the configuration document.

- * **InstanceId** (*string*) –

The ID of the instance.

- * **Date** (*datetime*) –

The date when the association was made.

- * **Status** (*dict*) –

The association status.

- **Date** (*datetime*) –

The date when the status changed.

- **Name** (*string*) –

The status.

- **Message** (*string*) –
The reason for the status.
- **AdditionalInfo** (*string*) –
A user-defined string.

StorageGateway

Table of Contents

- StorageGateway
 - Client
 - Paginators

Client

class StorageGateway.**Client**

A low-level client representing AWS Storage Gateway:

```
import boto3
```

```
client = boto3.client('storagegateway')
```

These are the available methods:

- `activate_gateway()`
- `add_cache()`
- `add_tags_to_resource()`
- `add_upload_buffer()`
- `add_working_storage()`
- `can_paginate()`
- `cancel_archival()`
- `cancel_retrieval()`
- `create_cached_iscsi_volume()`
- `create_snapshot()`
- `create_snapshot_from_volume_recovery_point()`
- `create_stored_iscsi_volume()`
- `create_tapes()`
- `delete_bandwidth_rate_limit()`
- `delete_chap_credentials()`
- `delete_gateway()`
- `delete_snapshot_schedule()`
- `delete_tape()`
- `delete_tape_archive()`
- `delete_volume()`
- `describe_bandwidth_rate_limit()`
- `describe_cache()`
- `describe_cached_iscsi_volumes()`
- `describe_chap_credentials()`
- `describe_gateway_information()`
- `describe_maintenance_start_time()`
- `describe_snapshot_schedule()`

- describe_stored_iscsi_volumes()
- describe_tape_archives()
- describe_tape_recovery_points()
- describe_tapes()
- describe_upload_buffer()
- describe_vtl_devices()
- describe_working_storage()
- disable_gateway()
- generate_presigned_url()
- get_paginator()
- get_waiter()
- list_gateways()
- list_local_disks()
- list_tags_for_resource()
- list_volume_initiators()
- list_volume_recovery_points()
- list_volumes()
- remove_tags_from_resource()
- reset_cache()
- retrieve_tape_archive()
- retrieve_tape_recovery_point()
- shutdown_gateway()
- start_gateway()
- update_bandwidth_rate_limit()
- update_chap_credentials()
- update_gateway_information()
- update_gateway_software_now()
- update_maintenance_start_time()
- update_snapshot_schedule()
- update_vtl_device_type()

activate_gateway (**kwargs)

This operation activates the gateway you previously deployed on your host. For more information, see [Activate the AWS Storage Gateway](#) . In the activation process, you specify information such as the region you want to use for storing snapshots, the time zone for scheduled snapshots the gateway snapshot schedule window, an activation key, and a name for your gateway. The activation process also associates your gateway with your account; for more information, see [UpdateGatewayInformation](#) .

Note: You must turn on the gateway VM before you can activate your gateway.

Request Syntax

```
response = client.activate_gateway(
    ActivationKey='string',
    GatewayName='string',
    GatewayTimezone='string',
    GatewayRegion='string',
    GatewayType='string',
    TapeDriveType='string',
    MediumChangerType='string'
)
```

Parameters

- **ActivationKey** (*string*) – [REQUIRED]

Your gateway activation key. You can obtain the activation key by sending an

HTTP GET request with redirects enabled to the gateway IP address (port 80). The redirect URL returned in the response provides you the activation key for your gateway in the query string parameter `activationKey`. It may also include other activation-related parameters, however, these are merely defaults – the arguments you pass to the `ActivateGateway` API call determine the actual configuration of your gateway.

- **GatewayName** (*string*) – [REQUIRED]

A unique identifier for your gateway. This name becomes part of the gateway Amazon Resources Name (ARN) which is what you use as an input to other operations.

- **GatewayTimezone** (*string*) – [REQUIRED]

One of the values that indicates the time zone you want to set for the gateway. The time zone is used, for example, for scheduling snapshots and your gateway's maintenance schedule.

- **GatewayRegion** (*string*) – [REQUIRED]

One of the values that indicates the region where you want to store the snapshot backups. The gateway region specified must be the same region as the region in your `Host` header in the request. For more information about available regions and endpoints for AWS Storage Gateway, see [Regions and Endpoints](#) in the *Amazon Web Services Glossary*.

Valid Values : “us-east-1”, “us-west-1”, “us-west-2”, “eu-west-1”, “eu-central-1”, “ap-northeast-1”, “ap-southeast-1”, “ap-southeast-2”, “sa-east-1”

- **GatewayType** (*string*) – One of the values that defines the type of gateway to activate. The type specified is critical to all later functions of the gateway and cannot be changed after activation. The default value is `STORED`.
- **TapeDriveType** (*string*) – The value that indicates the type of tape drive to use for gateway-VTL. This field is optional.

Valid Values : “IBM-ULT3580-TD5”

- **MediumChangerType** (*string*) – The value that indicates the type of medium changer to use for gateway-VTL. This field is optional.

Valid Values : “STK-L700”, “AWS-Gateway-VTL”

Return type dict

Returns

Response Syntax

```
{
    'GatewayARN': 'string'
}
```

Response Structure

- (*dict*) –

AWS Storage Gateway returns the Amazon Resource Name (ARN) of the activated gateway. It is a string made of information such as your account, gateway name, and region. This ARN is used to reference the gateway in other API operations as well as resource-based authorization.

- **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the `ListGateways` operation to return a list of gateways for your account and region.

add_cache (***kwargs*)

This operation configures one or more gateway local disks as cache for a cached-volume gateway. This operation is supported only for the gateway-cached volume architecture (see [Storage Gateway Concepts](#)).

In the request, you specify the gateway Amazon Resource Name (ARN) to which you want to add cache, and one or more disk IDs that you want to configure as cache.

Request Syntax

```
response = client.add_cache(
    GatewayARN='string',
    DiskIds=[
        'string',
    ]
)
```

Parameters

- **GatewayARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **DiskIds** (*list*) – **[REQUIRED]**
 - (*string*) –

Return type dict

Returns**Response Syntax**

```
{
    'GatewayARN': 'string'
}
```

Response Structure

- (*dict*) –
 - **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

add_tags_to_resource (***kwargs*)

This operation adds one or more tags to the specified resource. You use tags to add metadata to resources, which you can use to categorize these resources. For example, you can categorize resources by purpose, owner, environment, or team. Each tag consists of a key and a value, which you define. You can add tags to the following AWS Storage Gateway resources:

- Storage gateways of all types
- Storage Volumes
- Virtual Tapes

You can create a maximum of 10 tags for each resource. Virtual tapes and storage volumes that are recovered to a new gateway maintain their tags.

Request Syntax

```
response = client.add_tags_to_resource(
    ResourceARN='string',
    Tags=[
        {
            'Key': 'string',
```

```
        'Value': 'string'
    },
]
)
```

Parameters

- **ResourceARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the resource you want to add tags to.

- **Tags** (*list*) – [REQUIRED]

The key-value pair that represents the tag you want to add to the resource. The value can be an empty string.

Note: Valid characters for key and value are letters, spaces, and numbers representable in UTF-8 format, and the following special characters: + - = . _ : / @.

– (*dict*) –

- * **Key** (*string*) – [REQUIRED]
- * **Value** (*string*) – [REQUIRED]

Return type dict

Returns

Response Syntax

```
{
    'ResourceARN': 'string'
}
```

Response Structure

- (*dict*) –

AddTagsToResourceOutput

- **ResourceARN** (*string*) –

The Amazon Resource Name (ARN) of the resource you want to add tags to.

add_upload_buffer (***kwargs*)

This operation configures one or more gateway local disks as upload buffer for a specified gateway. This operation is supported for both the gateway-stored and gateway-cached volume architectures.

In the request, you specify the gateway Amazon Resource Name (ARN) to which you want to add upload buffer, and one or more disk IDs that you want to configure as upload buffer.

Request Syntax

```
response = client.add_upload_buffer(
    GatewayARN='string',
    DiskIds=[
        'string',
    ]
)
```

Parameters

- **GatewayARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **DiskIds** (*list*) – [REQUIRED]
 - (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'GatewayARN': 'string'
}
```

Response Structure

- (*dict*) –
 - **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

add_working_storage (***kwargs*)

This operation configures one or more gateway local disks as working storage for a gateway. This operation is supported only for the gateway-stored volume architecture. This operation is deprecated method in cached-volumes API version (20120630). Use AddUploadBuffer instead.

Note: Working storage is also referred to as upload buffer. You can also use the AddUploadBuffer operation to add upload buffer to a stored-volume gateway.

In the request, you specify the gateway Amazon Resource Name (ARN) to which you want to add working storage, and one or more disk IDs that you want to configure as working storage.

Request Syntax

```
response = client.add_working_storage(
    GatewayARN='string',
    DiskIds=[
        'string',
    ]
)
```

Parameters

- **GatewayARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **DiskIds** (*list*) – [REQUIRED]

An array of strings that identify disks that are to be configured as working storage. Each string have a minimum length of 1 and maximum length of 300. You can get the disk IDs from the ListLocalDisks API.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'GatewayARN': 'string'
}
```

Response Structure

- (*dict*) –

A JSON object containing the of the gateway for which working storage was configured.

- **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

cancel_archival (***kwargs*)

Cancels archiving of a virtual tape to the virtual tape shelf (VTS) after the archiving process is initiated.

Request Syntax

```
response = client.cancel_archival(
    GatewayARN='string',
    TapeARN='string'
)
```

Parameters

- **GatewayARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **TapeARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the virtual tape you want to cancel archiving for.

Return type dict

Returns**Response Syntax**

```
{
    'TapeARN': 'string'
}
```

Response Structure

- (*dict*) –

CancelArchivalOutput

- **TapeARN** (*string*) –

The Amazon Resource Name (ARN) of the virtual tape for which archiving was canceled.

cancel_retrieval (***kwargs*)

Cancels retrieval of a virtual tape from the virtual tape shelf (VTS) to a gateway after the retrieval process is initiated. The virtual tape is returned to the VTS.

Request Syntax

```
response = client.cancel_retrieval(
    GatewayARN='string',
    TapeARN='string'
)
```

Parameters

- **GatewayARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **TapeARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the virtual tape you want to cancel retrieval for.

Return type dict

Returns**Response Syntax**

```
{
    'TapeARN': 'string'
}
```

Response Structure

- (*dict*) –

CancelRetrievalOutput

- **TapeARN** (*string*) –

The Amazon Resource Name (ARN) of the virtual tape for which retrieval was canceled.

create_cached_iscsi_volume (***kwargs*)

This operation creates a cached volume on a specified cached gateway. This operation is supported only for the gateway-cached volume architecture.

Note: Cache storage must be allocated to the gateway before you can create a cached volume. Use the AddCache operation to add cache storage to a gateway.

In the request, you must specify the gateway, size of the volume in bytes, the iSCSI target name, an IP address on which to expose the target, and a unique client token. In response, AWS Storage Gateway creates the volume and returns information about it such as the volume Amazon Resource Name (ARN), its size, and the iSCSI target ARN that initiators can use to connect to the volume target.

Request Syntax

```
response = client.create_cached_iscsi_volume(
    GatewayARN='string',
    VolumeSizeInBytes=123,
    SnapshotId='string',
    TargetName='string',
    NetworkInterfaceId='string',
```

```
ClientToken='string'  
)
```

Parameters

- **GatewayARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **VolumeSizeInBytes** (*integer*) – [REQUIRED]
- **SnapshotId** (*string*) –
- **TargetName** (*string*) – [REQUIRED]
- **NetworkInterfaceId** (*string*) – [REQUIRED]
- **ClientToken** (*string*) – [REQUIRED]

Return type dict

Returns**Response Syntax**

```
{  
    'VolumeARN': 'string',  
    'TargetARN': 'string'  
}
```

Response Structure

- (*dict*) –
 - **VolumeARN** (*string*) –
 - **TargetARN** (*string*) –

create_snapshot (***kwargs*)

This operation initiates a snapshot of a volume.

AWS Storage Gateway provides the ability to back up point-in-time snapshots of your data to Amazon Simple Storage (S3) for durable off-site recovery, as well as import the data to an Amazon Elastic Block Store (EBS) volume in Amazon Elastic Compute Cloud (EC2). You can take snapshots of your gateway volume on a scheduled or ad-hoc basis. This API enables you to take ad-hoc snapshot. For more information, see [Working With Snapshots in the AWS Storage Gateway Console](#).

In the CreateSnapshot request you identify the volume by providing its Amazon Resource Name (ARN). You must also provide description for the snapshot. When AWS Storage Gateway takes the snapshot of specified volume, the snapshot and description appears in the AWS Storage Gateway Console. In response, AWS Storage Gateway returns you a snapshot ID. You can use this snapshot ID to check the snapshot progress or later use it when you want to create a volume from a snapshot.

Note: To list or delete a snapshot, you must use the Amazon EC2 API. For more information, see DescribeSnapshots or DeleteSnapshot in the [EC2 API reference](#).

Request Syntax

```
response = client.create_snapshot(  
    VolumeARN='string',  
    SnapshotDescription='string'  
)
```

Parameters

- **VolumeARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the volume. Use the ListVolumes operation to return a list of gateway volumes.

- **SnapshotDescription** (*string*) – [REQUIRED]

Textual description of the snapshot that appears in the Amazon EC2 console, Elastic Block Store snapshots panel in the **Description** field, and in the AWS Storage Gateway snapshot **Details** pane, **Description** field

Return type dict

Returns

Response Syntax

```
{
  'VolumeARN': 'string',
  'SnapshotId': 'string'
}
```

Response Structure

- (*dict*) –

A JSON object containing the following fields:

- **VolumeARN** (*string*) –

The Amazon Resource Name (ARN) of the volume of which the snapshot was taken.

- **SnapshotId** (*string*) –

The snapshot ID that is used to refer to the snapshot in future operations such as describing snapshots (Amazon Elastic Compute Cloud API `DescribeSnapshots`) or creating a volume from a snapshot (`CreateStorediSCSIVolume`).

`create_snapshot_from_volume_recovery_point` (***kwargs*)

This operation initiates a snapshot of a gateway from a volume recovery point. This operation is supported only for the gateway-cached volume architecture (see).

A volume recovery point is a point in time at which all data of the volume is consistent and from which you can create a snapshot. To get a list of volume recovery point for gateway-cached volumes, use `ListVolumeRecoveryPoints`.

In the `CreateSnapshotFromVolumeRecoveryPoint` request, you identify the volume by providing its Amazon Resource Name (ARN). You must also provide a description for the snapshot. When AWS Storage Gateway takes a snapshot of the specified volume, the snapshot and its description appear in the AWS Storage Gateway console. In response, AWS Storage Gateway returns you a snapshot ID. You can use this snapshot ID to check the snapshot progress or later use it when you want to create a volume from a snapshot.

Note: To list or delete a snapshot, you must use the Amazon EC2 API. For more information, in *Amazon Elastic Compute Cloud API Reference*.

Request Syntax

```
response = client.create_snapshot_from_volume_recovery_point(
    VolumeARN='string',
    SnapshotDescription='string'
)
```

Parameters

- **VolumeARN** (*string*) – [REQUIRED]
- **SnapshotDescription** (*string*) – [REQUIRED]

Return type dict

Returns

Response Syntax

```
{
    'SnapshotId': 'string',
    'VolumeARN': 'string',
    'VolumeRecoveryPointTime': 'string'
}
```

Response Structure

- (*dict*) –
 - **SnapshotId** (*string*) –
 - **VolumeARN** (*string*) –
 - **VolumeRecoveryPointTime** (*string*) –

`create_stored_iscsi_volume` (**kwargs)

This operation creates a volume on a specified gateway. This operation is supported only for the gateway-stored volume architecture.

The size of the volume to create is inferred from the disk size. You can choose to preserve existing data on the disk, create volume from an existing snapshot, or create an empty volume. If you choose to create an empty gateway volume, then any existing data on the disk is erased.

In the request you must specify the gateway and the disk information on which you are creating the volume. In response, AWS Storage Gateway creates the volume and returns volume information such as the volume Amazon Resource Name (ARN), its size, and the iSCSI target ARN that initiators can use to connect to the volume target.

Request Syntax

```
response = client.create_stored_iscsi_volume(
    GatewayARN='string',
    DiskId='string',
    SnapshotId='string',
    PreserveExistingData=True|False,
    TargetName='string',
    NetworkInterfaceId='string'
)
```

Parameters

- **GatewayARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the gateway. Use the `ListGateways` operation to return a list of gateways for your account and region.

- **DiskId** (*string*) – **[REQUIRED]**

The unique identifier for the gateway local disk that is configured as a stored volume. Use `ListLocalDisks` to list disk IDs for a gateway.

- **SnapshotId** (*string*) – The snapshot ID (e.g. “snap-1122aabb”) of the snapshot to restore as the new stored volume. Specify this field if you want to create the iSCSI storage volume from a snapshot otherwise do not include this field. To list snapshots for your account use `DescribeSnapshots` in the *Amazon Elastic Compute Cloud API Reference*.

- **PreserveExistingData** (*boolean*) – **[REQUIRED]**

Specify this field as true if you want to preserve the data on the local disk. Otherwise, specifying this field as false creates an empty volume.

Valid Values : true, false

- **TargetName** (*string*) – [REQUIRED]

The name of the iSCSI target used by initiators to connect to the target and as a suffix for the target ARN. For example, specifying `TargetName` as `myvolume` results in the target ARN of `arn:aws:storagegateway:us-east-1:111122223333:gateway/mygateway/target/iqn.1997-05.com.amazon:myvolume`. The target name must be unique across all volumes of a gateway.

- **NetworkInterfaceId** (*string*) – [REQUIRED]

The network interface of the gateway on which to expose the iSCSI target. Only IPv4 addresses are accepted. Use `DescribeGatewayInformation` to get a list of the network interfaces available on a gateway.

Valid Values : A valid IP address.

Return type dict

Returns

Response Syntax

```
{
    'VolumeARN': 'string',
    'VolumeSizeInBytes': 123,
    'TargetARN': 'string'
}
```

Response Structure

- (*dict*) –

A JSON object containing the following fields:

- **VolumeARN** (*string*) –

The Amazon Resource Name (ARN) of the configured volume.

- **VolumeSizeInBytes** (*integer*) –

The size of the volume in bytes.

- **TargetARN** (*string*) –

The Amazon Resource Name (ARN) of the volume target that includes the iSCSI name that initiators can use to connect to the target.

create_tapes (***kwargs*)

Creates one or more virtual tapes. You write data to the virtual tapes and then archive the tapes.

Note: Cache storage must be allocated to the gateway before you can create virtual tapes. Use the `AddCache` operation to add cache storage to a gateway.

Request Syntax

```
response = client.create_tapes(
    GatewayARN='string',
    TapeSizeInBytes=123,
    ClientToken='string',
    NumTapesToCreate=123,
    TapeBarcodePrefix='string'
)
```

Parameters

- **GatewayARN** (*string*) – [REQUIRED]

The unique Amazon Resource Name(ARN) that represents the gateway to associate the virtual tapes with. Use the ListGateways operation to return a list of gateways for your account and region.

- **TapeSizeInBytes** (*integer*) – [REQUIRED]

The size, in bytes, of the virtual tapes you want to create.

Note: The size must be gigabyte (1024*1024*1024 byte) aligned.

- **ClientToken** (*string*) – [REQUIRED]

A unique identifier that you use to retry a request. If you retry a request, use the same ClientToken you specified in the initial request.

Note: Using the same ClientToken prevents creating the tape multiple times.

- **NumTapesToCreate** (*integer*) – [REQUIRED]

The number of virtual tapes you want to create.

- **TapeBarcodePrefix** (*string*) – [REQUIRED]

A prefix you append to the barcode of the virtual tape you are creating. This makes a barcode unique.

Note: The prefix must be 1 to 4 characters in length and must be upper-case letters A-Z.

Return type dict

Returns

Response Syntax

```
{
    'TapeARNs': [
        'string',
    ]
}
```

Response Structure

- (*dict*) –

CreateTapeOutput

- **TapeARNs** (*list*) –

A list of unique Amazon Resource Named (ARN) that represents the virtual tapes that were created.

* (*string*) –

delete_bandwidth_rate_limit (***kwargs*)

This operation deletes the bandwidth rate limits of a gateway. You can delete either the upload and download bandwidth rate limit, or you can delete both. If you delete only one of the limits, the other limit remains unchanged. To specify which gateway to work with, use the Amazon Resource Name (ARN) of the gateway in your request.

Request Syntax

```
response = client.delete_bandwidth_rate_limit(
    GatewayARN='string',
    BandwidthType='string'
)
```

Parameters

- **GatewayARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **BandwidthType** (*string*) – [REQUIRED]

Return type dict

Returns

Response Syntax

```
{
    'GatewayARN': 'string'
}
```

Response Structure

- (*dict*) –

A JSON object containing the of the gateway whose bandwidth rate information was deleted.

- **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

delete_chap_credentials (***kwargs*)

This operation deletes Challenge-Handshake Authentication Protocol (CHAP) credentials for a specified iSCSI target and initiator pair.

Request Syntax

```
response = client.delete_chap_credentials(
    TargetARN='string',
    InitiatorName='string'
)
```

Parameters

- **TargetARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the iSCSI volume target. Use the DescribeStorediSCSIVolumes operation to return to retrieve the TargetARN for specified VolumeARN.

- **InitiatorName** (*string*) – [REQUIRED]

The iSCSI initiator that connects to the target.

Return type dict

Returns

Response Syntax

```
{
    'TargetARN': 'string',
    'InitiatorName': 'string'
}
```

Response Structure

- (*dict*) –

A JSON object containing the following fields:

- **TargetARN** (*string*) –

The Amazon Resource Name (ARN) of the target.

- **InitiatorName** (*string*) –

The iSCSI initiator that connects to the target.

delete_gateway (***kwargs*)

This operation deletes a gateway. To specify which gateway to delete, use the Amazon Resource Name (ARN) of the gateway in your request. The operation deletes the gateway; however, it does not delete the gateway virtual machine (VM) from your host computer.

After you delete a gateway, you cannot reactivate it. Completed snapshots of the gateway volumes are not deleted upon deleting the gateway, however, pending snapshots will not complete. After you delete a gateway, your next step is to remove it from your environment.

Warning: You no longer pay software charges after the gateway is deleted; however, your existing Amazon EBS snapshots persist and you will continue to be billed for these snapshots. You can choose to remove all remaining Amazon EBS snapshots by canceling your Amazon EC2 subscription. If you prefer not to cancel your Amazon EC2 subscription, you can delete your snapshots using the Amazon EC2 console. For more information, see the [AWS Storage Gateway Detail Page](#) .

Request Syntax

```
response = client.delete_gateway(  
    GatewayARN='string'  
)
```

Parameters **GatewayARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

Return type dict

Returns**Response Syntax**

```
{  
    'GatewayARN': 'string'  
}
```

Response Structure

- (*dict*) –

A JSON object containing the id of the deleted gateway.

- **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

delete_snapshot_schedule (***kwargs*)

This operation deletes a snapshot of a volume.

You can take snapshots of your gateway volumes on a scheduled or ad-hoc basis. This API enables you to delete a snapshot schedule for a volume. For more information, see [Working with Snapshots](#) . In

the `DeleteSnapshotSchedule` request, you identify the volume by providing its Amazon Resource Name (ARN).

Note: To list or delete a snapshot, you must use the Amazon EC2 API. in *Amazon Elastic Compute Cloud API Reference*.

Request Syntax

```
response = client.delete_snapshot_schedule(
    VolumeARN='string'
)
```

Parameters `VolumeARN` (*string*) – [REQUIRED]

Return type dict

Returns

Response Syntax

```
{
    'VolumeARN': 'string'
}
```

Response Structure

- (*dict*) –
 - `VolumeARN` (*string*) –

delete_tape (***kwargs*)

Deletes the specified virtual tape.

Request Syntax

```
response = client.delete_tape(
    GatewayARN='string',
    TapeARN='string'
)
```

Parameters

- **GatewayARN** (*string*) – [REQUIRED]

The unique Amazon Resource Name (ARN) of the gateway that the virtual tape to delete is associated with. Use the `ListGateways` operation to return a list of gateways for your account and region.

- **TapeARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the virtual tape to delete.

Return type dict

Returns

Response Syntax

```
{
    'TapeARN': 'string'
}
```

Response Structure

- (*dict*) –
 - `DeleteTapeOutput`

– **TapeARN** (*string*) –

The Amazon Resource Name (ARN) of the deleted virtual tape.

delete_tape_archive (***kwargs*)

Deletes the specified virtual tape from the virtual tape shelf (VTS).

Request Syntax

```
response = client.delete_tape_archive(  
    TapeARN='string'  
)
```

Parameters **TapeARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the virtual tape to delete from the virtual tape shelf (VTS).

Return type dict

Returns

Response Syntax

```
{  
    'TapeARN': 'string'  
}
```

Response Structure

- (*dict*) –

DeleteTapeArchiveOutput

– **TapeARN** (*string*) –

The Amazon Resource Name (ARN) of the virtual tape that was deleted from the virtual tape shelf (VTS).

delete_volume (***kwargs*)

This operation deletes the specified gateway volume that you previously created using the CreateCachediSCSIVolume or CreateStorediSCSIVolume API. For gateway-stored volumes, the local disk that was configured as the storage volume is not deleted. You can reuse the local disk to create another storage volume.

Before you delete a gateway volume, make sure there are no iSCSI connections to the volume you are deleting. You should also make sure there is no snapshot in progress. You can use the Amazon Elastic Compute Cloud (Amazon EC2) API to query snapshots on the volume you are deleting and check the snapshot status. For more information, go to [DescribeSnapshots](#) in the *Amazon Elastic Compute Cloud API Reference*.

In the request, you must provide the Amazon Resource Name (ARN) of the storage volume you want to delete.

Request Syntax

```
response = client.delete_volume(  
    VolumeARN='string'  
)
```

Parameters **VolumeARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the volume. Use the ListVolumes operation to return a list of gateway volumes.

Return type dict

Returns**Response Syntax**

```
{
    'VolumeARN': 'string'
}
```

Response Structure

- (*dict*) –

A JSON object containing the of the storage volume that was deleted

- **VolumeARN** (*string*) –

The Amazon Resource Name (ARN) of the storage volume that was deleted. It is the same ARN you provided in the request.

describe_bandwidth_rate_limit (***kwargs*)

This operation returns the bandwidth rate limits of a gateway. By default, these limits are not set, which means no bandwidth rate limiting is in effect.

This operation only returns a value for a bandwidth rate limit only if the limit is set. If no limits are set for the gateway, then this operation returns only the gateway ARN in the response body. To specify which gateway to describe, use the Amazon Resource Name (ARN) of the gateway in your request.

Request Syntax

```
response = client.describe_bandwidth_rate_limit(
    GatewayARN='string'
)
```

Parameters **GatewayARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

Return type dict

Returns**Response Syntax**

```
{
    'GatewayARN': 'string',
    'AverageUploadRateLimitInBitsPerSec': 123,
    'AverageDownloadRateLimitInBitsPerSec': 123
}
```

Response Structure

- (*dict*) –

A JSON object containing the following fields:

- **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **AverageUploadRateLimitInBitsPerSec** (*integer*) –

The average upload bandwidth rate limit in bits per second. This field does not appear in the response if the upload rate limit is not set.

- **AverageDownloadRateLimitInBitsPerSec** (*integer*) –

The average download bandwidth rate limit in bits per second. This field does not appear in the response if the download rate limit is not set.

describe_cache (***kwargs*)

This operation returns information about the cache of a gateway. This operation is supported only for the gateway-cached volume architecture.

The response includes disk IDs that are configured as cache, and it includes the amount of cache allocated and used.

Request Syntax

```
response = client.describe_cache(  
    GatewayARN='string'  
)
```

Parameters **GatewayARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

Return type dict

Returns**Response Syntax**

```
{  
    'GatewayARN': 'string',  
    'DiskIds': [  
        'string',  
    ],  
    'CacheAllocatedInBytes': 123,  
    'CacheUsedPercentage': 123.0,  
    'CacheDirtyPercentage': 123.0,  
    'CacheHitPercentage': 123.0,  
    'CacheMissPercentage': 123.0  
}
```

Response Structure

- (*dict*) –
 - **GatewayARN** (*string*) –
The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.
 - **DiskIds** (*list*) –
 - * (*string*) –
 - **CacheAllocatedInBytes** (*integer*) –
 - **CacheUsedPercentage** (*float*) –
 - **CacheDirtyPercentage** (*float*) –
 - **CacheHitPercentage** (*float*) –
 - **CacheMissPercentage** (*float*) –

describe_cached_iscsi_volumes (***kwargs*)

This operation returns a description of the gateway volumes specified in the request. This operation is supported only for the gateway-cached volume architecture.

The list of gateway volumes in the request must be from one gateway. In the response Amazon Storage Gateway returns volume information sorted by volume Amazon Resource Name (ARN).

Request Syntax

```
response = client.describe_cached_iscsi_volumes(
    VolumeARNs=[
        'string',
    ]
)
```

Parameters **VolumeARNs** (*list*) – **[REQUIRED]**

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'CachediSCSIVolumes': [
        {
            'VolumeARN': 'string',
            'VolumeId': 'string',
            'VolumeType': 'string',
            'VolumeStatus': 'string',
            'VolumeSizeInBytes': 123,
            'VolumeProgress': 123.0,
            'SourceSnapshotId': 'string',
            'VolumeiSCSIAttributes': {
                'TargetARN': 'string',
                'NetworkInterfaceId': 'string',
                'NetworkInterfacePort': 123,
                'LunNumber': 123,
                'ChapEnabled': True|False
            }
        },
    ]
}
```

Response Structure

- (*dict*) –

A JSON object containing the following fields:

- **CachediSCSIVolumes** (*list*) –

An array of objects where each object contains metadata about one cached volume.

- * (*dict*) –

- **VolumeARN** (*string*) –
- **VolumeId** (*string*) –
- **VolumeType** (*string*) –
- **VolumeStatus** (*string*) –
- **VolumeSizeInBytes** (*integer*) –
- **VolumeProgress** (*float*) –
- **SourceSnapshotId** (*string*) –
- **VolumeiSCSIAttributes** (*dict*) –

Lists iSCSI information about a volume.

- **TargetARN** (*string*) –

The Amazon Resource Name (ARN) of the volume target.

- **NetworkInterfaceId** (*string*) –
The network interface identifier.
- **NetworkInterfacePort** (*integer*) –
The port used to communicate with iSCSI targets.
- **LunNumber** (*integer*) –
The logical disk number.
- **ChapEnabled** (*boolean*) –
Indicates whether mutual CHAP is enabled for the iSCSI target.

describe_chap_credentials (***kwargs*)

This operation returns an array of Challenge-Handshake Authentication Protocol (CHAP) credentials information for a specified iSCSI target, one for each target-initiator pair.

Request Syntax

```
response = client.describe_chap_credentials(  
    TargetARN='string'  
)
```

Parameters **TargetARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the iSCSI volume target. Use the DescribeStorediSCSIVolumes operation to return to retrieve the TargetARN for specified VolumeARN.

Return type dict

Returns

Response Syntax

```
{  
    'ChapCredentials': [  
        {  
            'TargetARN': 'string',  
            'SecretToAuthenticateInitiator': 'string',  
            'InitiatorName': 'string',  
            'SecretToAuthenticateTarget': 'string'  
        },  
    ],  
}
```

Response Structure

- (*dict*) –

A JSON object containing a .

- **ChapCredentials** (*list*) –

An array of ChapInfo objects that represent CHAP credentials. Each object in the array contains CHAP credential information for one target-initiator pair. If no CHAP credentials are set, an empty array is returned. CHAP credential information is provided in a JSON object with the following fields:

- * **InitiatorName** : The iSCSI initiator that connects to the target.
- * **SecretToAuthenticateInitiator** : The secret key that the initiator (for example, the Windows client) must provide to participate in mutual CHAP with the target.

- * **SecretToAuthenticateTarget** : The secret key that the target must provide to participate in mutual CHAP with the initiator (e.g. Windows client).
- * **TargetARN** : The Amazon Resource Name (ARN) of the storage volume.
- * *(dict)* –

Describes Challenge-Handshake Authentication Protocol (CHAP) information that supports authentication between your gateway and iSCSI initiators.

- **TargetARN** (*string*) –

The Amazon Resource Name (ARN) of the volume.

Valid Values : 50 to 500 lowercase letters, numbers, periods (.), and hyphens (-).

- **SecretToAuthenticateInitiator** (*string*) –

The secret key that the initiator (for example, the Windows client) must provide to participate in mutual CHAP with the target.

- **InitiatorName** (*string*) –

The iSCSI initiator that connects to the target.

- **SecretToAuthenticateTarget** (*string*) –

The secret key that the target must provide to participate in mutual CHAP with the initiator (e.g. Windows client).

describe_gateway_information (***kwargs*)

This operation returns metadata about a gateway such as its name, network interfaces, configured time zone, and the state (whether the gateway is running or not). To specify which gateway to describe, use the Amazon Resource Name (ARN) of the gateway in your request.

Request Syntax

```
response = client.describe_gateway_information(
    GatewayARN='string'
)
```

Parameters GatewayARN (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

Return type dict

Returns

Response Syntax

```
{
    'GatewayARN': 'string',
    'GatewayId': 'string',
    'GatewayName': 'string',
    'GatewayTimezone': 'string',
    'GatewayState': 'string',
    'GatewayNetworkInterfaces': [
        {
            'Ipv4Address': 'string',
            'MacAddress': 'string',
```

```
        'Ipv6Address': 'string'
    },
],
'GatewayType': 'string',
'NextUpdateAvailabilityDate': 'string',
'LastSoftwareUpdate': 'string'
}
```

Response Structure

- (*dict*) –

A JSON object containing the following fields:

- **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **GatewayId** (*string*) –

The gateway ID.

- **GatewayName** (*string*) –

The gateway name.

- **GatewayTimezone** (*string*) –

One of the values that indicates the time zone configured for the gateway.

- **GatewayState** (*string*) –

One of the values that indicates the operating state of the gateway.

- **GatewayNetworkInterfaces** (*list*) –

A NetworkInterface array that contains descriptions of the gateway network interfaces.

* (*dict*) –

Describes a gateway's network interface.

- **Ipv4Address** (*string*) –

The Internet Protocol version 4 (IPv4) address of the interface.

- **MacAddress** (*string*) –

The Media Access Control (MAC) address of the interface.

Note: This is currently unsupported and will not be returned in output.

- **Ipv6Address** (*string*) –

The Internet Protocol version 6 (IPv6) address of the interface. *Currently not supported*.

- **GatewayType** (*string*) –

The type of the gateway.

- **NextUpdateAvailabilityDate** (*string*) –

The date on which an update to the gateway is available. This date is in the time zone of the gateway. If the gateway is not available for an update this field is not returned in the response.

- **LastSoftwareUpdate** (*string*) –

The date on which the last software update was applied to the gateway. If the gateway has never been updated, this field does not return a value in the response.

describe_maintenance_start_time (***kwargs*)

This operation returns your gateway's weekly maintenance start time including the day and time of the week. Note that values are in terms of the gateway's time zone.

Request Syntax

```
response = client.describe_maintenance_start_time(
    GatewayARN='string'
)
```

Parameters **GatewayARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

Return type dict

Returns

Response Syntax

```
{
    'GatewayARN': 'string',
    'HourOfDay': 123,
    'MinuteOfHour': 123,
    'DayOfWeek': 123,
    'Timezone': 'string'
}
```

Response Structure

- (*dict*) –
 - **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.
 - **HourOfDay** (*integer*) –
 - **MinuteOfHour** (*integer*) –
 - **DayOfWeek** (*integer*) –
 - **Timezone** (*string*) –

describe_snapshot_schedule (***kwargs*)

This operation describes the snapshot schedule for the specified gateway volume. The snapshot schedule information includes intervals at which snapshots are automatically initiated on the volume.

Request Syntax

```
response = client.describe_snapshot_schedule(
    VolumeARN='string'
)
```

Parameters **VolumeARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the volume. Use the ListVolumes operation to return a list of gateway volumes.

Return type dict

Returns

Response Syntax

```
{
    'VolumeARN': 'string',
    'StartAt': 123,
    'RecurrenceInHours': 123,
    'Description': 'string',
    'Timezone': 'string'
}
```

Response Structure

- (*dict*) –
 - **VolumeARN** (*string*) –
 - **StartAt** (*integer*) –
 - **RecurrenceInHours** (*integer*) –
 - **Description** (*string*) –
 - **Timezone** (*string*) –

`describe_stored_iscsi_volumes` (**kwargs)

This operation returns the description of the gateway volumes specified in the request. The list of gateway volumes in the request must be from one gateway. In the response Amazon Storage Gateway returns volume information sorted by volume ARNs.

Request Syntax

```
response = client.describe_stored_iscsi_volumes(
    VolumeARNs=[
        'string',
    ]
)
```

Parameters **VolumeARNs** (*list*) – [REQUIRED]

An array of strings where each string represents the Amazon Resource Name (ARN) of a stored volume. All of the specified stored volumes must from the same gateway. Use `ListVolumes` to get volume ARNs for a gateway.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'StorediSCSIVolumes': [
        {
            'VolumeARN': 'string',
            'VolumeId': 'string',
            'VolumeType': 'string',
            'VolumeStatus': 'string',
            'VolumeSizeInBytes': 123,
            'VolumeProgress': 123.0,
            'VolumeDiskId': 'string',
            'SourceSnapshotId': 'string',
            'PreservedExistingData': True|False,
            'VolumeiSCSIAttributes': {
                'TargetARN': 'string',
```



```

        'NetworkInterfaceId': 'string',
        'NetworkInterfacePort': 123,
        'LunNumber': 123,
        'ChapEnabled': True|False
    },
    ],
}

```

Response Structure

- (dict) –
 - **StorediSCSIVolumes** (list) –
 - * (dict) –
 - **VolumeARN** (string) –
Lists iSCSI information about a volume.
 - **VolumeId** (string) –
 - **VolumeType** (string) –
 - **VolumeStatus** (string) –
 - **VolumeSizeInBytes** (integer) –
 - **VolumeProgress** (float) –
 - **VolumeDiskId** (string) –
 - **SourceSnapshotId** (string) –
 - **PreservedExistingData** (boolean) –
 - **VolumeiSCSIAttributes** (dict) –
 - **TargetARN** (string) –
The Amazon Resource Name (ARN) of the volume target.
 - **NetworkInterfaceId** (string) –
The network interface identifier.
 - **NetworkInterfacePort** (integer) –
The port used to communicate with iSCSI targets.
 - **LunNumber** (integer) –
The logical disk number.
 - **ChapEnabled** (boolean) –
Indicates whether mutual CHAP is enabled for the iSCSI target.

describe_tape_archives (**kwargs)

Returns a description of specified virtual tapes in the virtual tape shelf (VTS).

If a specific TapeARN is not specified, AWS Storage Gateway returns a description of all virtual tapes found in the VTS associated with your account.

Request Syntax

```

response = client.describe_tape_archives(
    TapeARNs=[
        'string',
    ],
    Marker='string',
    Limit=123
)

```

Parameters

- **TapeARNs** (*list*) – Specifies one or more unique Amazon Resource Names (ARNs) that represent the virtual tapes you want to describe.
 - (*string*) –
- **Marker** (*string*) – An opaque string that indicates the position at which to begin describing virtual tapes.
- **Limit** (*integer*) – Specifies that the number of virtual tapes described be limited to the specified number.

Return type dict

Returns

Response Syntax

```
{
  'TapeArchives': [
    {
      'TapeARN': 'string',
      'TapeBarcode': 'string',
      'TapeSizeInBytes': 123,
      'CompletionTime': datetime(2015, 1, 1),
      'RetrievedTo': 'string',
      'TapeStatus': 'string'
    },
  ],
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

DescribeTapeArchivesOutput

- **TapeArchives** (*list*) –

An array of virtual tape objects in the virtual tape shelf (VTS). The description includes of the Amazon Resource Name(ARN) of the virtual tapes. The information returned includes the Amazon Resource Names (ARNs) of the tapes, size of the tapes, status of the tapes, progress of the description and tape barcode.

- * (*dict*) –

Represents a virtual tape that is archived in the virtual tape shelf (VTS).

- **TapeARN** (*string*) –

The Amazon Resource Name (ARN) of an archived virtual tape.

- **TapeBarcode** (*string*) –

The barcode that identifies the archived virtual tape.

- **TapeSizeInBytes** (*integer*) –

The size, in bytes, of the archived virtual tape.

- **CompletionTime** (*datetime*) –

The time that the archiving of the virtual tape was completed.

The string format of the completion time is in the ISO8601 extended YYYY-MM-DD'T'HH:MM:SS'Z' format.

- **RetrievedTo** (*string*) –

The Amazon Resource Name (ARN) of the gateway-VTL that the virtual tape is being retrieved to.

The virtual tape is retrieved from the virtual tape shelf (VTS).

- **TapeStatus** (*string*) –

The current state of the archived virtual tape.

- **Marker** (*string*) –

An opaque string that indicates the position at which the virtual tapes that were fetched for description ended. Use this marker in your next request to fetch the next set of virtual tapes in the virtual tape shelf (VTS). If there are no more virtual tapes to describe, this field does not appear in the response.

describe_tape_recovery_points (***kwargs*)

Returns a list of virtual tape recovery points that are available for the specified gateway-VTL.

A recovery point is a point in time view of a virtual tape at which all the data on the virtual tape is consistent. If your gateway crashes, virtual tapes that have recovery points can be recovered to a new gateway.

Request Syntax

```
response = client.describe_tape_recovery_points(
    GatewayARN='string',
    Marker='string',
    Limit=123
)
```

Parameters

- **GatewayARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **Marker** (*string*) – An opaque string that indicates the position at which to begin describing the virtual tape recovery points.
- **Limit** (*integer*) – Specifies that the number of virtual tape recovery points that are described be limited to the specified number.

Return type dict

Returns

Response Syntax

```
{
  'GatewayARN': 'string',
  'TapeRecoveryPointInfos': [
    {
      'TapeARN': 'string',
      'TapeRecoveryPointTime': datetime(2015, 1, 1),
      'TapeSizeInBytes': 123,
      'TapeStatus': 'string'
    },
  ],
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

DescribeTapeRecoveryPointsOutput

- **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **TapeRecoveryPointInfos** (*list*) –

An array of TapeRecoveryPointInfos that are available for the specified gateway.

- * (*dict*) –

Describes a recovery point.

- **TapeARN** (*string*) –

The Amazon Resource Name (ARN) of the virtual tape.

- **TapeRecoveryPointTime** (*datetime*) –

The time when the point-in-time view of the virtual tape was replicated for later recovery.

The string format of the tape recovery point time is in the ISO8601 extended YYYY-MM-DD'T'HH:MM:SS'Z' format.

- **TapeSizeInBytes** (*integer*) –

The size, in bytes, of the virtual tapes to recover.

- **TapeStatus** (*string*) –

- **Marker** (*string*) –

An opaque string that indicates the position at which the virtual tape recovery points that were listed for description ended.

Use this marker in your next request to list the next set of virtual tape recovery points in the list. If there are no more recovery points to describe, this field does not appear in the response.

describe_tapes (***kwargs*)

Returns a description of the specified Amazon Resource Name (ARN) of virtual tapes. If a TapeARN is not specified, returns a description of all virtual tapes associated with the specified gateway.

Request Syntax

```
response = client.describe_tapes(  
    GatewayARN='string',  
    TapeARNs=[  
        'string',  
    ],  
    Marker='string',  
    Limit=123  
)
```

Parameters

- **GatewayARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **TapeARNs** (*list*) – Specifies one or more unique Amazon Resource Names (ARNs) that represent the virtual tapes you want to describe. If this param-

ter is not specified, AWS Storage Gateway returns a description of all virtual tapes associated with the specified gateway.

– (*string*) –

- **Marker** (*string*) – A marker value, obtained in a previous call to `DescribeTapes`. This marker indicates which page of results to retrieve.

If not specified, the first page of results is retrieved.

- **Limit** (*integer*) – Specifies that the number of virtual tapes described be limited to the specified number.

Note: Amazon Web Services may impose its own limit, if this field is not set.

Return type dict

Returns

Response Syntax

```
{
  'Tapes': [
    {
      'TapeARN': 'string',
      'TapeBarcode': 'string',
      'TapeSizeInBytes': 123,
      'TapeStatus': 'string',
      'VTLDevice': 'string',
      'Progress': 123.0
    },
  ],
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –

`DescribeTapesOutput`

– **Tapes** (*list*) –

An array of virtual tape descriptions.

* (*dict*) –

Describes a virtual tape object.

- **TapeARN** (*string*) –

The Amazon Resource Name (ARN) of the virtual tape.

- **TapeBarcode** (*string*) –

The barcode that identifies a specific virtual tape.

- **TapeSizeInBytes** (*integer*) –

The size, in bytes, of the virtual tape.

- **TapeStatus** (*string*) –

The current state of the virtual tape.

- **VTLDevice** (*string*) –

The virtual tape library (VTL) device that the virtual tape is associated with.

- **Progress** (*float*) –

For archiving virtual tapes, indicates how much data remains to be uploaded before archiving is complete.

Range: 0 (not started) to 100 (complete).

– **Marker** (*string*) –

An opaque string which can be used as part of a subsequent DescribeTapes call to retrieve the next page of results.

If a response does not contain a marker, then there are no more results to be retrieved.

describe_upload_buffer (***kwargs*)

This operation returns information about the upload buffer of a gateway. This operation is supported for both the gateway-stored and gateway-cached volume architectures.

The response includes disk IDs that are configured as upload buffer space, and it includes the amount of upload buffer space allocated and used.

Request Syntax

```
response = client.describe_upload_buffer(
    GatewayARN='string'
)
```

Parameters **GatewayARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

Return type dict

Returns

Response Syntax

```
{
    'GatewayARN': 'string',
    'DiskIds': [
        'string',
    ],
    'UploadBufferUsedInBytes': 123,
    'UploadBufferAllocatedInBytes': 123
}
```

Response Structure

• (*dict*) –

– **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

– **DiskIds** (*list*) –

* (*string*) –

– **UploadBufferUsedInBytes** (*integer*) –

– **UploadBufferAllocatedInBytes** (*integer*) –

describe_vtl_devices (***kwargs*)

Returns a description of virtual tape library (VTL) devices for the specified gateway. In the response, AWS Storage Gateway returns VTL device information.

The list of VTL devices must be from one gateway.

Request Syntax

```

response = client.describe_vtl_devices(
    GatewayARN='string',
    VTLDeviceARNs=[
        'string',
    ],
    Marker='string',
    Limit=123
)

```

Parameters

- **GatewayARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **VTLDeviceARNs** (*list*) – An array of strings, where each string represents the Amazon Resource Name (ARN) of a VTL device.

Note: All of the specified VTL devices must be from the same gateway. If no VTL devices are specified, the result will contain all devices on the specified gateway.

– (*string*) –

- **Marker** (*string*) – An opaque string that indicates the position at which to begin describing the VTL devices.
- **Limit** (*integer*) – Specifies that the number of VTL devices described be limited to the specified number.

Return type dict

Returns

Response Syntax

```

{
  'GatewayARN': 'string',
  'VTLDevices': [
    {
      'VTLDeviceARN': 'string',
      'VTLDeviceType': 'string',
      'VTLDeviceVendor': 'string',
      'VTLDeviceProductIdentifier': 'string',
      'DeviceiSCSIAttributes': {
        'TargetARN': 'string',
        'NetworkInterfaceId': 'string',
        'NetworkInterfacePort': 123,
        'ChapEnabled': True|False
      }
    },
  ],
  'Marker': 'string'
}

```

Response Structure

- (*dict*) –

DescribeVTLDevicesOutput

- **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

– **VTLDevices** (*list*) –

An array of VTL device objects composed of the Amazon Resource Name(ARN) of the VTL devices.

* (*dict*) –

Represents a device object associated with a gateway-VTL.

· **VTLDeviceARN** (*string*) –

Specifies the unique Amazon Resource Name (ARN) of the device (tape drive or media changer).

· **VTLDeviceType** (*string*) –

· **VTLDeviceVendor** (*string*) –

· **VTLDeviceProductIdentifier** (*string*) –

· **DeviceiSCSIAttributes** (*dict*) –

A list of iSCSI information about a VTL device.

· **TargetARN** (*string*) –

Specifies the unique Amazon Resource Name(ARN) that encodes the iSCSI qualified name(iqn) of a tape drive or media changer target.

· **NetworkInterfaceId** (*string*) –

The network interface identifier of the VTL device.

· **NetworkInterfacePort** (*integer*) –

The port used to communicate with iSCSI VTL device targets.

· **ChapEnabled** (*boolean*) –

Indicates whether mutual CHAP is enabled for the iSCSI target.

– **Marker** (*string*) –

An opaque string that indicates the position at which the VTL devices that were fetched for description ended. Use the marker in your next request to fetch the next set of VTL devices in the list. If there are no more VTL devices to describe, this field does not appear in the response.

describe_working_storage (***kwargs*)

This operation returns information about the working storage of a gateway. This operation is supported only for the gateway-stored volume architecture. This operation is deprecated in cached-volumes API version (20120630). Use DescribeUploadBuffer instead.

Note: Working storage is also referred to as upload buffer. You can also use the DescribeUploadBuffer operation to add upload buffer to a stored-volume gateway.

The response includes disk IDs that are configured as working storage, and it includes the amount of working storage allocated and used.

Request Syntax

```
response = client.describe_working_storage(  
    GatewayARN='string'  
)
```

Parameters **GatewayARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

Return type dict**Returns****Response Syntax**

```
{
    'GatewayARN': 'string',
    'DiskIds': [
        'string',
    ],
    'WorkingStorageUsedInBytes': 123,
    'WorkingStorageAllocatedInBytes': 123
}
```

Response Structure

- (*dict*) –

A JSON object containing the following fields:

- **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **DiskIds** (*list*) –

An array of the gateway's local disk IDs that are configured as working storage. Each local disk ID is specified as a string (minimum length of 1 and maximum length of 300). If no local disks are configured as working storage, then the DiskIds array is empty.

* (*string*) –

- **WorkingStorageUsedInBytes** (*integer*) –

The total working storage in bytes in use by the gateway. If no working storage is configured for the gateway, this field returns 0.

- **WorkingStorageAllocatedInBytes** (*integer*) –

The total working storage in bytes allocated for the gateway. If no working storage is configured for the gateway, this field returns 0.

disable_gateway (***kwargs*)

Disables a gateway when the gateway is no longer functioning. For example, if your gateway VM is damaged, you can disable the gateway so you can recover virtual tapes.

Use this operation for a gateway-VTL that is not reachable or not functioning.

Warning: Once a gateway is disabled it cannot be enabled.

Request Syntax

```
response = client.disable_gateway(
    GatewayARN='string'
)
```

Parameters **GatewayARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

Return type dict

Returns**Response Syntax**

```
{
    'GatewayARN': 'string'
}
```

Response Structure

- (*dict*) –

DisableGatewayOutput

- **GatewayARN** (*string*) –

The unique Amazon Resource Name of the disabled gateway.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to *ClientMethod*.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type L{botocore.paginate.Paginator}

Returns A paginator object.

get_waiter (*waiter_name*)

list_gateways (***kwargs*)

This operation lists gateways owned by an AWS account in a region specified in the request. The returned list is ordered by gateway Amazon Resource Name (ARN).

By default, the operation returns a maximum of 100 gateways. This operation supports pagination that allows you to optionally reduce the number of gateways returned in a response.

If you have more gateways than are returned in a response—that is, the response returns only a truncated list of your gateways—the response contains a marker that you can specify in your next request to fetch the next page of gateways.

Request Syntax

```
response = client.list_gateways(
    Marker='string',
    Limit=123
)
```

Parameters

- **Marker** (*string*) – An opaque string that indicates the position at which to begin the returned list of gateways.
- **Limit** (*integer*) – Specifies that the list of gateways returned be limited to the specified number of items.

Return type dict

Returns

Response Syntax

```
{
  'Gateways': [
    {
      'GatewayARN': 'string',
      'GatewayType': 'string',
      'GatewayOperationalState': 'string',
      'GatewayName': 'string'
    },
  ],
  'Marker': 'string'
}
```

Response Structure

- (*dict*) –
 - **Gateways** (*list*) –
 - * (*dict*) –
 - **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.
 - **GatewayType** (*string*) –
 - **GatewayOperationalState** (*string*) –
 - **GatewayName** (*string*) –
 - **Marker** (*string*) –

list_local_disks (***kwargs*)

This operation returns a list of the gateway’s local disks. To specify which gateway to describe, you use the Amazon Resource Name (ARN) of the gateway in the body of the request.

The request returns a list of all disks, specifying which are configured as working storage, cache storage, or stored volume or not configured at all. The response includes a `DiskStatus` field. This field can have a value of `present` (the disk is available to use), `missing` (the disk is no longer connected to the gateway), or `mismatch` (the disk node is occupied by a disk that has incorrect metadata or the disk content is corrupted).

Request Syntax

```
response = client.list_local_disks(
    GatewayARN='string'
)
```

Parameters **GatewayARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

Return type dict

Returns

Response Syntax

```
{
  'GatewayARN': 'string',
  'Disks': [
    {
      'DiskId': 'string',
      'DiskPath': 'string',
      'DiskNode': 'string',
      'DiskStatus': 'string',
      'DiskSizeInBytes': 123,
      'DiskAllocationType': 'string',
      'DiskAllocationResource': 'string'
    },
  ]
}
```

Response Structure

- (*dict*) –
 - **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.
 - **Disks** (*list*) –
 - * (*dict*) –
 - **DiskId** (*string*) –
 - **DiskPath** (*string*) –
 - **DiskNode** (*string*) –
 - **DiskStatus** (*string*) –
 - **DiskSizeInBytes** (*integer*) –
 - **DiskAllocationType** (*string*) –
 - **DiskAllocationResource** (*string*) –

`list_tags_for_resource` (**kwargs)

This operation lists the tags that have been added to the specified resource.

Request Syntax

```
response = client.list_tags_for_resource(
    ResourceARN='string',
    Marker='string',
    Limit=123
)
```

Parameters

- **ResourceARN** (*string*) – The Amazon Resource Name (ARN) of the resource for which you want to list tags.
- **Marker** (*string*) – An opaque string that indicates the position at which to begin returning the list of tags.
- **Limit** (*integer*) – Specifies that the list of tags returned be limited to the specified number of items.

Return type dict

Returns

Response Syntax

```
{
  'ResourceARN': 'string',
  'Marker': 'string',
}
```

```

    'Tags': [
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
}

```

Response Structure

- (*dict*) –

ListTagsForResourceOutput

- **ResourceARN** (*string*) –

The Amazon Resource Name (ARN) of the resource for which you want to list tags.

- **Marker** (*string*) –

An opaque string that indicates the position at which to stop returning the list of tags.

- **Tags** (*list*) –

An array that contains the tags for the specified resource.

- * (*dict*) –
 - **Key** (*string*) –
 - **Value** (*string*) –

list_volume_initiators (***kwargs*)

This operation lists iSCSI initiators that are connected to a volume. You can use this operation to determine whether a volume is being used or not.

Request Syntax

```

response = client.list_volume_initiators(
    VolumeARN='string'
)

```

Parameters **VolumeARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the volume. Use the ListVolumes operation to return a list of gateway volumes for the gateway.

Return type dict

Returns

Response Syntax

```

{
    'Initiators': [
        'string',
    ]
}

```

Response Structure

- (*dict*) –

ListVolumeInitiatorsOutput

- **Initiators** (*list*) –

The host names and port numbers of all iSCSI initiators that are connected to the gateway.

* (*string*) –

list_volume_recovery_points (***kwargs*)

This operation lists the recovery points for a specified gateway. This operation is supported only for the gateway-cached volume architecture.

Each gateway-cached volume has one recovery point. A volume recovery point is a point in time at which all data of the volume is consistent and from which you can create a snapshot. To create a snapshot from a volume recovery point use the `CreateSnapshotFromVolumeRecoveryPoint` operation.

Request Syntax

```
response = client.list_volume_recovery_points(
    GatewayARN='string'
)
```

Parameters **GatewayARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the gateway. Use the `ListGateways` operation to return a list of gateways for your account and region.

Return type dict

Returns

Response Syntax

```
{
  'GatewayARN': 'string',
  'VolumeRecoveryPointInfos': [
    {
      'VolumeARN': 'string',
      'VolumeSizeInBytes': 123,
      'VolumeUsageInBytes': 123,
      'VolumeRecoveryPointTime': 'string'
    },
  ]
}
```

Response Structure

- (*dict*) –
 - **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the `ListGateways` operation to return a list of gateways for your account and region.
 - **VolumeRecoveryPointInfos** (*list*) –
 - * (*dict*) –
 - **VolumeARN** (*string*) –
 - **VolumeSizeInBytes** (*integer*) –
 - **VolumeUsageInBytes** (*integer*) –
 - **VolumeRecoveryPointTime** (*string*) –

list_volumes (***kwargs*)

This operation lists the iSCSI stored volumes of a gateway. Results are sorted by volume ARN. The response includes only the volume ARNs. If you want additional volume information, use the `DescribeStorediSCSIVolumes` API.

The operation supports pagination. By default, the operation returns a maximum of up to 100 volumes. You can optionally specify the `Limit` field in the body to limit the number of volumes in the response. If the number of volumes returned in the response is truncated, the response includes a `Marker` field. You can use this `Marker` value in your subsequent request to retrieve the next set of volumes.

Request Syntax

```
response = client.list_volumes(
    GatewayARN='string',
    Marker='string',
    Limit=123
)
```

Parameters

- **GatewayARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **Marker** (*string*) – A string that indicates the position at which to begin the returned list of volumes. Obtain the marker from the response of a previous List iSCSI Volumes request.
- **Limit** (*integer*) – Specifies that the list of volumes returned be limited to the specified number of items.

Return type dict

Returns

Response Syntax

```
{
    'GatewayARN': 'string',
    'Marker': 'string',
    'VolumeInfos': [
        {
            'VolumeARN': 'string',
            'VolumeType': 'string'
        },
    ]
}
```

Response Structure

- (*dict*) –
 - **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.
 - **Marker** (*string*) –
 - **VolumeInfos** (*list*) –
 - * (*dict*) –
 - **VolumeARN** (*string*) –
 - **VolumeType** (*string*) –

remove_tags_from_resource (**kwargs)

This operation removes one or more tags from the specified resource.

Request Syntax

```
response = client.remove_tags_from_resource(
    ResourceARN='string',
    TagKeys=[
        'string',
    ]
)
```

Parameters

- **ResourceARN** (*string*) – The Amazon Resource Name (ARN) of the resource you want to remove the tags from.
- **TagKeys** (*list*) – The keys of the tags you want to remove from the specified resource. A tag is composed of a key/value pair.
 - (*string*) –

Return type dict**Returns****Response Syntax**

```
{
    'ResourceARN': 'string'
}
```

Response Structure

- (*dict*) –
 - RemoveTagsFromResourceOutput
 - **ResourceARN** (*string*) –

The Amazon Resource Name (ARN) of the resource that the tags were removed from.

reset_cache (***kwargs*)

This operation resets all cache disks that have encountered a error and makes the disks available for reconfiguration as cache storage. If your cache disk encounters a error, the gateway prevents read and write operations on virtual tapes in the gateway. For example, an error can occur when a disk is corrupted or removed from the gateway. When a cache is reset, the gateway loses its cache storage. At this point you can reconfigure the disks as cache disks.

Warning: If the cache disk you are resetting contains data that has not been uploaded to Amazon S3 yet, that data can be lost. After you reset cache disks, there will be no configured cache disks left in the gateway, so you must configure at least one new cache disk for your gateway to function properly.

Request Syntax

```
response = client.reset_cache(
    GatewayARN='string'
)
```

Parameters **GatewayARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

Return type dict**Returns****Response Syntax**

```
{
    'GatewayARN': 'string'
}
```

Response Structure

- (*dict*) –

– **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

retrieve_tape_archive (***kwargs*)

Retrieves an archived virtual tape from the virtual tape shelf (VTS) to a gateway-VTL. Virtual tapes archived in the VTS are not associated with any gateway. However after a tape is retrieved, it is associated with a gateway, even though it is also listed in the VTS.

Once a tape is successfully retrieved to a gateway, it cannot be retrieved again to another gateway. You must archive the tape again before you can retrieve it to another gateway.

Request Syntax

```
response = client.retrieve_tape_archive(
    TapeARN='string',
    GatewayARN='string'
)
```

Parameters

• **TapeARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the virtual tape you want to retrieve from the virtual tape shelf (VTS).

• **GatewayARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the gateway you want to retrieve the virtual tape to. Use the ListGateways operation to return a list of gateways for your account and region.

You retrieve archived virtual tapes to only one gateway and the gateway must be a gateway-VTL.

Return type dict

Returns

Response Syntax

```
{
    'TapeARN': 'string'
}
```

Response Structure

• (*dict*) –

RetrieveTapeArchiveOutput

– **TapeARN** (*string*) –

The Amazon Resource Name (ARN) of the retrieved virtual tape.

retrieve_tape_recovery_point (***kwargs*)

Retrieves the recovery point for the specified virtual tape.

A recovery point is a point in time view of a virtual tape at which all the data on the tape is consistent. If your gateway crashes, virtual tapes that have recovery points can be recovered to a new gateway.

Note: The virtual tape can be retrieved to only one gateway. The retrieved tape is read-only. The virtual tape can be retrieved to only a gateway-VTL. There is no charge for retrieving recovery points.

Request Syntax

```
response = client.retrieve_tape_recovery_point(
    TapeARN='string',
    GatewayARN='string'
)
```

Parameters

- **TapeARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the virtual tape for which you want to retrieve the recovery point.

- **GatewayARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

Return type dict

Returns**Response Syntax**

```
{
    'TapeARN': 'string'
}
```

Response Structure

- (*dict*) –

RetrieveTapeRecoveryPointOutput

- **TapeARN** (*string*) –

The Amazon Resource Name (ARN) of the virtual tape for which the recovery point was retrieved.

shutdown_gateway (***kwargs*)

This operation shuts down a gateway. To specify which gateway to shut down, use the Amazon Resource Name (ARN) of the gateway in the body of your request.

The operation shuts down the gateway service component running in the storage gateway's virtual machine (VM) and not the VM.

Note: If you want to shut down the VM, it is recommended that you first shut down the gateway component in the VM to avoid unpredictable conditions.

After the gateway is shutdown, you cannot call any other API except StartGateway , DescribeGatewayInformation , and ListGateways . For more information, see ActivateGateway . Your applications cannot read from or write to the gateway's storage volumes, and there are no snapshots taken.

Note: When you make a shutdown request, you will get a 200 OK success response immediately. However, it might take some time for the gateway to shut down. You can call the DescribeGatewayInformation API to check the status. For more information, see ActivateGateway .

If do not intend to use the gateway again, you must delete the gateway (using DeleteGateway) to no longer pay software charges associated with the gateway.

Request Syntax

```
response = client.shutdown_gateway(
    GatewayARN='string'
)
```

Parameters `GatewayARN` (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the gateway. Use the `ListGateways` operation to return a list of gateways for your account and region.

Return type dict

Returns**Response Syntax**

```
{
    'GatewayARN': 'string'
}
```

Response Structure

- (*dict*) –

A JSON object containing the of the gateway that was shut down.

- **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the `ListGateways` operation to return a list of gateways for your account and region.

start_gateway (***kwargs*)

This operation starts a gateway that you previously shut down (see `ShutdownGateway`). After the gateway starts, you can then make other API calls, your applications can read from or write to the gateway's storage volumes and you will be able to take snapshot backups.

Note: When you make a request, you will get a 200 OK success response immediately. However, it might take some time for the gateway to be ready. You should call `DescribeGatewayInformation` and check the status before making any additional API calls. For more information, see `ActivateGateway` .

To specify which gateway to start, use the Amazon Resource Name (ARN) of the gateway in your request.

Request Syntax

```
response = client.start_gateway(
    GatewayARN='string'
)
```

Parameters `GatewayARN` (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the gateway. Use the `ListGateways` operation to return a list of gateways for your account and region.

Return type dict

Returns**Response Syntax**

```
{
    'GatewayARN': 'string'
}
```

Response Structure

- (*dict*) –

A JSON object containing the of the gateway that was restarted.

– **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

update_bandwidth_rate_limit (***kwargs*)

This operation updates the bandwidth rate limits of a gateway. You can update both the upload and download bandwidth rate limit or specify only one of the two. If you don't set a bandwidth rate limit, the existing rate limit remains.

By default, a gateway's bandwidth rate limits are not set. If you don't set any limit, the gateway does not have any limitations on its bandwidth usage and could potentially use the maximum available bandwidth.

To specify which gateway to update, use the Amazon Resource Name (ARN) of the gateway in your request.

Request Syntax

```
response = client.update_bandwidth_rate_limit(
    GatewayARN='string',
    AverageUploadRateLimitInBitsPerSec=123,
    AverageDownloadRateLimitInBitsPerSec=123
)
```

Parameters

• **GatewayARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

• **AverageUploadRateLimitInBitsPerSec** (*integer*) – The average upload bandwidth rate limit in bits per second.

• **AverageDownloadRateLimitInBitsPerSec** (*integer*) – The average download bandwidth rate limit in bits per second.

Return type dict

Returns

Response Syntax

```
{
    'GatewayARN': 'string'
}
```

Response Structure

• (*dict*) –

A JSON object containing the of the gateway whose throttle information was updated.

– **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

update_chap_credentials (***kwargs*)

This operation updates the Challenge-Handshake Authentication Protocol (CHAP) credentials for a specified iSCSI target. By default, a gateway does not have CHAP enabled; however, for added security, you might use it.

Warning: When you update CHAP credentials, all existing connections on the target are closed and initiators must reconnect with the new credentials.
--

Request Syntax

```
response = client.update_chap_credentials(
    TargetARN='string',
    SecretToAuthenticateInitiator='string',
    InitiatorName='string',
    SecretToAuthenticateTarget='string'
)
```

Parameters

- **TargetARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the iSCSI volume target. Use the DescribeStorediSCSIVolumes operation to return the TargetARN for specified VolumeARN.

- **SecretToAuthenticateInitiator** (*string*) – [REQUIRED]

The secret key that the initiator (for example, the Windows client) must provide to participate in mutual CHAP with the target.

Note: The secret key must be between 12 and 16 bytes when encoded in UTF-8.

- **InitiatorName** (*string*) – [REQUIRED]

The iSCSI initiator that connects to the target.

- **SecretToAuthenticateTarget** (*string*) – The secret key that the target must provide to participate in mutual CHAP with the initiator (e.g. Windows client).

Byte constraints: Minimum bytes of 12. Maximum bytes of 16.

Note: The secret key must be between 12 and 16 bytes when encoded in UTF-8.

Return type dict

Returns

Response Syntax

```
{
    'TargetARN': 'string',
    'InitiatorName': 'string'
}
```

Response Structure

- (*dict*) –

A JSON object containing the following fields:

- **TargetARN** (*string*) –

The Amazon Resource Name (ARN) of the target. This is the same target specified in the request.

- **InitiatorName** (*string*) –

The iSCSI initiator that connects to the target. This is the same initiator name specified in the request.

update_gateway_information (***kwargs*)

This operation updates a gateway’s metadata, which includes the gateway’s name and time zone. To specify which gateway to update, use the Amazon Resource Name (ARN) of the gateway in your request.

Request Syntax

```
response = client.update_gateway_information(
    GatewayARN='string',
    GatewayName='string',
    GatewayTimezone='string'
)
```

Parameters

- **GatewayARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **GatewayName** (*string*) – A unique identifier for your gateway. This name becomes part of the gateway Amazon Resources Name (ARN) which is what you use as an input to other operations.
- **GatewayTimezone** (*string*) –

Return type dict

Returns**Response Syntax**

```
{
    'GatewayARN': 'string',
    'GatewayName': 'string'
}
```

Response Structure

- (*dict*) –

A JSON object containing the of the gateway that was updated.

- **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **GatewayName** (*string*) –

update_gateway_software_now (***kwargs*)

This operation updates the gateway virtual machine (VM) software. The request immediately triggers the software update.

Note: When you make this request, you get a 200 OK success response immediately. However, it might take some time for the update to complete. You can call DescribeGatewayInformation to verify the gateway is in the STATE_RUNNING state.

Warning: A software update forces a system restart of your gateway. You can minimize the chance of any disruption to your applications by increasing your iSCSI Initiators' timeouts. For more information about increasing iSCSI Initiator timeouts for Windows and Linux, see [Customizing Your Windows iSCSI Settings](#) and [Customizing Your Linux iSCSI Settings](#) , respectively.

Request Syntax

```
response = client.update_gateway_software_now(
    GatewayARN='string'
)
```

Parameters **GatewayARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

Return type dict

Returns

Response Syntax

```
{
    'GatewayARN': 'string'
}
```

Response Structure

- (*dict*) –

A JSON object containing the of the gateway that was updated.

- **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

update_maintenance_start_time (***kwargs*)

This operation updates a gateway’s weekly maintenance start time information, including day and time of the week. The maintenance time is the time in your gateway’s time zone.

Request Syntax

```
response = client.update_maintenance_start_time(
    GatewayARN='string',
    HourOfDay=123,
    MinuteOfHour=123,
    DayOfWeek=123
)
```

Parameters

- **GatewayARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **HourOfDay** (*integer*) – **[REQUIRED]**

The hour component of the maintenance start time represented as *hh* , where *hh* is the hour (00 to 23). The hour of the day is in the time zone of the gateway.

- **MinuteOfHour** (*integer*) – **[REQUIRED]**

The minute component of the maintenance start time represented as *mm* , where *mm* is the minute (00 to 59). The minute of the hour is in the time zone of the gateway.

- **DayOfWeek** (*integer*) – **[REQUIRED]**

The maintenance start time day of the week.

Return type dict

Returns

Response Syntax

```
{
    'GatewayARN': 'string'
}
```

Response Structure

- (*dict*) –

A JSON object containing the of the gateway whose maintenance start time is updated.

- **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

update_snapshot_schedule (***kwargs*)

This operation updates a snapshot schedule configured for a gateway volume.

The default snapshot schedule for volume is once every 24 hours, starting at the creation time of the volume. You can use this API to change the snapshot schedule configured for the volume.

In the request you must identify the gateway volume whose snapshot schedule you want to update, and the schedule information, including when you want the snapshot to begin on a day and the frequency (in hours) of snapshots.

Request Syntax

```
response = client.update_snapshot_schedule(  
    VolumeARN='string',  
    StartAt=123,  
    RecurrenceInHours=123,  
    Description='string'  
)
```

Parameters

- **VolumeARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the volume. Use the ListVolumes operation to return a list of gateway volumes.

- **StartAt** (*integer*) – [REQUIRED]

The hour of the day at which the snapshot schedule begins represented as *hh* , where *hh* is the hour (0 to 23). The hour of the day is in the time zone of the gateway.

- **RecurrenceInHours** (*integer*) – [REQUIRED]

Frequency of snapshots. Specify the number of hours between snapshots.

- **Description** (*string*) – Optional description of the snapshot that overwrites the existing description.

Return type dict

Returns

Response Syntax

```
{  
    'VolumeARN': 'string'  
}
```

Response Structure

- (*dict*) –

A JSON object containing the of the updated storage volume.

- **VolumeARN** (*string*) –

update_vtl_device_type (**kwargs)

This operation updates the type of medium changer in a gateway-VTL. When you activate a gateway-VTL, you select a medium changer type for the gateway-VTL. This operation enables you to select a different type of medium changer after a gateway-VTL is activated.

Request Syntax

```
response = client.update_vtl_device_type(
    VTLDeviceARN='string',
    DeviceType='string'
)
```

Parameters

- **VTLDeviceARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the medium changer you want to select.

- **DeviceType** (*string*) – **[REQUIRED]**

The type of medium changer you want to select.

Valid Values : “STK-L700”, “AWS-Gateway-VTL”

Return type dict

Returns**Response Syntax**

```
{
    'VTLDeviceARN': 'string'
}
```

Response Structure

- (*dict*) –

UpdateVTLDeviceTypeOutput

- **VTLDeviceARN** (*string*) –

The Amazon Resource Name (ARN) of the medium changer you have selected.

Paginators

The available paginators are:

- `StorageGateway.Paginator.DescribeTapeArchives`
- `StorageGateway.Paginator.DescribeTapeRecoveryPoints`
- `StorageGateway.Paginator.DescribeTapes`
- `StorageGateway.Paginator.DescribeVTLDevices`
- `StorageGateway.Paginator.ListGateways`
- `StorageGateway.Paginator.ListVolumes`

class `StorageGateway.Paginator.DescribeTapeArchives`

```
paginator = client.get_paginator('describe_tape_archives')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `StorageGateway.Client.describe_tape_archives()`.

Request Syntax

```
response_iterator = paginator.paginate(
    TapeARNs=[
        'string',
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **TapeARNs** (*list*) – Specifies one or more unique Amazon Resource Names (ARNs) that represent the virtual tapes you want to describe.
 - (*string*) –
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns**Response Syntax**

```
{
    'TapeArchives': [
        {
            'TapeARN': 'string',
            'TapeBarcode': 'string',
            'TapeSizeInBytes': 123,
            'CompletionTime': datetime(2015, 1, 1),
            'RetrievedTo': 'string',
            'TapeStatus': 'string'
        },
    ],
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
`DescribeTapeArchivesOutput`

– **TapeArchives** (*list*) –

An array of virtual tape objects in the virtual tape shelf (VTS). The description includes of the Amazon Resource Name(ARN) of the virtual tapes. The information returned includes the Amazon Resource Names (ARNs) of the tapes, size of the tapes, status of the tapes, progress of the description and tape barcode.

* (*dict*) –

Represents a virtual tape that is archived in the virtual tape shelf (VTS).

· **TapeARN** (*string*) –

The Amazon Resource Name (ARN) of an archived virtual tape.

· **TapeBarcode** (*string*) –

The barcode that identifies the archived virtual tape.

· **TapeSizeInBytes** (*integer*) –

The size, in bytes, of the archived virtual tape.

· **CompletionTime** (*datetime*) –

The time that the archiving of the virtual tape was completed.

The string format of the completion time is in the ISO8601 extended YYYY-MM-DD'T'HH:MM:SS'Z' format.

· **RetrievedTo** (*string*) –

The Amazon Resource Name (ARN) of the gateway-VTL that the virtual tape is being retrieved to.

The virtual tape is retrieved from the virtual tape shelf (VTS).

· **TapeStatus** (*string*) –

The current state of the archived virtual tape.

– **NextToken** (*string*) –

A token to resume pagination.

class StorageGateway.Paginator.**DescribeTapeRecoveryPoints**

```
paginator = client.get_paginator('describe_tape_recovery_points')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `StorageGateway.Client.describe_tape_recovery_points()`.

Request Syntax

```
response_iterator = paginator.paginate(
    GatewayARN='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **GatewayARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'GatewayARN': 'string',
  'TapeRecoveryPointInfos': [
    {
      'TapeARN': 'string',
      'TapeRecoveryPointTime': datetime(2015, 1, 1),
      'TapeSizeInBytes': 123,
      'TapeStatus': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

DescribeTapeRecoveryPointsOutput

- **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **TapeRecoveryPointInfos** (*list*) –

An array of TapeRecoveryPointInfos that are available for the specified gateway.

- * (*dict*) –

Describes a recovery point.

- **TapeARN** (*string*) –

The Amazon Resource Name (ARN) of the virtual tape.

- **TapeRecoveryPointTime** (*datetime*) –

The time when the point-in-time view of the virtual tape was replicated for later recovery.

The string format of the tape recovery point time is in the ISO8601 extended YYYY-MM-DD'T'HH:MM:SS'Z' format.

- **TapeSizeInBytes** (*integer*) –

The size, in bytes, of the virtual tapes to recover.

- **TapeStatus** (*string*) –

- **NextToken** (*string*) –

A token to resume pagination.

class StorageGateway.Paginator.**DescribeTapes**

```
paginator = client.get_paginator('describe_tapes')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `StorageGateway.Client.describe_tapes()`.

Request Syntax

```
response_iterator = paginator.paginate(
    GatewayARN='string',
    TapeARNs=[
        'string',
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **GatewayARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **TapeARNs** (*list*) – Specifies one or more unique Amazon Resource Names (ARNs) that represent the virtual tapes you want to describe. If this parameter is not specified, AWS Storage Gateway returns a description of all virtual tapes associated with the specified gateway.

– (*string*) –

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'Tapes': [
    {
      'TapeARN': 'string',
      'TapeBarcode': 'string',
      'TapeSizeInBytes': 123,
      'TapeStatus': 'string',
      'VTLDevice': 'string',
      'Progress': 123.0
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

DescribeTapesOutput

- **Tapes** (*list*) –

An array of virtual tape descriptions.

- * (*dict*) –

Describes a virtual tape object.

- **TapeARN** (*string*) –

The Amazon Resource Name (ARN) of the virtual tape.

- **TapeBarcode** (*string*) –

The barcode that identifies a specific virtual tape.

- **TapeSizeInBytes** (*integer*) –

The size, in bytes, of the virtual tape.

- **TapeStatus** (*string*) –

The current state of the virtual tape.

- **VTLDevice** (*string*) –

The virtual tape library (VTL) device that the virtual tape is associated with.

- **Progress** (*float*) –

For archiving virtual tapes, indicates how much data remains to be uploaded before archiving is complete.

Range: 0 (not started) to 100 (complete).

- **NextToken** (*string*) –

A token to resume pagination.

class StorageGateway.Paginator.**DescribeVTLDevices**

```
paginator = client.get_paginator('describe_vtl_devices')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `StorageGateway.Client.describe_vtl_devices()`.

Request Syntax

```
response_iterator = paginator.paginate(
    GatewayARN='string',
    VTLDeviceARNs=[
        'string',
    ],
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **GatewayARN** (*string*) – **[REQUIRED]**

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **VTLDeviceARNs** (*list*) – An array of strings, where each string represents the Amazon Resource Name (ARN) of a VTL device.

Note: All of the specified VTL devices must be from the same gateway. If no VTL devices are specified, the result will contain all devices on the specified gateway.

– (*string*) –

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
    'GatewayARN': 'string',
    'VTLDevices': [
        {
            'VTLDeviceARN': 'string',
            'VTLDeviceType': 'string',
            'VTLDeviceVendor': 'string',
            'VTLDeviceProductIdentifier': 'string',
            'DeviceiSCSIAttributes': {
                'TargetARN': 'string',
                'NetworkInterfaceId': 'string',
                'NetworkInterfacePort': 123,
```

```
        'ChapEnabled': True|False
    },
    ],
    'NextToken': 'string'
}
```

Response Structure

- *(dict)* –

DescribeVTLDevicesOutput

- **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the ListGateways operation to return a list of gateways for your account and region.

- **VTLDevices** (*list*) –

An array of VTL device objects composed of the Amazon Resource Name(ARN) of the VTL devices.

- * *(dict)* –

Represents a device object associated with a gateway-VTL.

- **VTLDeviceARN** (*string*) –

Specifies the unique Amazon Resource Name (ARN) of the device (tape drive or media changer).

- **VTLDeviceType** (*string*) –

- **VTLDeviceVendor** (*string*) –

- **VTLDeviceProductIdentifier** (*string*) –

- **DeviceiSCSIAttributes** (*dict*) –

A list of iSCSI information about a VTL device.

- **TargetARN** (*string*) –

Specifies the unique Amazon Resource Name(ARN) that encodes the iSCSI qualified name(iqn) of a tape drive or media changer target.

- **NetworkInterfaceId** (*string*) –

The network interface identifier of the VTL device.

- **NetworkInterfacePort** (*integer*) –

The port used to communicate with iSCSI VTL device targets.

- **ChapEnabled** (*boolean*) –

Indicates whether mutual CHAP is enabled for the iSCSI target.

- **NextToken** (*string*) –

A token to resume pagination.

class StorageGateway.Paginator.**ListGateways**

```
paginator = client.get_paginator('list_gateways')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `StorageGateway.Client.list_gateways()`.

Request Syntax


```

response_iterator = paginator.paginate(
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)

```

Parameters **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```

{
    'Gateways': [
        {
            'GatewayARN': 'string',
            'GatewayType': 'string',
            'GatewayOperationalState': 'string',
            'GatewayName': 'string'
        },
    ],
    'NextToken': 'string'
}

```

Response Structure

- (*dict*) –

- **Gateways** (*list*) –

* (*dict*) –

- **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the `ListGateways` operation to return a list of gateways for your account and region.

- **GatewayType** (*string*) –

- **GatewayOperationalState** (*string*) –

- **GatewayName** (*string*) –

- **NextToken** (*string*) –

A token to resume pagination.

`class StorageGateway.Paginator.ListVolumes`

```
paginator = client.get_paginator('list_volumes')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `StorageGateway.Client.list_volumes()`.

Request Syntax

```
response_iterator = paginator.paginate(
    GatewayARN='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **GatewayARN** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the gateway. Use the `ListGateways` operation to return a list of gateways for your account and region.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns**Response Syntax**

```
{
    'GatewayARN': 'string',
    'VolumeInfos': [
        {
            'VolumeARN': 'string',
            'VolumeType': 'string'
        },
    ],
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
 - **GatewayARN** (*string*) –

The Amazon Resource Name (ARN) of the gateway. Use the `ListGateways` operation to return a list of gateways for your account and region.

- **VolumeInfos** (*list*) –
 - * (*dict*) –
 - **VolumeARN** (*string*) –
 - **VolumeType** (*string*) –
- **NextToken** (*string*) –

A token to resume pagination.

STS

Table of Contents

- STS
 - Client

Client

class STS.Client

A low-level client representing AWS Security Token Service (STS):

```
import boto3

client = boto3.client('sts')
```

These are the available methods:

- `assume_role()`
- `assume_role_with_saml()`
- `assume_role_with_web_identity()`
- `can_paginate()`
- `decode_authorization_message()`
- `generate_presigned_url()`
- `get_federation_token()`
- `get_paginator()`
- `get_session_token()`
- `get_waiter()`

assume_role (***kwargs*)

Returns a set of temporary security credentials (consisting of an access key ID, a secret access key, and a security token) that you can use to access AWS resources that you might not normally have access to. Typically, you use `AssumeRole` for cross-account access or federation.

Important: You cannot call `AssumeRole` by using AWS account credentials; access will be denied. You must use IAM user credentials or temporary security credentials to call `AssumeRole`.

For cross-account access, imagine that you own multiple accounts and need to access resources in each account. You could create long-term credentials in each account to access those resources. However, managing all those credentials and remembering which one can access which account can be time consuming. Instead, you can create one set of long-term credentials in one account and then use temporary security credentials to access all the other accounts by assuming roles in those accounts. For more information about roles, see [IAM Roles \(Delegation and Federation\)](#) in *Using IAM*.

For federation, you can, for example, grant single sign-on access to the AWS Management Console. If you already have an identity and authentication system in your corporate network, you don't have to recreate user identities in AWS in order to grant those user identities access to AWS. Instead, after a user

has been authenticated, you call `AssumeRole` (and specify the role with the appropriate permissions) to get temporary security credentials for that user. With those temporary security credentials, you construct a sign-in URL that users can use to access the console. For more information, see [Scenarios for Granting Temporary Access](#) in *Using Temporary Security Credentials*.

The temporary security credentials are valid for the duration that you specified when calling `AssumeRole`, which can be from 900 seconds (15 minutes) to 3600 seconds (1 hour). The default is 1 hour.

Optionally, you can pass an IAM access policy to this operation. If you choose not to pass a policy, the temporary security credentials that are returned by the operation have the permissions that are defined in the access policy of the role that is being assumed. If you pass a policy to this operation, the temporary security credentials that are returned by the operation have the permissions that are allowed by both the access policy of the role that is being assumed, ***and*** the policy that you pass. This gives you a way to further restrict the permissions for the resulting temporary security credentials. You cannot use the passed policy to grant permissions that are in excess of those allowed by the access policy of the role that is being assumed. For more information, see [Permissions for AssumeRole](#), [AssumeRoleWithSAML](#), and [AssumeRoleWithWebIdentity](#) in *Using Temporary Security Credentials*.

To assume a role, your AWS account must be trusted by the role. The trust relationship is defined in the role's trust policy when the role is created. You must also have a policy that allows you to call `sts:AssumeRole`.

Using MFA with AssumeRole

You can optionally include multi-factor authentication (MFA) information when you call `AssumeRole`. This is useful for cross-account scenarios in which you want to make sure that the user who is assuming the role has been authenticated using an AWS MFA device. In that scenario, the trust policy of the role being assumed includes a condition that tests for MFA authentication; if the caller does not include valid MFA information, the request to assume the role is denied. The condition in a trust policy that tests for MFA authentication might look like the following example.

```
"Condition": { "Bool": { "aws:MultiFactorAuthPresent":
true}}
```

For more information, see [Configuring MFA-Protected API Access](#) in *Using IAM* guide.

To use MFA with `AssumeRole`, you pass values for the `SerialNumber` and `TokenCode` parameters. The `SerialNumber` value identifies the user's hardware or virtual MFA device. The `TokenCode` is the time-based one-time password (TOTP) that the MFA devices produces.

Request Syntax

```
response = client.assume_role(
    RoleArn='string',
    RoleSessionName='string',
    Policy='string',
    DurationSeconds=123,
    ExternalId='string',
    SerialNumber='string',
    TokenCode='string'
)
```

Parameters

- **RoleArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the role to assume.

- **RoleSessionName** (*string*) – [REQUIRED]

An identifier for the assumed role session.

Use the role session name to uniquely identify a session when the same role is assumed by different principals or for different reasons. In cross-account sce-

narios, the role session name is visible to, and can be logged by the account that owns the role. The role session name is also used in the ARN of the assumed role principal. This means that subsequent cross-account API requests using the temporary security credentials will expose the role session name to the external account in their CloudTrail logs.

- **Policy** (*string*) – An IAM policy in JSON format.

This parameter is optional. If you pass a policy, the temporary security credentials that are returned by the operation have the permissions that are allowed by both (the intersection of) the access policy of the role that is being assumed, *and* the policy that you pass. This gives you a way to further restrict the permissions for the resulting temporary security credentials. You cannot use the passed policy to grant permissions that are in excess of those allowed by the access policy of the role that is being assumed. For more information, see [Permissions for AssumeRole](#), [AssumeRoleWithSAML](#), and [AssumeRoleWithWebIdentity](#) in *Using Temporary Security Credentials*.

Note: The policy plain text must be 2048 bytes or shorter. However, an internal conversion compresses it into a packed binary format with a separate limit. The `PackedPolicySize` response element indicates by percentage how close to the upper size limit the policy is, with 100% equaling the maximum allowed size.

- **DurationSeconds** (*integer*) – The duration, in seconds, of the role session. The value can range from 900 seconds (15 minutes) to 3600 seconds (1 hour). By default, the value is set to 3600 seconds.
- **ExternalId** (*string*) – A unique identifier that is used by third parties when assuming roles in their customers' accounts. For each role that the third party can assume, they should instruct their customers to ensure the role's trust policy checks for the external ID that the third party generated. Each time the third party assumes the role, they should pass the customer's external ID. The external ID is useful in order to help third parties bind a role to the customer who created it. For more information about the external ID, see [How to Use External ID When Granting Access to Your AWS Resources](#) in *Using Temporary Security Credentials*.
- **SerialNumber** (*string*) – The identification number of the MFA device that is associated with the user who is making the `AssumeRole` call. Specify this value if the trust policy of the role being assumed includes a condition that requires MFA authentication. The value is either the serial number for a hardware device (such as GAHT12345678) or an Amazon Resource Name (ARN) for a virtual device (such as `arn:aws:iam::123456789012:mfa/user`).
- **TokenCode** (*string*) – The value provided by the MFA device, if the trust policy of the role being assumed requires MFA (that is, if the policy includes a condition that tests for MFA). If the role being assumed requires MFA and if the `TokenCode` value is missing or expired, the `AssumeRole` call returns an "access denied" error.

Return type dict

Returns

Response Syntax

```
{
  'Credentials': {
    'AccessKeyId': 'string',
    'SecretAccessKey': 'string',
    'SessionToken': 'string',
```

```
        'Expiration': datetime(2015, 1, 1)
    },
    'AssumedRoleUser': {
        'AssumedRoleId': 'string',
        'Arn': 'string'
    },
    'PackedPolicySize': 123
}
```

Response Structure

- (*dict*) –

Contains the response to a successful AssumeRole request, including temporary AWS credentials that can be used to make AWS requests.

- **Credentials** (*dict*) –

The temporary security credentials, which include an access key ID, a secret access key, and a security (or session) token.

- * **AccessKeyId** (*string*) –

The access key ID that identifies the temporary security credentials.

- * **SecretAccessKey** (*string*) –

The secret access key that can be used to sign requests.

- * **SessionToken** (*string*) –

The token that users must pass to the service API to use the temporary credentials.

- * **Expiration** (*datetime*) –

The date on which the current credentials expire.

- **AssumedRoleUser** (*dict*) –

The Amazon Resource Name (ARN) and the assumed role ID, which are identifiers that you can use to refer to the resulting temporary security credentials. For example, you can reference these credentials as a principal in a resource-based policy by using the ARN or assumed role ID. The ARN and ID include the `RoleSessionName` that you specified when you called `AssumeRole`.

- * **AssumedRoleId** (*string*) –

A unique identifier that contains the role ID and the role session name of the role that is being assumed. The role ID is generated by AWS when the role is created.

- * **Arn** (*string*) –

The ARN of the temporary security credentials that are returned from the AssumeRole action. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in *Using IAM*.

- **PackedPolicySize** (*integer*) –

A percentage value that indicates the size of the policy in packed form. The service rejects any policy with a packed size greater than 100 percent, which means the policy exceeded the allowed space.

`assume_role_with_saml` (**kwargs)

Returns a set of temporary security credentials for users who have been authenticated via a SAML authentication response. This operation provides a mechanism for tying an enterprise identity store or directory to role-based AWS access without user-specific credentials or configuration.

The temporary security credentials returned by this operation consist of an access key ID, a secret access key, and a security token. Applications can use these temporary security credentials to sign calls to AWS services. The credentials are valid for the duration that you specified when calling `AssumeRoleWithSAML`, which can be up to 3600 seconds (1 hour) or until the time specified in the SAML authentication response's `SessionNotOnOrAfter` value, whichever is shorter.

Note: The maximum duration for a session is 1 hour, and the minimum duration is 15 minutes, even if values outside this range are specified.

Optionally, you can pass an IAM access policy to this operation. If you choose not to pass a policy, the temporary security credentials that are returned by the operation have the permissions that are defined in the access policy of the role that is being assumed. If you pass a policy to this operation, the temporary security credentials that are returned by the operation have the permissions that are allowed by both the access policy of the role that is being assumed, *and* the policy that you pass. This gives you a way to further restrict the permissions for the resulting temporary security credentials. You cannot use the passed policy to grant permissions that are in excess of those allowed by the access policy of the role that is being assumed. For more information, see [Permissions for AssumeRoleWithSAML](#) in *Using Temporary Security Credentials*.

Before your application can call `AssumeRoleWithSAML`, you must configure your SAML identity provider (IdP) to issue the claims required by AWS. Additionally, you must use AWS Identity and Access Management (IAM) to create a SAML provider entity in your AWS account that represents your identity provider, and create an IAM role that specifies this SAML provider in its trust policy.

Calling `AssumeRoleWithSAML` does not require the use of AWS security credentials. The identity of the caller is validated by using keys in the metadata document that is uploaded for the SAML provider entity for your identity provider.

For more information, see the following resources:

- [Creating Temporary Security Credentials for SAML Federation](#).
- [SAML Providers in Using IAM](#).
- [Configuring a Relying Party and Claims in Using IAM](#).
- [Creating a Role for SAML-Based Federation in Using IAM](#).

Request Syntax

```
response = client.assume_role_with_saml(
    RoleArn='string',
    PrincipalArn='string',
    SAMLAssertion='string',
    Policy='string',
    DurationSeconds=123
)
```

Parameters

- **RoleArn** (*string*) – **[REQUIRED]**
The Amazon Resource Name (ARN) of the role that the caller is assuming.
- **PrincipalArn** (*string*) – **[REQUIRED]**
The Amazon Resource Name (ARN) of the SAML provider in IAM that describes the IdP.
- **SAMLAssertion** (*string*) – **[REQUIRED]**
The base-64 encoded SAML authentication response provided by the IdP.

For more information, see [Configuring a Relying Party and Adding Claims](#) in the *Using IAM* guide.

- **Policy** (*string*) – An IAM policy in JSON format.

The policy parameter is optional. If you pass a policy, the temporary security credentials that are returned by the operation have the permissions that are allowed by both the access policy of the role that is being assumed, ***and*** the policy that you pass. This gives you a way to further restrict the permissions for the resulting temporary security credentials. You cannot use the passed policy to grant permissions that are in excess of those allowed by the access policy of the role that is being assumed. For more information, see [Permissions for AssumeRoleWithSAML](#) in *Using Temporary Security Credentials*.

Note: The policy plain text must be 2048 bytes or shorter. However, an internal conversion compresses it into a packed binary format with a separate limit. The `PackedPolicySize` response element indicates by percentage how close to the upper size limit the policy is, with 100% equaling the maximum allowed size.

-
- **DurationSeconds** (*integer*) – The duration, in seconds, of the role session. The value can range from 900 seconds (15 minutes) to 3600 seconds (1 hour). By default, the value is set to 3600 seconds. An expiration can also be specified in the SAML authentication response's `SessionNotOnOrAfter` value. The actual expiration time is whichever value is shorter.

Note: The maximum duration for a session is 1 hour, and the minimum duration is 15 minutes, even if values outside this range are specified.

Return type dict

Returns

Response Syntax

```
{
  'Credentials': {
    'AccessKeyId': 'string',
    'SecretAccessKey': 'string',
    'SessionToken': 'string',
    'Expiration': datetime(2015, 1, 1)
  },
  'AssumedRoleUser': {
    'AssumedRoleId': 'string',
    'Arn': 'string'
  },
  'PackedPolicySize': 123,
  'Subject': 'string',
  'SubjectType': 'string',
  'Issuer': 'string',
  'Audience': 'string',
  'NameQualifier': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `AssumeRoleWithSAML` request, including temporary AWS credentials that can be used to make AWS requests.

- **Credentials** (*dict*) –

AWS credentials for API authentication.

- * **AccessKeyId** (*string*) –

- The access key ID that identifies the temporary security credentials.
- * **SecretAccessKey** (*string*) –
 - The secret access key that can be used to sign requests.
- * **SessionToken** (*string*) –
 - The token that users must pass to the service API to use the temporary credentials.
- * **Expiration** (*datetime*) –
 - The date on which the current credentials expire.
- **AssumedRoleUser** (*dict*) –
 - The identifiers for the temporary security credentials that the operation returns.
 - * **AssumedRoleId** (*string*) –
 - A unique identifier that contains the role ID and the role session name of the role that is being assumed. The role ID is generated by AWS when the role is created.
 - * **Arn** (*string*) –
 - The ARN of the temporary security credentials that are returned from the AssumeRole action. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in *Using IAM*.
- **PackedPolicySize** (*integer*) –
 - A percentage value that indicates the size of the policy in packed form. The service rejects any policy with a packed size greater than 100 percent, which means the policy exceeded the allowed space.
- **Subject** (*string*) –
 - The value of the NameID element in the Subject element of the SAML assertion.
- **SubjectType** (*string*) –
 - The format of the name ID, as defined by the Format attribute in the NameID element of the SAML assertion. Typical examples of the format are transient or persistent.
 - If the format includes the prefix urn:oasis:names:tc:SAML:2.0:nameid-format, that prefix is removed. For example, urn:oasis:names:tc:SAML:2.0:nameid-format:transient is returned as transient. If the format includes any other prefix, the format is returned with no modifications.
- **Issuer** (*string*) –
 - The value of the Issuer element of the SAML assertion.
- **Audience** (*string*) –
 - The value of the Recipient attribute of the SubjectConfirmationData element of the SAML assertion.
- **NameQualifier** (*string*) –
 - A hash value based on the concatenation of the Issuer response value, the AWS account ID, and the friendly name (the last part of the ARN) of the SAML provider in IAM. The combination of NameQualifier and Subject can be used to uniquely identify a federated user.
 - The following pseudocode shows how the hash value is calculated:

```
BASE64 ( SHA1 ( "https://example.com/saml"  
+ "123456789012" + "/MySAMLIdP" ) )
```

`assume_role_with_web_identity(**kwargs)`

Returns a set of temporary security credentials for users who have been authenticated in a mobile or web application with a web identity provider, such as Amazon Cognito, Login with Amazon, Facebook, Google, or any OpenID Connect-compatible identity provider.

Note: For mobile applications, we recommend that you use Amazon Cognito. You can use Amazon Cognito with the [AWS SDK for iOS](#) and the [AWS SDK for Android](#) to uniquely identify a user and supply the user with a consistent identity throughout the lifetime of an application.

To learn more about Amazon Cognito, see [Amazon Cognito Overview](#) in the *AWS SDK for Android Developer Guide* guide and [Amazon Cognito Overview](#) in the *AWS SDK for iOS Developer Guide* .

Calling `AssumeRoleWithWebIdentity` does not require the use of AWS security credentials. Therefore, you can distribute an application (for example, on mobile devices) that requests temporary security credentials without including long-term AWS credentials in the application, and without deploying server-based proxy services that use long-term AWS credentials. Instead, the identity of the caller is validated by using a token from the web identity provider.

The temporary security credentials returned by this API consist of an access key ID, a secret access key, and a security token. Applications can use these temporary security credentials to sign calls to AWS service APIs. The credentials are valid for the duration that you specified when calling `AssumeRoleWithWebIdentity` , which can be from 900 seconds (15 minutes) to 3600 seconds (1 hour). By default, the temporary security credentials are valid for 1 hour.

Optionally, you can pass an IAM access policy to this operation. If you choose not to pass a policy, the temporary security credentials that are returned by the operation have the permissions that are defined in the access policy of the role that is being assumed. If you pass a policy to this operation, the temporary security credentials that are returned by the operation have the permissions that are allowed by both the access policy of the role that is being assumed, ***and*** the policy that you pass. This gives you a way to further restrict the permissions for the resulting temporary security credentials. You cannot use the passed policy to grant permissions that are in excess of those allowed by the access policy of the role that is being assumed. For more information, see [Permissions for AssumeRoleWithWebIdentity](#) .

Before your application can call `AssumeRoleWithWebIdentity` , you must have an identity token from a supported identity provider and create a role that the application can assume. The role that your application assumes must trust the identity provider that is associated with the identity token. In other words, the identity provider must be specified in the role's trust policy.

For more information about how to use web identity federation and the `AssumeRoleWithWebIdentity` API, see the following resources:

- [Creating a Mobile Application with Third-Party Sign-In and Creating Temporary Security Credentials for Mobile Apps Using Third-Party Identity Providers](#) .
- [Web Identity Federation Playground](#) . This interactive website lets you walk through the process of authenticating via Login with Amazon, Facebook, or Google, getting temporary security credentials, and then using those credentials to make a request to AWS.
- [AWS SDK for iOS and AWS SDK for Android](#) . These toolkits contain sample apps that show how to invoke the identity providers, and then how to use the information from these providers to get and use temporary security credentials.
- [Web Identity Federation with Mobile Applications](#) . This article discusses web identity federation and shows an example of how to use web identity federation to get access to content in Amazon S3.

Request Syntax

```
response = client.assume_role_with_web_identity(  
    RoleArn='string',
```

```

RoleSessionName='string',
WebIdentityToken='string',
ProviderId='string',
Policy='string',
DurationSeconds=123
)

```

Parameters

- **RoleArn** (*string*) – [REQUIRED]

The Amazon Resource Name (ARN) of the role that the caller is assuming.

- **RoleSessionName** (*string*) – [REQUIRED]

An identifier for the assumed role session. Typically, you pass the name or identifier that is associated with the user who is using your application. That way, the temporary security credentials that your application will use are associated with that user. This session name is included as part of the ARN and assumed role ID in the `AssumedRoleUser` response element.

- **WebIdentityToken** (*string*) – [REQUIRED]

The OAuth 2.0 access token or OpenID Connect ID token that is provided by the identity provider. Your application must get this token by authenticating the user who is using your application with a web identity provider before the application makes an `AssumeRoleWithWebIdentity` call.

- **ProviderId** (*string*) – The fully qualified host component of the domain name of the identity provider.

Specify this value only for OAuth 2.0 access tokens. Currently `www.amazon.com` and `graph.facebook.com` are the only supported identity providers for OAuth 2.0 access tokens. Do not include URL schemes and port numbers.

Do not specify this value for OpenID Connect ID tokens.

- **Policy** (*string*) – An IAM policy in JSON format.

The policy parameter is optional. If you pass a policy, the temporary security credentials that are returned by the operation have the permissions that are allowed by both the access policy of the role that is being assumed, ***and*** the policy that you pass. This gives you a way to further restrict the permissions for the resulting temporary security credentials. You cannot use the passed policy to grant permissions that are in excess of those allowed by the access policy of the role that is being assumed. For more information, see [Permissions for AssumeRoleWithWebIdentity](#).

Note: The policy plain text must be 2048 bytes or shorter. However, an internal conversion compresses it into a packed binary format with a separate limit. The `PackedPolicySize` response element indicates by percentage how close to the upper size limit the policy is, with 100% equaling the maximum allowed size.

- **DurationSeconds** (*integer*) – The duration, in seconds, of the role session. The value can range from 900 seconds (15 minutes) to 3600 seconds (1 hour). By default, the value is set to 3600 seconds.

Return type dict

Returns

Response Syntax

```
{
  'Credentials': {
    'AccessKeyId': 'string',
    'SecretAccessKey': 'string',
    'SessionToken': 'string',
    'Expiration': datetime(2015, 1, 1)
  },
  'SubjectFromWebIdentityToken': 'string',
  'AssumedRoleUser': {
    'AssumedRoleId': 'string',
    'Arn': 'string'
  },
  'PackedPolicySize': 123,
  'Provider': 'string',
  'Audience': 'string'
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `AssumeRoleWithWebIdentity` request, including temporary AWS credentials that can be used to make AWS requests.

- **Credentials** (*dict*) –

The temporary security credentials, which include an access key ID, a secret access key, and a security token.

- * **AccessKeyId** (*string*) –

The access key ID that identifies the temporary security credentials.

- * **SecretAccessKey** (*string*) –

The secret access key that can be used to sign requests.

- * **SessionToken** (*string*) –

The token that users must pass to the service API to use the temporary credentials.

- * **Expiration** (*datetime*) –

The date on which the current credentials expire.

- **SubjectFromWebIdentityToken** (*string*) –

The unique user identifier that is returned by the identity provider. This identifier is associated with the `WebIdentityToken` that was submitted with the `AssumeRoleWithWebIdentity` call. The identifier is typically unique to the user and the application that acquired the `WebIdentityToken` (pairwise identifier). For OpenID Connect ID tokens, this field contains the value returned by the identity provider as the token's `sub` (Subject) claim.

- **AssumedRoleUser** (*dict*) –

The Amazon Resource Name (ARN) and the assumed role ID, which are identifiers that you can use to refer to the resulting temporary security credentials. For example, you can reference these credentials as a principal in a resource-based policy by using the ARN or assumed role ID. The ARN and ID include the `RoleSessionName` that you specified when you called `AssumeRole`.

- * **AssumedRoleId** (*string*) –

A unique identifier that contains the role ID and the role session

name of the role that is being assumed. The role ID is generated by AWS when the role is created.

* **Arn** (*string*) –

The ARN of the temporary security credentials that are returned from the AssumeRole action. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in *Using IAM*.

– **PackedPolicySize** (*integer*) –

A percentage value that indicates the size of the policy in packed form. The service rejects any policy with a packed size greater than 100 percent, which means the policy exceeded the allowed space.

– **Provider** (*string*) –

The issuing authority of the web identity token presented. For OpenID Connect ID Tokens this contains the value of the `iss` field. For OAuth 2.0 access tokens, this contains the value of the `ProviderId` parameter that was passed in the AssumeRoleWithWebIdentity request.

– **Audience** (*string*) –

The intended audience (also known as client ID) of the web identity token. This is traditionally the client identifier issued to the application that requested the web identity token.

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

decode_authorization_message (***kwargs*)

Decodes additional information about the authorization status of a request from an encoded message returned in response to an AWS request.

For example, if a user is not authorized to perform an action that he or she has requested, the request returns a `Client.UnauthorizedOperation` response (an HTTP 403 response). Some AWS actions additionally return an encoded message that can provide details about this authorization failure.

Note: Only certain AWS actions return an encoded authorization message. The documentation for an individual action indicates whether that action returns an encoded message in addition to returning an HTTP code.

The message is encoded because the details of the authorization status can constitute privileged information that the user who requested the action should not see. To decode an authorization status message, a user must be granted permissions via an IAM policy to request the `DecodeAuthorizationMessage` (`sts:DecodeAuthorizationMessage`) action.

The decoded message includes the following type of information:

- Whether the request was denied due to an explicit deny or due to the absence of an explicit allow. For more information, see [Determining Whether a Request is Allowed or Denied](#) in *Using IAM*.
- The principal who made the request.
- The requested action.
- The requested resource.
- The values of condition keys in the context of the user's request.

Request Syntax

```
response = client.decode_authorization_message(  
    EncodedMessage='string'  
)
```

Parameters `EncodedMessage` (*string*) – [REQUIRED]

The encoded message that was returned with the response.

Return type dict

Returns

Response Syntax

```
{  
    'DecodedMessage': 'string'  
}
```

Response Structure

- (*dict*) –

A document that contains additional information about the authorization status of a request from an encoded message that is returned in response to an AWS request.

- **DecodedMessage** (*string*) –

An XML document that contains the decoded message. For more information, see `DecodeAuthorizationMessage`.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_federation_token (***kwargs*)

Returns a set of temporary security credentials (consisting of an access key ID, a secret access key, and a security token) for a federated user. A typical use is in a proxy application that gets temporary security credentials on behalf of distributed applications inside a corporate network. Because you must call the `GetFederationToken` action using the long-term security credentials of an IAM user, this call is appropriate in contexts where those credentials can be safely stored, usually in a server-based application.

Note: If you are creating a mobile-based or browser-based app that can authenticate users using a web identity provider like Login with Amazon, Facebook, Google, or an OpenID Connect-compatible identity provider, we recommend that you use [Amazon Cognito](#) or `AssumeRoleWithWebIdentity`. For more information, see [Creating Temporary Security Credentials for Mobile Apps Using Identity Providers](#).

The `GetFederationToken` action must be called by using the long-term AWS security credentials of an IAM user. You can also call `GetFederationToken` using the security credentials of an AWS account (root), but this is not recommended. Instead, we recommend that you create an IAM user for the purpose of the proxy application and then attach a policy to the IAM user that limits federated users to only the actions and resources they need access to. For more information, see [IAM Best Practices in Using IAM](#).

The temporary security credentials that are obtained by using the long-term credentials of an IAM user are valid for the specified duration, between 900 seconds (15 minutes) and 129600 seconds (36 hours). Temporary credentials that are obtained by using AWS account (root) credentials have a maximum duration of 3600 seconds (1 hour)

Permissions

The permissions for the temporary security credentials returned by `GetFederationToken` are determined by a combination of the following:

- The policy or policies that are attached to the IAM user whose credentials are used to call `GetFederationToken`.
- The policy that is passed as a parameter in the call.

The passed policy is attached to the temporary security credentials that result from the `GetFederationToken` API call—that is, to the *federated user*. When the federated user makes an AWS request, AWS evaluates the policy attached to the federated user in combination with the policy or policies attached to the IAM user whose credentials were used to call `GetFederationToken`. AWS allows the federated user's request only when both the federated user **and** the IAM user are explicitly allowed to perform the requested action. The passed policy cannot grant more permissions than those that are defined in the IAM user policy.

A typical use case is that the permissions of the IAM user whose credentials are used to call `GetFederationToken` are designed to allow access to all the actions and resources that any federated user will need. Then, for individual users, you pass a policy to the operation that scopes down the permissions to a level that's appropriate to that individual user, using a policy that allows only a subset of permissions that are granted to the IAM user.

If you do not pass a policy, the resulting temporary security credentials have no effective permissions. The only exception is when the temporary security credentials are used to access a resource that has a resource-based policy that specifically allows the federated user to access the resource.

For more information about how permissions work, see [Permissions for GetFederationToken](#). For information about using `GetFederationToken` to create temporary security credentials, see [Creating Temporary Credentials to Enable Access for Federated Users](#).

Request Syntax

```
response = client.get_federation_token(
    Name='string',
    Policy='string',
    DurationSeconds=123
)
```

Parameters

- **Name** (*string*) – [REQUIRED]

The name of the federated user. The name is used as an identifier for the temporary security credentials (such as `Bob`). For example, you can reference the federated user name in a resource-based policy, such as in an Amazon S3 bucket policy.

- **Policy** (*string*) – An IAM policy in JSON format that is passed with the `GetFederationToken` call and evaluated along with the policy or policies that are attached to the IAM user whose credentials are used to call `GetFederationToken`. The passed policy is used to scope down the permissions that are available to the IAM user, by allowing only a subset of the permissions that are granted to the IAM user. The passed policy cannot grant more permissions than those granted to the IAM user. The final permissions for the federated user are the most restrictive set based on the intersection of the passed policy and the IAM user policy.

If you do not pass a policy, the resulting temporary security credentials have no effective permissions. The only exception is when the temporary security credentials are used to access a resource that has a resource-based policy that specifically allows the federated user to access the resource.

Note: The policy plain text must be 2048 bytes or shorter. However, an internal conversion compresses it into a packed binary format with a separate limit. The `PackedPolicySize` response element indicates by percentage how close to the upper size limit the policy is, with 100% equaling the maximum allowed size.

For more information about how permissions work, see [Permissions for GetFederationToken](#).

- **DurationSeconds** (*integer*) – The duration, in seconds, that the session should last. Acceptable durations for federation sessions range from 900 seconds (15 minutes) to 129600 seconds (36 hours), with 43200 seconds (12 hours) as the default. Sessions obtained using AWS account (root) credentials are restricted to a maximum of 3600 seconds (one hour). If the specified duration is longer than one hour, the session obtained by using AWS account (root) credentials defaults to one hour.

Return type dict

Returns

Response Syntax

```
{
  'Credentials': {
    'AccessKeyId': 'string',
    'SecretAccessKey': 'string',
    'SessionToken': 'string',
    'Expiration': datetime(2015, 1, 1)
  },
  'FederatedUser': {
    'FederatedUserId': 'string',
    'Arn': 'string'
  },
  'PackedPolicySize': 123
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `GetFederationToken` request, including temporary AWS credentials that can be used to make AWS requests.

- **Credentials** (*dict*) –

Credentials for the service API authentication.

- * **AccessKeyId** (*string*) –

The access key ID that identifies the temporary security credentials.

- * **SecretAccessKey** (*string*) –

The secret access key that can be used to sign requests.

- * **SessionToken** (*string*) –

The token that users must pass to the service API to use the temporary credentials.

- * **Expiration** (*datetime*) –

The date on which the current credentials expire.

– **FederatedUser** (*dict*) –

Identifiers for the federated user associated with the credentials (such as `arn:aws:sts::123456789012:federated-user/Bob` or `123456789012:Bob`). You can use the federated user's ARN in your resource-based policies, such as an Amazon S3 bucket policy.

* **FederatedUserId** (*string*) –

The string that identifies the federated user associated with the credentials, similar to the unique ID of an IAM user.

* **Arn** (*string*) –

The ARN that specifies the federated user that is associated with the credentials. For more information about ARNs and how to use them in policies, see [IAM Identifiers](#) in *Using IAM*.

– **PackedPolicySize** (*integer*) –

A percentage value indicating the size of the policy in packed form. The service rejects policies for which the packed size is greater than 100 percent of the allowed value.

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_session_token (***kwargs*)

Returns a set of temporary credentials for an AWS account or IAM user. The credentials consist of an access key ID, a secret access key, and a security token. Typically, you use `GetSessionToken` if you want to use MFA to protect programmatic calls to specific AWS APIs like Amazon EC2 `StopInstances`. MFA-enabled IAM users would need to call `GetSessionToken` and submit an MFA code that is associated with their MFA device. Using the temporary security credentials that are returned from the call, IAM users can then make programmatic calls to APIs that require MFA authentication.

The `GetSessionToken` action must be called by using the long-term AWS security credentials of the AWS account or an IAM user. Credentials that are created by IAM users are valid for the duration that you specify, between 900 seconds (15 minutes) and 129600 seconds (36 hours); credentials that are created by using account credentials have a maximum duration of 3600 seconds (1 hour).

Note: We recommend that you do not call `GetSessionToken` with root account credentials. Instead, follow our [best practices](#) by creating one or more IAM users, giving them the necessary permissions, and using IAM users for everyday interaction with AWS.

The permissions associated with the temporary security credentials returned by `GetSessionToken` are based on the permissions associated with account or IAM user whose credentials are used to call the action. If `GetSessionToken` is called using root account credentials, the temporary credentials have root account permissions. Similarly, if `GetSessionToken` is called using the credentials of an IAM user, the temporary credentials have the same permissions as the IAM user.

For more information about using `GetSessionToken` to create temporary credentials, go to [Creating Temporary Credentials to Enable Access for IAM Users](#).

Request Syntax

```
response = client.get_session_token(
    DurationSeconds=123,
    SerialNumber='string',
    TokenCode='string'
)
```

Parameters

- **DurationSeconds** (*integer*) – The duration, in seconds, that the credentials should remain valid. Acceptable durations for IAM user sessions range from 900 seconds (15 minutes) to 129600 seconds (36 hours), with 43200 seconds (12 hours) as the default. Sessions for AWS account owners are restricted to a maximum of 3600 seconds (one hour). If the duration is longer than one hour, the session for AWS account owners defaults to one hour.
- **SerialNumber** (*string*) – The identification number of the MFA device that is associated with the IAM user who is making the `GetSessionToken` call. Specify this value if the IAM user has a policy that requires MFA authentication. The value is either the serial number for a hardware device (such as `GAHT12345678`) or an Amazon Resource Name (ARN) for a virtual device (such as `arn:aws:iam::123456789012:mfa/user`). You can find the device for an IAM user by going to the AWS Management Console and viewing the user's security credentials.
- **TokenCode** (*string*) – The value provided by the MFA device, if MFA is required. If any policy requires the IAM user to submit an MFA code, specify this value. If MFA authentication is required, and the user does not provide a code when requesting a set of temporary security credentials, the user will receive an “access denied” response when requesting resources that require MFA authentication.

Return type dict

Returns

Response Syntax

```
{
  'Credentials': {
    'AccessKeyId': 'string',
    'SecretAccessKey': 'string',
    'SessionToken': 'string',
    'Expiration': datetime(2015, 1, 1)
  }
}
```

Response Structure

- (*dict*) –

Contains the response to a successful `GetSessionToken` request, including temporary AWS credentials that can be used to make AWS requests.

- **Credentials** (*dict*) –

The session credentials for API authentication.

- * **AccessKeyId** (*string*) –

The access key ID that identifies the temporary security credentials.

* **SecretAccessKey** (*string*) –

The secret access key that can be used to sign requests.

* **SessionToken** (*string*) –

The token that users must pass to the service API to use the temporary credentials.

* **Expiration** (*datetime*) –

The date on which the current credentials expire.

get_waiter (*waiter_name*)

Support

Table of Contents

- Support
 - Client
 - Paginators

Client

class `Support.Client`

A low-level client representing AWS Support:

```
import boto3
```

```
client = boto3.client('support')
```

These are the available methods:

- `add_attachments_to_set()`
- `add_communication_to_case()`
- `can_paginate()`
- `create_case()`
- `describe_attachment()`
- `describe_cases()`
- `describe_communications()`
- `describe_services()`
- `describe_severity_levels()`
- `describe_trusted_advisor_check_refresh_statuses()`
- `describe_trusted_advisor_check_result()`
- `describe_trusted_advisor_check_summaries()`
- `describe_trusted_advisor_checks()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_waiter()`
- `refresh_trusted_advisor_check()`
- `resolve_case()`

add_attachments_to_set (***kwargs*)

Adds one or more attachments to an attachment set. If an `AttachmentSetId` is not specified, a new attachment set is created, and the ID of the set is returned in the response. If an `AttachmentSetId` is specified, the attachments are added to the specified set, if it exists.

An attachment set is a temporary container for attachments that are to be added to a case or case communication. The set is available for one hour after it is created; the `ExpiryTime` returned in the response indicates when the set expires. The maximum number of attachments in a set is 3, and the maximum size of any attachment in the set is 5 MB.

Request Syntax

```
response = client.add_attachments_to_set(
    attachmentSetId='string',
    attachments=[
        {
            'fileName': 'string',
            'data': b'bytes'
        },
    ],
)
```

Parameters

- **attachmentSetId** (*string*) – The ID of the attachment set. If an `AttachmentSetId` is not specified, a new attachment set is created, and the ID of the set is returned in the response. If an `AttachmentSetId` is specified, the attachments are added to the specified set, if it exists.
- **attachments** (*list*) – **[REQUIRED]**

One or more attachments to add to the set. The limit is 3 attachments per set, and the size limit is 5 MB per attachment.

– (*dict*) –

An attachment to a case communication. The attachment consists of the file name and the content of the file.

* **fileName** (*string*) –

The name of the attachment file.

* **data** (*bytes*) –

The content of the attachment file.

Return type dict

Returns

Response Syntax

```
{
    'attachmentSetId': 'string',
    'expiryTime': 'string'
}
```

Response Structure

- (*dict*) –

The ID and expiry time of the attachment set returned by the `AddAttachmentsToSet` operation.

– **attachmentSetId** (*string*) –

The ID of the attachment set. If an `AttachmentSetId` was not specified, a new attachment set is created, and the ID of the set is returned in the response. If an `AttachmentSetId` was specified, the attachments are added to the specified set, if it exists.

– **expiryTime** (*string*) –

The time and date when the attachment set expires.

add_communication_to_case (**kwargs)

Adds additional customer communication to an AWS Support case. You use the `CaseId` value to identify the case to add communication to. You can list a set of email addresses to copy on the communication using the `CcEmailAddresses` value. The `CommunicationBody` value contains the text of the communication.

The response indicates the success or failure of the request.

This operation implements a subset of the features of the AWS Support Center.

Request Syntax

```
response = client.add_communication_to_case(
    caseId='string',
    communicationBody='string',
    ccEmailAddresses=[
        'string',
    ],
    attachmentSetId='string'
)
```

Parameters

- **caseId** (*string*) – The AWS Support case ID requested or returned in the call. The case ID is an alphanumeric string formatted as shown in this example: `case-12345678910-2013-c4c1d2bf33c5cf47`
- **communicationBody** (*string*) – **[REQUIRED]**
The body of an email communication to add to the support case.
- **ccEmailAddresses** (*list*) – The email addresses in the CC line of an email to be added to the support case.
– (*string*) –
- **attachmentSetId** (*string*) – The ID of a set of one or more attachments for the communication to add to the case. Create the set by calling `AddAttachmentsToSet`

Return type dict

Returns**Response Syntax**

```
{
    'result': True|False
}
```

Response Structure

- (*dict*) –
The result of the `AddCommunicationToCase` operation.
– **result** (*boolean*) –
True if `AddCommunicationToCase` succeeds. Otherwise, returns an error.

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

create_case (**kwargs)

Creates a new case in the AWS Support Center. This operation is modeled on the behavior of the AWS Support Center [Create Case](#) page. Its parameters require you to specify the following information:

- **IssueType.** The type of issue for the case. You can specify either “customer-service” or “technical.” If you do not indicate a value, the default is “technical.”
- **ServiceCode.** The code for an AWS service. You obtain the `ServiceCode` by calling `DescribeServices`.
- **CategoryCode.** The category for the service defined for the `ServiceCode` value. You also obtain the category code for a service by calling `DescribeServices`. Each AWS service defines its own set of category codes.
- **SeverityCode.** A value that indicates the urgency of the case, which in turn determines the response time according to your service level agreement with AWS Support. You obtain the `SeverityCode` by calling `DescribeSeverityLevels`.
- **Subject.** The **Subject** field on the AWS Support Center [Create Case](#) page.
- **CommunicationBody.** The **Description** field on the AWS Support Center [Create Case](#) page.
- **AttachmentSetId.** The ID of a set of attachments that has been created by using `AddAttachmentsToSet`.
- **Language.** The human language in which AWS Support handles the case. English and Japanese are currently supported.
- **CcEmailAddresses.** The AWS Support Center **CC** field on the [Create Case](#) page. You can list email addresses to be copied on any correspondence about the case. The account that opens the case is already identified by passing the AWS Credentials in the HTTP POST method or in a method or function call from one of the programming languages supported by an [AWS SDK](#).

Note: To add additional communication or attachments to an existing case, use `AddCommunicationToCase`.

A successful `CreateCase` request returns an AWS Support case number. Case numbers are used by the `DescribeCases` operation to retrieve existing AWS Support cases.

Request Syntax

```
response = client.create_case(
    subject='string',
    serviceCode='string',
    severityCode='string',
    categoryCode='string',
    communicationBody='string',
    ccEmailAddresses=[
        'string',
    ],
    language='string',
    issueType='string',
    attachmentSetId='string'
)
```

Parameters

- **subject** (*string*) – **[REQUIRED]**
The title of the AWS Support case.
- **serviceCode** (*string*) – The code for the AWS service returned by the call to `DescribeServices`.
- **severityCode** (*string*) – The code for the severity level returned by the call to `DescribeSeverityLevels`.

Note: The availability of severity levels depends on each customer’s support

subscription. In other words, your subscription may not necessarily require the urgent level of response time.

- **categoryCode** (*string*) – The category of problem for the AWS Support case.
- **communicationBody** (*string*) – **[REQUIRED]**
The communication body text when you create an AWS Support case by calling `CreateCase`.
- **ccEmailAddresses** (*list*) – A list of email addresses that AWS Support copies on case correspondence.
– (*string*) –
- **language** (*string*) – The ISO 639-1 code for the language in which AWS provides support. AWS Support currently supports English (“en”) and Japanese (“ja”). Language parameters must be passed explicitly for operations that take them.
- **issueType** (*string*) – The type of issue for the case. You can specify either “customer-service” or “technical.” If you do not indicate a value, the default is “technical.”
- **attachmentSetId** (*string*) – The ID of a set of one or more attachments for the case. Create the set by using `AddAttachmentsToSet`.

Return type dict

Returns

Response Syntax

```
{
    'caseId': 'string'
}
```

Response Structure

- (*dict*) –

The AWS Support case ID returned by a successful completion of the `CreateCase` operation.

- **caseId** (*string*) –

The AWS Support case ID requested or returned in the call. The case ID is an alphanumeric string formatted as shown in this example: case-12345678910-2013-c4c1d2bf33c5cf47

describe_attachment (***kwargs*)

Returns the attachment that has the specified ID. Attachment IDs are generated by the case management system when you add an attachment to a case or case communication. Attachment IDs are returned in the `AttachmentDetails` objects that are returned by the `DescribeCommunications` operation.

Request Syntax

```
response = client.describe_attachment(
    attachmentId='string'
)
```

Parameters **attachmentId** (*string*) – **[REQUIRED]**

The ID of the attachment to return. Attachment IDs are returned by the `DescribeCommunications` operation.

Return type dict

Returns

Response Syntax

```
{
    'attachment': {
        'fileName': 'string',
        'data': b'bytes'
    }
}
```

Response Structure

- *(dict)* –

The content and file name of the attachment returned by the DescribeAttachment operation.

- **attachment** (*dict*) –

The attachment content and file name.

- * **fileName** (*string*) –

The name of the attachment file.

- * **data** (*bytes*) –

The content of the attachment file.

describe_cases (***kwargs*)

Returns a list of cases that you specify by passing one or more case IDs. In addition, you can filter the cases by date by setting values for the `AfterTime` and `BeforeTime` request parameters. You can set values for the `IncludeResolvedCases` and `IncludeCommunications` request parameters to control how much information is returned.

Case data is available for 12 months after creation. If a case was created more than 12 months ago, a request for data might cause an error.

The response returns the following in JSON format:

- One or more `CaseDetails` data types.
- One or more `NextToken` values, which specify where to paginate the returned records represented by the `CaseDetails` objects.

Request Syntax

```
response = client.describe_cases(
    caseIdList=[
        'string',
    ],
    displayId='string',
    afterTime='string',
    beforeTime='string',
    includeResolvedCases=True|False,
    nextToken='string',
    maxResults=123,
    language='string',
    includeCommunications=True|False
)
```

Parameters

- **caseIdList** (*list*) – A list of ID numbers of the support cases you want returned. The maximum number of cases is 100.
 - (*string*) –
- **displayId** (*string*) – The ID displayed for a case in the AWS Support Center user interface.

- **afterTime** (*string*) – The start date for a filtered date search on support case communications. Case communications are available for 12 months after creation.
- **beforeTime** (*string*) – The end date for a filtered date search on support case communications. Case communications are available for 12 months after creation.
- **includeResolvedCases** (*boolean*) – Specifies whether resolved support cases should be included in the DescribeCases results. The default is *false*.
- **nextToken** (*string*) – A resumption point for pagination.
- **maxResults** (*integer*) – The maximum number of results to return before paginating.
- **language** (*string*) – The ISO 639-1 code for the language in which AWS provides support. AWS Support currently supports English (“en”) and Japanese (“ja”). Language parameters must be passed explicitly for operations that take them.
- **includeCommunications** (*boolean*) – Specifies whether communications should be included in the DescribeCases results. The default is *true*.

Return type dict

Returns

Response Syntax

```
{
  'cases': [
    {
      'caseId': 'string',
      'displayId': 'string',
      'subject': 'string',
      'status': 'string',
      'serviceCode': 'string',
      'categoryCode': 'string',
      'severityCode': 'string',
      'submittedBy': 'string',
      'timeCreated': 'string',
      'recentCommunications': {
        'communications': [
          {
            'caseId': 'string',
            'body': 'string',
            'submittedBy': 'string',
            'timeCreated': 'string',
            'attachmentSet': [
              {
                'attachmentId': 'string',
                'fileName': 'string'
              }
            ]
          }
        ]
      },
      'nextToken': 'string'
    },
    'ccEmailAddresses': [
      'string',
    ],
    'language': 'string'
  ],
  'nextToken': 'string'
}
```

Response Structure

- (*dict*) –

Returns an array of `CaseDetails` objects and a `NextToken` that defines a point for pagination in the result set.

- **cases** (*list*) –

The details for the cases that match the request.

- * (*dict*) –

A JSON-formatted object that contains the metadata for a support case. It is contained the response from a `DescribeCases` request.

CaseDetails contains the following fields:

- **CaseID**. The AWS Support case ID requested or returned in the call. The case ID is an alphanumeric string formatted as shown in this example: `case-12345678910-2013-c4c1d2bf33c5cf47`.
- **CategoryCode**. The category of problem for the AWS Support case. Corresponds to the `CategoryCode` values returned by a call to `DescribeServices`.
- **DisplayId**. The identifier for the case on pages in the AWS Support Center.
- **Language**. The ISO 639-1 code for the language in which AWS provides support. AWS Support currently supports English (“en”) and Japanese (“ja”). Language parameters must be passed explicitly for operations that take them.
- **RecentCommunications**. One or more `Communication` objects. Fields of these objects are `Attachments`, `Body`, `CaseId`, `SubmittedBy`, and `TimeCreated`.
- **NextToken**. A resumption point for pagination.
- **ServiceCode**. The identifier for the AWS service that corresponds to the service code defined in the call to `DescribeServices`.
- **SeverityCode**. The severity code assigned to the case. Contains one of the values returned by the call to `DescribeSeverityLevels`.
- **Status**. The status of the case in the AWS Support Center.
- **Subject**. The subject line of the case.
- **SubmittedBy**. The email address of the account that submitted the case.
- **TimeCreated**. The time the case was created, in ISO-8601 format.
- **caseId** (*string*) –
The AWS Support case ID requested or returned in the call. The case ID is an alphanumeric string formatted as shown in this example: `case-12345678910-2013-c4c1d2bf33c5cf47`
- **displayId** (*string*) –
The ID displayed for the case in the AWS Support Center. This is a numeric string.
- **subject** (*string*) –
The subject line for the case in the AWS Support Center.
- **status** (*string*) –
The status of the case.

- **serviceCode** (*string*) –
The code for the AWS service returned by the call to `DescribeServices`.
- **categoryCode** (*string*) –
The category of problem for the AWS Support case.
- **severityCode** (*string*) –
The code for the severity level returned by the call to `DescribeSeverityLevels`.
- **submittedBy** (*string*) –
The email address of the account that submitted the case.
- **timeCreated** (*string*) –
The time that the case was created in the AWS Support Center.
- **recentCommunications** (*dict*) –
The five most recent communications between you and AWS Support Center, including the IDs of any attachments to the communications. Also includes a `nextToken` that you can use to retrieve earlier communications.
- **communications** (*list*) –
The five most recent communications associated with the case.
- (*dict*) –
A communication associated with an AWS Support case. The communication consists of the case ID, the message body, attachment information, the account email address, and the date and time of the communication.
- **caseId** (*string*) –
The AWS Support case ID requested or returned in the call. The case ID is an alphanumeric string formatted as shown in this example: `case-12345678910-2013-c4c1d2bf33c5cf47`
- **body** (*string*) –
The text of the communication between the customer and AWS Support.
- **submittedBy** (*string*) –
The email address of the account that submitted the AWS Support case.
- **timeCreated** (*string*) –
The time the communication was created.
- **attachmentSet** (*list*) –
Information about the attachments to the case communication.
- (*dict*) –
The file name and ID of an attachment to a case communication. You can use the ID to retrieve the attachment with the `DescribeAttachment` operation.
- **attachmentId** (*string*) –

The ID of the attachment.

- **fileName** (*string*) –

The file name of the attachment.

- **nextToken** (*string*) –

A resumption point for pagination.

- **ccEmailAddresses** (*list*) –

The email addresses that receive copies of communication about the case.

- (*string*) –

- **language** (*string*) –

The ISO 639-1 code for the language in which AWS provides support. AWS Support currently supports English (“en”) and Japanese (“ja”). Language parameters must be passed explicitly for operations that take them.

- **nextToken** (*string*) –

A resumption point for pagination.

describe_communications (***kwargs*)

Returns communications (and attachments) for one or more support cases. You can use the `AfterTime` and `BeforeTime` parameters to filter by date. You can use the `CaseId` parameter to restrict the results to a particular case.

Case data is available for 12 months after creation. If a case was created more than 12 months ago, a request for data might cause an error.

You can use the `MaxResults` and `NextToken` parameters to control the pagination of the result set. Set `MaxResults` to the number of cases you want displayed on each page, and use `NextToken` to specify the resumption of pagination.

Request Syntax

```
response = client.describe_communications(  
    caseId='string',  
    beforeTime='string',  
    afterTime='string',  
    nextToken='string',  
    maxResults=123  
)
```

Parameters

- **caseId** (*string*) – [REQUIRED]

The AWS Support case ID requested or returned in the call. The case ID is an alphanumeric string formatted as shown in this example: `case-12345678910-2013-c4c1d2bf33c5cf47`

- **beforeTime** (*string*) – The end date for a filtered date search on support case communications. Case communications are available for 12 months after creation.
- **afterTime** (*string*) – The start date for a filtered date search on support case communications. Case communications are available for 12 months after creation.
- **nextToken** (*string*) – A resumption point for pagination.
- **maxResults** (*integer*) – The maximum number of results to return before paginating.

Return type dict

Returns

Response Syntax

```
{
  'communications': [
    {
      'caseId': 'string',
      'body': 'string',
      'submittedBy': 'string',
      'timeCreated': 'string',
      'attachmentSet': [
        {
          'attachmentId': 'string',
          'fileName': 'string'
        }
      ],
    },
  ],
  'nextToken': 'string'
}
```

Response Structure

- (*dict*) –

The communications returned by the DescribeCommunications operation.

- **communications** (*list*) –

The communications for the case.

- * (*dict*) –

A communication associated with an AWS Support case. The communication consists of the case ID, the message body, attachment information, the account email address, and the date and time of the communication.

- **caseId** (*string*) –

The AWS Support case ID requested or returned in the call. The case ID is an alphanumeric string formatted as shown in this example: case-12345678910-2013-c4c1d2bf33c5cf47

- **body** (*string*) –

The text of the communication between the customer and AWS Support.

- **submittedBy** (*string*) –

The email address of the account that submitted the AWS Support case.

- **timeCreated** (*string*) –

The time the communication was created.

- **attachmentSet** (*list*) –

Information about the attachments to the case communication.

- (*dict*) –

The file name and ID of an attachment to a case communication. You can use the ID to retrieve the attachment with the DescribeAttachment operation.

- **attachmentId** (*string*) –
The ID of the attachment.
- **fileName** (*string*) –
The file name of the attachment.
- **nextToken** (*string*) –
A resumption point for pagination.

describe_services (***kwargs*)

Returns the current list of AWS services and a list of service categories that applies to each one. You then use service names and categories in your CreateCase requests. Each AWS service has its own set of categories.

The service codes and category codes correspond to the values that are displayed in the **Service** and **Category** drop-down lists on the AWS Support Center [Create Case](#) page. The values in those fields, however, do not necessarily match the service codes and categories returned by the `DescribeServices` request. Always use the service codes and categories obtained programmatically. This practice ensures that you always have the most recent set of service and category codes.

Request Syntax

```
response = client.describe_services(  
    serviceCodeList=[  
        'string',  
    ],  
    language='string'  
)
```

Parameters

- **serviceCodeList** (*list*) – A JSON-formatted list of service codes available for AWS services.
 - (*string*) –
- **language** (*string*) – The ISO 639-1 code for the language in which AWS provides support. AWS Support currently supports English (“en”) and Japanese (“ja”). Language parameters must be passed explicitly for operations that take them.

Return type dict

Returns**Response Syntax**

```
{  
    'services': [  
        {  
            'code': 'string',  
            'name': 'string',  
            'categories': [  
                {  
                    'code': 'string',  
                    'name': 'string'  
                },  
            ],  
        },  
    ],  
}
```

Response Structure

- *(dict)* –

The list of AWS services returned by the DescribeServices operation.

- **services** (*list*) –

A JSON-formatted list of AWS services.

- * *(dict)* –

Information about an AWS service returned by the DescribeServices operation.

- **code** (*string*) –

The code for an AWS service returned by the DescribeServices response. The `Name` element contains the corresponding friendly name.

- **name** (*string*) –

The friendly name for an AWS service. The `Code` element contains the corresponding code.

- **categories** (*list*) –

A list of categories that describe the type of support issue a case describes. Categories consist of a category name and a category code. Category names and codes are passed to AWS Support when you call `CreateCase`.

- *(dict)* –

A JSON-formatted name/value pair that represents the category name and category code of the problem, selected from the DescribeServices response for each AWS service.

- **code** (*string*) –

The category code for the support case.

- **name** (*string*) –

The category name for the support case.

describe_severity_levels (***kwargs*)

Returns the list of severity levels that you can assign to an AWS Support case. The severity level for a case is also a field in the `CaseDetails` data type included in any `CreateCase` request.

Request Syntax

```
response = client.describe_severity_levels(
    language='string'
)
```

Parameters **language** (*string*) – The ISO 639-1 code for the language in which AWS provides support. AWS Support currently supports English (“en”) and Japanese (“ja”). Language parameters must be passed explicitly for operations that take them.

Return type dict

Returns

Response Syntax

```
{
    'severityLevels': [
        {
            'code': 'string',
            'name': 'string'
        }
    ]
}
```

```
    },  
    ]  
}
```

Response Structure

- (*dict*) –

The list of severity levels returned by the DescribeSeverityLevels operation.

- **severityLevels** (*list*) –

The available severity levels for the support case. Available severity levels are defined by your service level agreement with AWS.

- * (*dict*) –

A code and name pair that represent a severity level that can be applied to a support case.

- **code** (*string*) –

One of four values: “low,” “medium,” “high,” and “urgent”. These values correspond to response times returned to the caller in `SeverityLevel.name`.

- **name** (*string*) –

The name of the severity level that corresponds to the severity level code.

describe_trusted_advisor_check_refresh_statuses (**kwargs)

Returns the refresh status of the Trusted Advisor checks that have the specified check IDs. Check IDs can be obtained by calling `DescribeTrustedAdvisorChecks`.

Request Syntax

```
response = client.describe_trusted_advisor_check_refresh_statuses(  
    checkIds=[  
        'string',  
    ]  
)
```

Parameters **checkIds** (*list*) – [REQUIRED]

The IDs of the Trusted Advisor checks.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{  
    'statuses': [  
        {  
            'checkId': 'string',  
            'status': 'string',  
            'millisUntilNextRefreshable': 123  
        },  
    ]  
}
```

Response Structure

- (*dict*) –

The statuses of the Trusted Advisor checks returned by the DescribeTrustedAdvisorCheckRefreshStatuses operation.

- **statuses** (*list*) –

The refresh status of the specified Trusted Advisor checks.

- * (*dict*) –

The refresh status of a Trusted Advisor check.

- **checkId** (*string*) –

The unique identifier for the Trusted Advisor check.

- **status** (*string*) –

The status of the Trusted Advisor check for which a refresh has been requested: “none”, “enqueued”, “processing”, “success”, or “abandoned”.

- **millisUntilNextRefreshable** (*integer*) –

The amount of time, in milliseconds, until the Trusted Advisor check is eligible for refresh.

describe_trusted_advisor_check_result (***kwargs*)

Returns the results of the Trusted Advisor check that has the specified check ID. Check IDs can be obtained by calling DescribeTrustedAdvisorChecks .

The response contains a TrustedAdvisorCheckResult object, which contains these three objects:

- TrustedAdvisorCategorySpecificSummary
- TrustedAdvisorResourceDetail
- TrustedAdvisorResourcesSummary

In addition, the response contains these fields:

- Status.** The alert status of the check: “ok” (green), “warning” (yellow), “error” (red), or “not_available”.
- Timestamp.** The time of the last refresh of the check.
- CheckId.** The unique identifier for the check.

Request Syntax

```
response = client.describe_trusted_advisor_check_result(
    checkId='string',
    language='string'
)
```

Parameters

- **checkId** (*string*) – [REQUIRED]

The unique identifier for the Trusted Advisor check.

- **language** (*string*) – The ISO 639-1 code for the language in which AWS provides support. AWS Support currently supports English (“en”) and Japanese (“ja”). Language parameters must be passed explicitly for operations that take them.

Return type dict

Returns

Response Syntax

```
{
    'result': {
        'checkId': 'string',
        'timestamp': 'string',
```

```
{
  'status': 'string',
  'resourcesSummary': {
    'resourcesProcessed': 123,
    'resourcesFlagged': 123,
    'resourcesIgnored': 123,
    'resourcesSuppressed': 123
  },
  'categorySpecificSummary': {
    'costOptimizing': {
      'estimatedMonthlySavings': 123.0,
      'estimatedPercentMonthlySavings': 123.0
    }
  },
  'flaggedResources': [
    {
      'status': 'string',
      'region': 'string',
      'resourceId': 'string',
      'isSuppressed': True|False,
      'metadata': [
        'string',
      ]
    }
  ],
}
```

Response Structure

- (*dict*) –

The result of the Trusted Advisor check returned by the DescribeTrustedAdvisorCheckResult operation.

- **result** (*dict*) –

The detailed results of the Trusted Advisor check.

- * **checkId** (*string*) –

The unique identifier for the Trusted Advisor check.

- * **timestamp** (*string*) –

The time of the last refresh of the check.

- * **status** (*string*) –

The alert status of the check: “ok” (green), “warning” (yellow), “error” (red), or “not_available”.

- * **resourcesSummary** (*dict*) –

Details about AWS resources that were analyzed in a call to Trusted Advisor DescribeTrustedAdvisorCheckSummaries .

- **resourcesProcessed** (*integer*) –

The number of AWS resources that were analyzed by the Trusted Advisor check.

- **resourcesFlagged** (*integer*) –

The number of AWS resources that were flagged (listed) by the Trusted Advisor check.

- **resourcesIgnored** (*integer*) –

The number of AWS resources ignored by Trusted Advisor

because information was unavailable.

- **resourcesSuppressed** (*integer*) –

The number of AWS resources ignored by Trusted Advisor because they were marked as suppressed by the user.

- * **categorySpecificSummary** (*dict*) –

Summary information that relates to the category of the check. Cost Optimizing is the only category that is currently supported.

- **costOptimizing** (*dict*) –

The summary information about cost savings for a Trusted Advisor check that is in the Cost Optimizing category.

- **estimatedMonthlySavings** (*float*) –

The estimated monthly savings that might be realized if the recommended actions are taken.

- **estimatedPercentMonthlySavings** (*float*) –

The estimated percentage of savings that might be realized if the recommended actions are taken.

- * **flaggedResources** (*list*) –

The details about each resource listed in the check result.

- (*dict*) –

Contains information about a resource identified by a Trusted Advisor check.

- **status** (*string*) –

The status code for the resource identified in the Trusted Advisor check.

- **region** (*string*) –

The AWS region in which the identified resource is located.

- **resourceId** (*string*) –

The unique identifier for the identified resource.

- **isSuppressed** (*boolean*) –

Specifies whether the AWS resource was ignored by Trusted Advisor because it was marked as suppressed by the user.

- **metadata** (*list*) –

Additional information about the identified resource. The exact metadata and its order can be obtained by inspecting the `TrustedAdvisorCheckDescription` object returned by the call to `DescribeTrustedAdvisorChecks`. **Metadata** contains all the data that is shown in the Excel download, even in those cases where the UI shows just summary data.

- (*string*) –

describe_trusted_advisor_check_summaries (***kwargs*)

Returns the summaries of the results of the Trusted Advisor checks that have the specified check IDs. Check IDs can be obtained by calling `DescribeTrustedAdvisorChecks`.

The response contains an array of `TrustedAdvisorCheckSummary` objects.

Request Syntax

```
response = client.describe_trusted_advisor_check_summaries(
    checkIds=[
        'string',
    ]
)
```

Parameters `checkIds` (*list*) – **[REQUIRED]**

The IDs of the Trusted Advisor checks.

- (*string*) –

Return type dict

Returns

Response Syntax

```
{
    'summaries': [
        {
            'checkId': 'string',
            'timestamp': 'string',
            'status': 'string',
            'hasFlaggedResources': True|False,
            'resourcesSummary': {
                'resourcesProcessed': 123,
                'resourcesFlagged': 123,
                'resourcesIgnored': 123,
                'resourcesSuppressed': 123
            },
            'categorySpecificSummary': {
                'costOptimizing': {
                    'estimatedMonthlySavings': 123.0,
                    'estimatedPercentMonthlySavings': 123.0
                }
            }
        },
    ]
}
```

Response Structure

- (*dict*) –

The summaries of the Trusted Advisor checks returned by the DescribeTrustedAdvisorCheckSummaries operation.

- **summaries** (*list*) –

The summary information for the requested Trusted Advisor checks.

- * (*dict*) –

A summary of a Trusted Advisor check result, including the alert status, last refresh, and number of resources examined.

- **checkId** (*string*) –

The unique identifier for the Trusted Advisor check.

- **timestamp** (*string*) –

The time of the last refresh of the check.

- **status** (*string*) –

The alert status of the check: “ok” (green), “warning” (yellow), “error” (red), or “not_available”.

- **hasFlaggedResources** (*boolean*) –
Specifies whether the Trusted Advisor check has flagged resources.
- **resourcesSummary** (*dict*) –
Details about AWS resources that were analyzed in a call to `Trusted Advisor DescribeTrustedAdvisorCheckSummaries`.
- **resourcesProcessed** (*integer*) –
The number of AWS resources that were analyzed by the Trusted Advisor check.
- **resourcesFlagged** (*integer*) –
The number of AWS resources that were flagged (listed) by the Trusted Advisor check.
- **resourcesIgnored** (*integer*) –
The number of AWS resources ignored by Trusted Advisor because information was unavailable.
- **resourcesSuppressed** (*integer*) –
The number of AWS resources ignored by Trusted Advisor because they were marked as suppressed by the user.
- **categorySpecificSummary** (*dict*) –
Summary information that relates to the category of the check. Cost Optimizing is the only category that is currently supported.
- **costOptimizing** (*dict*) –
The summary information about cost savings for a Trusted Advisor check that is in the Cost Optimizing category.
- **estimatedMonthlySavings** (*float*) –
The estimated monthly savings that might be realized if the recommended actions are taken.
- **estimatedPercentMonthlySavings** (*float*) –
The estimated percentage of savings that might be realized if the recommended actions are taken.

describe_trusted_advisor_checks (***kwargs*)

Returns information about all available Trusted Advisor checks, including name, ID, category, description, and metadata. You must specify a language code; English (“en”) and Japanese (“ja”) are currently supported. The response contains a `TrustedAdvisorCheckDescription` for each check.

Request Syntax

```
response = client.describe_trusted_advisor_checks(
    language='string'
)
```

Parameters *language* (*string*) – [REQUIRED]

The ISO 639-1 code for the language in which AWS provides support. AWS Support currently supports English (“en”) and Japanese (“ja”). Language parameters must be passed explicitly for operations that take them.

Return type dict

Returns

Response Syntax

```
{
  'checks': [
    {
      'id': 'string',
      'name': 'string',
      'description': 'string',
      'category': 'string',
      'metadata': [
        'string',
      ]
    },
  ]
}
```

Response Structure

- (*dict*) –

Information about the Trusted Advisor checks returned by the DescribeTrustedAdvisorChecks operation.

- **checks** (*list*) –

Information about all available Trusted Advisor checks.

- * (*dict*) –

The description and metadata for a Trusted Advisor check.

- **id** (*string*) –

The unique identifier for the Trusted Advisor check.

- **name** (*string*) –

The display name for the Trusted Advisor check.

- **description** (*string*) –

The description of the Trusted Advisor check, which includes the alert criteria and recommended actions (contains HTML markup).

- **category** (*string*) –

The category of the Trusted Advisor check.

- **metadata** (*list*) –

The column headings for the data returned by the Trusted Advisor check. The order of the headings corresponds to the order of the data in the **Metadata** element of the TrustedAdvisorResourceDetail for the check. **Metadata** contains all the data that is shown in the Excel download, even in those cases where the UI shows just summary data.

- (*string*) –

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to *ClientMethod*.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)

- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method’s model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you’d normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_waiter (*waiter_name*)

refresh_trusted_advisor_check (***kwargs*)

Requests a refresh of the Trusted Advisor check that has the specified check ID. Check IDs can be obtained by calling `DescribeTrustedAdvisorChecks`.

The response contains a `TrustedAdvisorCheckRefreshStatus` object, which contains these fields:

- **Status**. The refresh status of the check: “none”, “enqueued”, “processing”, “success”, or “abandoned”.
- **MillisUntilNextRefreshable**. The amount of time, in milliseconds, until the check is eligible for refresh.
- **CheckId**. The unique identifier for the check.

Request Syntax

```
response = client.refresh_trusted_advisor_check(
    checkId='string'
)
```

Parameters **checkId** (*string*) – [REQUIRED]

The unique identifier for the Trusted Advisor check.

Return type `dict`

Returns

Response Syntax

```
{
    'status': {
        'checkId': 'string',
        'status': 'string',
        'millisUntilNextRefreshable': 123
    }
}
```

Response Structure

- (*dict*) –

The current refresh status of a Trusted Advisor check.

- **status** (*dict*) –

The current refresh status for a check, including the amount of time until the check is eligible for refresh.

- * **checkId** (*string*) –

The unique identifier for the Trusted Advisor check.

- * **status** (*string*) –

The status of the Trusted Advisor check for which a refresh has been requested: “none”, “enqueued”, “processing”, “success”, or “abandoned”.

- * **millisUntilNextRefreshable** (*integer*) –

The amount of time, in milliseconds, until the Trusted Advisor check is eligible for refresh.

resolve_case (***kwargs*)

Takes a `CaseId` and returns the initial state of the case along with the state of the case after the call to `ResolveCase` completed.

Request Syntax

```
response = client.resolve_case(
    caseId='string'
)
```

Parameters **caseId** (*string*) – The AWS Support case ID requested or returned in the call.

The case ID is an alphanumeric string formatted as shown in this example: `case-12345678910-2013-c4c1d2bf33c5cf47`

Return type dict

Returns

Response Syntax

```
{
    'initialCaseStatus': 'string',
    'finalCaseStatus': 'string'
}
```

Response Structure

- (*dict*) –

The status of the case returned by the `ResolveCase` operation.

- **initialCaseStatus** (*string*) –

The status of the case when the `ResolveCase` request was sent.

- **finalCaseStatus** (*string*) –

The status of the case after the `ResolveCase` request was processed.

Paginators

The available paginators are:

- `Support.Paginator.DescribeCases`
- `Support.Paginator.DescribeCommunications`

class `Support.Paginator.DescribeCases`

```
paginator = client.get_paginator('describe_cases')
```


paginate (**kwargs)

Creates an iterator that will paginate through responses from `Support.Client.describe_cases()`.

Request Syntax

```
response_iterator = paginator.paginate(
    caseIdList=[
        'string',
    ],
    displayId='string',
    afterTime='string',
    beforeTime='string',
    includeResolvedCases=True|False,
    language='string',
    includeCommunications=True|False,
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **caseIdList** (*list*) – A list of ID numbers of the support cases you want returned. The maximum number of cases is 100.
 - (*string*) –
- **displayId** (*string*) – The ID displayed for a case in the AWS Support Center user interface.
- **afterTime** (*string*) – The start date for a filtered date search on support case communications. Case communications are available for 12 months after creation.
- **beforeTime** (*string*) – The end date for a filtered date search on support case communications. Case communications are available for 12 months after creation.
- **includeResolvedCases** (*boolean*) – Specifies whether resolved support cases should be included in the DescribeCases results. The default is *false*.
- **language** (*string*) – The ISO 639-1 code for the language in which AWS provides support. AWS Support currently supports English (“en”) and Japanese (“ja”). Language parameters must be passed explicitly for operations that take them.
- **includeCommunications** (*boolean*) – Specifies whether communications should be included in the DescribeCases results. The default is *true*.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –

The size of each page.
 - **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'cases': [
    {
      'caseId': 'string',
      'displayId': 'string',
      'subject': 'string',
      'status': 'string',
      'serviceCode': 'string',
      'categoryCode': 'string',
      'severityCode': 'string',
      'submittedBy': 'string',
      'timeCreated': 'string',
      'recentCommunications': {
        'communications': [
          {
            'caseId': 'string',
            'body': 'string',
            'submittedBy': 'string',
            'timeCreated': 'string',
            'attachmentSet': [
              {
                'attachmentId': 'string',
                'fileName': 'string'
              }
            ]
          }
        ]
      },
      'nextToken': 'string'
    },
    {
      'ccEmailAddresses': [
        'string',
      ],
      'language': 'string'
    }
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Returns an array of `CaseDetails` objects and a `NextToken` that defines a point for pagination in the result set.

- **cases** (*list*) –

The details for the cases that match the request.

- * (*dict*) –

A JSON-formatted object that contains the metadata for a support case. It is contained the response from a `DescribeCases` request.

CaseDetails contains the following fields:

- **CaseID.** The AWS Support case ID requested or returned in the call. The case ID is an alphanumeric string formatted as shown in this example: `case-12345678910-2013-c4c1d2bf33c5cf47`.
- **CategoryCode.** The category of problem for the AWS Support case. Corresponds to the `CategoryCode` values returned

by a call to `DescribeServices` .

- **DisplayId.** The identifier for the case on pages in the AWS Support Center.
- **Language.** The ISO 639-1 code for the language in which AWS provides support. AWS Support currently supports English (“en”) and Japanese (“ja”). Language parameters must be passed explicitly for operations that take them.
- **RecentCommunications.** One or more Communication objects. Fields of these objects are `Attachments` , `Body` , `CaseId` , `SubmittedBy` , and `TimeCreated` .
- **NextToken.** A resumption point for pagination.
- **ServiceCode.** The identifier for the AWS service that corresponds to the service code defined in the call to `DescribeServices` .
- **SeverityCode.** The severity code assigned to the case. Contains one of the values returned by the call to `DescribeSeverityLevels` .
- **Status.** The status of the case in the AWS Support Center.
- **Subject.** The subject line of the case.
- **SubmittedBy.** The email address of the account that submitted the case.
- **TimeCreated.** The time the case was created, in ISO-8601 format.

- **caseId** (*string*) –

The AWS Support case ID requested or returned in the call. The case ID is an alphanumeric string formatted as shown in this example: `case-12345678910-2013-c4c1d2bf33c5cf47`

- **displayId** (*string*) –

The ID displayed for the case in the AWS Support Center. This is a numeric string.

- **subject** (*string*) –

The subject line for the case in the AWS Support Center.

- **status** (*string*) –

The status of the case.

- **serviceCode** (*string*) –

The code for the AWS service returned by the call to `DescribeServices` .

- **categoryCode** (*string*) –

The category of problem for the AWS Support case.

- **severityCode** (*string*) –

The code for the severity level returned by the call to `DescribeSeverityLevels` .

- **submittedBy** (*string*) –

The email address of the account that submitted the case.

- **timeCreated** (*string*) –

The time that the case was case created in the AWS Support Center.

- **recentCommunications** (*dict*) –

The five most recent communications between you and AWS Support Center, including the IDs of any attachments to the communications. Also includes a `nextToken` that you can use to retrieve earlier communications.

- **communications** (*list*) –

The five most recent communications associated with the case.

- (*dict*) –

A communication associated with an AWS Support case. The communication consists of the case ID, the message body, attachment information, the account email address, and the date and time of the communication.

- **caseId** (*string*) –

The AWS Support case ID requested or returned in the call. The case ID is an alphanumeric string formatted as shown in this example: `case-12345678910-2013-c4c1d2bf33c5cf47`

- **body** (*string*) –

The text of the communication between the customer and AWS Support.

- **submittedBy** (*string*) –

The email address of the account that submitted the AWS Support case.

- **timeCreated** (*string*) –

The time the communication was created.

- **attachmentSet** (*list*) –

Information about the attachments to the case communication.

- (*dict*) –

The file name and ID of an attachment to a case communication. You can use the ID to retrieve the attachment with the `DescribeAttachment` operation.

- **attachmentId** (*string*) –

The ID of the attachment.

- **fileName** (*string*) –

The file name of the attachment.

- **nextToken** (*string*) –

A resumption point for pagination.

- **ccEmailAddresses** (*list*) –

The email addresses that receive copies of communication about the case.

- (*string*) –

- **language** (*string*) –

The ISO 639-1 code for the language in which AWS provides support. AWS Support currently supports English (“en”) and Japanese (“ja”). Language parameters must be passed explicitly for operations that take them.

– **NextToken** (*string*) –

A token to resume pagination.

class `Support.Paginator.DescribeCommunications`

```
paginator = client.get_paginator('describe_communications')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `Support.Client.describe_communications()`.

Request Syntax

```
response_iterator = paginator.paginate(
    caseId='string',
    beforeTime='string',
    afterTime='string',
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **caseId** (*string*) – **[REQUIRED]**

The AWS Support case ID requested or returned in the call. The case ID is an alphanumeric string formatted as shown in this example: `case-12345678910-2013-c4c1d2bf33c5cf47`

- **beforeTime** (*string*) – The end date for a filtered date search on support case communications. Case communications are available for 12 months after creation.
- **afterTime** (*string*) – The start date for a filtered date search on support case communications. Case communications are available for 12 months after creation.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

– **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

– **PageSize** (*integer*) –

The size of each page.

– **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
    'communications': [
        {
```

```
        'caseId': 'string',
        'body': 'string',
        'submittedBy': 'string',
        'timeCreated': 'string',
        'attachmentSet': [
            {
                'attachmentId': 'string',
                'fileName': 'string'
            },
        ],
    },
],
'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

The communications returned by the DescribeCommunications operation.

- **communications** (*list*) –

The communications for the case.

- * (*dict*) –

A communication associated with an AWS Support case. The communication consists of the case ID, the message body, attachment information, the account email address, and the date and time of the communication.

- **caseId** (*string*) –

The AWS Support case ID requested or returned in the call. The case ID is an alphanumeric string formatted as shown in this example: case-12345678910-2013-c4c1d2bf33c5cf47

- **body** (*string*) –

The text of the communication between the customer and AWS Support.

- **submittedBy** (*string*) –

The email address of the account that submitted the AWS Support case.

- **timeCreated** (*string*) –

The time the communication was created.

- **attachmentSet** (*list*) –

Information about the attachments to the case communication.

- (*dict*) –

The file name and ID of an attachment to a case communication. You can use the ID to retrieve the attachment with the DescribeAttachment operation.

- **attachmentId** (*string*) –

The ID of the attachment.

- **fileName** (*string*) –

The file name of the attachment.

- **NextToken** (*string*) –

A token to resume pagination.

SWF

Table of Contents

- [SWF](#)
 - [Client](#)
 - [Paginators](#)

Client

class `SWF.Client`

A low-level client representing Amazon Simple Workflow Service (SWF):

```
import boto3
```

```
client = boto3.client('swf')
```

These are the available methods:

- `can_paginate()`
- `count_closed_workflow_executions()`
- `count_open_workflow_executions()`
- `count_pending_activity_tasks()`
- `count_pending_decision_tasks()`
- `deprecate_activity_type()`
- `deprecate_domain()`
- `deprecate_workflow_type()`
- `describe_activity_type()`
- `describe_domain()`
- `describe_workflow_execution()`
- `describe_workflow_type()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_waiter()`
- `get_workflow_execution_history()`
- `list_activity_types()`
- `list_closed_workflow_executions()`
- `list_domains()`
- `list_open_workflow_executions()`
- `list_workflow_types()`
- `poll_for_activity_task()`
- `poll_for_decision_task()`
- `record_activity_task_heartbeat()`
- `register_activity_type()`
- `register_domain()`
- `register_workflow_type()`
- `request_cancel_workflow_execution()`
- `respond_activity_task_canceled()`
- `respond_activity_task_completed()`
- `respond_activity_task_failed()`

- `respond_decision_task_completed()`
- `signal_workflow_execution()`
- `start_workflow_execution()`
- `terminate_workflow_execution()`

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

count_closed_workflow_executions (***kwargs*)

Returns the number of closed workflow executions within the given domain that meet the specified filtering criteria.

Note: This operation is eventually consistent. The results are best effort and may not exactly reflect recent updates and changes.

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- Constrain the following parameters by using a `Condition` element with the appropriate keys. *
 - `tagFilter.tag`: String constraint. The key is `swf:tagFilter.tag`.
 - `typeFilter.name`: String constraint. The key is `swf:typeFilter.name`.
 - `typeFilter.version`: String constraint. The key is `swf:typeFilter.version`.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.count_closed_workflow_executions(
    domain='string',
    startTimeFilter={
        'oldestDate': datetime(2015, 1, 1),
        'latestDate': datetime(2015, 1, 1)
    },
    closeTimeFilter={
        'oldestDate': datetime(2015, 1, 1),
        'latestDate': datetime(2015, 1, 1)
    },
    executionFilter={
        'workflowId': 'string'
    },
    typeFilter={
        'name': 'string',
        'version': 'string'
    },
    tagFilter={
        'tag': 'string'
    },
    closeStatusFilter={
```



```

    'status': 'COMPLETED' | 'FAILED' | 'CANCELED' | 'TERMINATED' | 'CONTINUED_AS_NEW' | 'TIMED_OUT'
}
)

```

Parameters

- **domain** (*string*) – [REQUIRED]

The name of the domain containing the workflow executions to count.

- **startTimeFilter** (*dict*) – If specified, only workflow executions that meet the start time criteria of the filter are counted.

Note: `startTimeFilter` and `closeTimeFilter` are mutually exclusive. You must specify one of these in a request but not both.

- **oldestDate** (*datetime*) – [REQUIRED]

Specifies the oldest start or close date and time to return.

- **latestDate** (*datetime*) –

Specifies the latest start or close date and time to return.

- **closeTimeFilter** (*dict*) – If specified, only workflow executions that meet the close time criteria of the filter are counted.

Note: `startTimeFilter` and `closeTimeFilter` are mutually exclusive. You must specify one of these in a request but not both.

- **oldestDate** (*datetime*) – [REQUIRED]

Specifies the oldest start or close date and time to return.

- **latestDate** (*datetime*) –

Specifies the latest start or close date and time to return.

- **executionFilter** (*dict*) – If specified, only workflow executions matching the `WorkflowId` in the filter are counted.

Note: `closeStatusFilter`, `executionFilter`, `typeFilter` and `tagFilter` are mutually exclusive. You can specify at most one of these in a request.

- **workflowId** (*string*) – [REQUIRED]

The `workflowId` to pass of match the criteria of this filter.

- **typeFilter** (*dict*) – If specified, indicates the type of the workflow executions to be counted.

Note: `closeStatusFilter`, `executionFilter`, `typeFilter` and `tagFilter` are mutually exclusive. You can specify at most one of these in a request.

- **name** (*string*) – [REQUIRED]

Required. Name of the workflow type.

- **version** (*string*) –

Version of the workflow type.

- **tagFilter** (*dict*) – If specified, only executions that have a tag that matches the filter are counted.

Note: `closeStatusFilter`, `executionFilter`, `typeFilter` and `tagFilter` are mutually exclusive. You can specify at most one of these in a request.

– **tag** (*string*) – [REQUIRED]

Required. Specifies the tag that must be associated with the execution for it to meet the filter criteria.

- **closeStatusFilter** (*dict*) – If specified, only workflow executions that match this close status are counted. This filter has an affect only if `executionStatus` is specified as `CLOSED`.

Note: `closeStatusFilter`, `executionFilter`, `typeFilter` and `tagFilter` are mutually exclusive. You can specify at most one of these in a request.

– **status** (*string*) – [REQUIRED]

Required. The close status that must match the close status of an execution for it to meet the criteria of this filter.

Return type `dict`

Returns

Response Syntax

```
{
    'count': 123,
    'truncated': True|False
}
```

Response Structure

- (*dict*) –

Contains the count of workflow executions returned from `CountOpenWorkflowExecutions` or `CountClosedWorkflowExecutions`

– **count** (*integer*) –

The number of workflow executions.

– **truncated** (*boolean*) –

If set to true, indicates that the actual count was more than the maximum supported by this API and the count returned is the truncated value.

count_open_workflow_executions (***kwargs*)

Returns the number of open workflow executions within the given domain that meet the specified filtering criteria.

Note: This operation is eventually consistent. The results are best effort and may not exactly reflect recent updates and changes.

Access Control

You can use IAM policies to control this action’s access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- Constrain the following parameters by using a `Condition` element with the appropriate keys. *
`tagFilter.tag`: String constraint. The key is `swf:tagFilter.tag`.

- `typeFilter.name`: String constraint. The key is `swf:typeFilter.name`.
- `typeFilter.version`: String constraint. The key is `swf:typeFilter.version`.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.count_open_workflow_executions(
    domain='string',
    startTimeFilter={
        'oldestDate': datetime(2015, 1, 1),
        'latestDate': datetime(2015, 1, 1)
    },
    typeFilter={
        'name': 'string',
        'version': 'string'
    },
    tagFilter={
        'tag': 'string'
    },
    executionFilter={
        'workflowId': 'string'
    }
)
```

Parameters

- **domain** (*string*) – [REQUIRED]
The name of the domain containing the workflow executions to count.
- **startTimeFilter** (*dict*) – [REQUIRED]
Specifies the start time criteria that workflow executions must meet in order to be counted.
 - **oldestDate** (*datetime*) – [REQUIRED]
Specifies the oldest start or close date and time to return.
 - **latestDate** (*datetime*) –
Specifies the latest start or close date and time to return.
- **typeFilter** (*dict*) – Specifies the type of the workflow executions to be counted.

Note: `executionFilter`, `typeFilter` and `tagFilter` are mutually exclusive. You can specify at most one of these in a request.

- **name** (*string*) – [REQUIRED]
Required. Name of the workflow type.
- **version** (*string*) –
Version of the workflow type.
- **tagFilter** (*dict*) – If specified, only executions that have a tag that matches the filter are counted.

Note: `executionFilter`, `typeFilter` and `tagFilter` are mutually exclusive. You can specify at most one of these in a request.

- **tag** (*string*) – [REQUIRED]

Required. Specifies the tag that must be associated with the execution for it to meet the filter criteria.

- **executionFilter** (*dict*) – If specified, only workflow executions matching the WorkflowId in the filter are counted.

Note: `executionFilter`, `typeFilter` and `tagFilter` are mutually exclusive. You can specify at most one of these in a request.

- **workflowId** (*string*) – [REQUIRED]

The workflowId to pass of match the criteria of this filter.

Return type dict

Returns

Response Syntax

```
{
    'count': 123,
    'truncated': True|False
}
```

Response Structure

- (*dict*) –

Contains the count of workflow executions returned from CountOpenWorkflowExecutions or CountClosedWorkflowExecutions

- **count** (*integer*) –

The number of workflow executions.

- **truncated** (*boolean*) –

If set to true, indicates that the actual count was more than the maximum supported by this API and the count returned is the truncated value.

count_pending_activity_tasks (***kwargs*)

Returns the estimated number of activity tasks in the specified task list. The count returned is an approximation and is not guaranteed to be exact. If you specify a task list that no activity task was ever scheduled in then 0 will be returned.

Access Control

You can use IAM policies to control this action’s access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- Constrain the `taskList.name` parameter by using a **Condition** element with the `swf:taskList.name` key to allow the action to access only certain task lists.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute’s **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.count_pending_activity_tasks(
    domain='string',
    taskList={
        'name': 'string'
```

```
}
)
```

Parameters

- **domain** (*string*) – [REQUIRED]
The name of the domain that contains the task list.
- **taskList** (*dict*) – [REQUIRED]
The name of the task list.
 - **name** (*string*) – [REQUIRED]
The name of the task list.

Return type dict

Returns

Response Syntax

```
{
    'count': 123,
    'truncated': True|False
}
```

Response Structure

- (*dict*) –
Contains the count of tasks in a task list.
 - **count** (*integer*) –
The number of tasks in the task list.
 - **truncated** (*boolean*) –
If set to true, indicates that the actual count was more than the maximum supported by this API and the count returned is the truncated value.

count_pending_decision_tasks (***kwargs*)

Returns the estimated number of decision tasks in the specified task list. The count returned is an approximation and is not guaranteed to be exact. If you specify a task list that no decision task was ever scheduled in then 0 will be returned.

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a **Resource** element with the domain name to limit the action to only specified domains.
- Use an **Action** element to allow or deny permission to call this action.
- Constrain the `taskList.name` parameter by using a **Condition** element with the `swf:taskList.name` key to allow the action to access only certain task lists.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.count_pending_decision_tasks(
    domain='string',
    taskList={
        'name': 'string'
    }
)
```

Parameters

- **domain** (*string*) – [REQUIRED]
The name of the domain that contains the task list.
- **taskList** (*dict*) – [REQUIRED]
The name of the task list.
 - **name** (*string*) – [REQUIRED]
The name of the task list.

Return type dict**Returns****Response Syntax**

```
{
    'count': 123,
    'truncated': True|False
}
```

Response Structure

- (*dict*) –
Contains the count of tasks in a task list.
 - **count** (*integer*) –
The number of tasks in the task list.
 - **truncated** (*boolean*) –
If set to true, indicates that the actual count was more than the maximum supported by this API and the count returned is the truncated value.

deprecate_activity_type (***kwargs*)

Deprecates the specified *activity type* . After an activity type has been deprecated, you cannot create new tasks of that activity type. Tasks of this type that were scheduled before the type was deprecated will continue to run.

Note: This operation is eventually consistent. The results are best effort and may not exactly reflect recent updates and changes.

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- Constrain the following parameters by using a `Condition` element with the appropriate keys. *
 - `activityType.name` : String constraint. The key is `swf:activityType.name` .
 - `activityType.version` : String constraint. The key is `swf:activityType.version` .

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#) .

Request Syntax

```
response = client.deprecate_activity_type(
    domain='string',
    activityType={
        'name': 'string',
```

```

        'version': 'string'
    }
)

```

Parameters

- **domain** (*string*) – [REQUIRED]

The name of the domain in which the activity type is registered.

- **activityType** (*dict*) – [REQUIRED]

The activity type to deprecate.

- **name** (*string*) – [REQUIRED]

The name of this activity.

Note: The combination of activity type name and version must be unique within a domain.

- **version** (*string*) – [REQUIRED]

The version of this activity.

Note: The combination of activity type name and version must be unique with in a domain.

Returns None

deprecate_domain (***kwargs*)

Deprecates the specified domain. After a domain has been deprecated it cannot be used to create new workflow executions or register new types. However, you can still use visibility actions on this domain. Deprecating a domain also deprecates all activity and workflow types registered in the domain. Executions that were started before the domain was deprecated will continue to run.

Note: This operation is eventually consistent. The results are best effort and may not exactly reflect recent updates and changes.

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- You cannot use an IAM policy to constrain this action's parameters.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```

response = client.deprecate_domain(
    name='string'
)

```

Parameters **name** (*string*) – [REQUIRED]

The name of the domain to deprecate.

Returns None

deprecate_workflow_type (**kwargs)

Deprecates the specified *workflow type* . After a workflow type has been deprecated, you cannot create new executions of that type. Executions that were started before the type was deprecated will continue to run. A deprecated workflow type may still be used when calling visibility actions.

Note: This operation is eventually consistent. The results are best effort and may not exactly reflect recent updates and changes.

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- Constrain the following parameters by using a `Condition` element with the appropriate keys. *
 - `workflowType.name` : String constraint. The key is `swf:workflowType.name` .
 - `workflowType.version` : String constraint. The key is `swf:workflowType.version` .

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#) .

Request Syntax

```
response = client.deprecate_workflow_type(
    domain='string',
    workflowType={
        'name': 'string',
        'version': 'string'
    }
)
```

Parameters

- **domain** (*string*) – [REQUIRED]

The name of the domain in which the workflow type is registered.

- **workflowType** (*dict*) – [REQUIRED]

The workflow type to deprecate.

- **name** (*string*) – [REQUIRED]

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique within a domain.

- **version** (*string*) – [REQUIRED]

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique within a domain.

Returns None**describe_activity_type** (**kwargs)

Returns information about the specified activity type. This includes configuration settings provided when the type was registered and other general information about the type.

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- Constrain the following parameters by using a `Condition` element with the appropriate keys. *
 - `activityType.name`: String constraint. The key is `swf:activityType.name`.
 - `activityType.version`: String constraint. The key is `swf:activityType.version`.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.describe_activity_type(
    domain='string',
    activityType={
        'name': 'string',
        'version': 'string'
    }
)
```

Parameters

- **domain** (*string*) – [REQUIRED]

The name of the domain in which the activity type is registered.

- **activityType** (*dict*) – [REQUIRED]

The activity type to get information about. Activity types are identified by the name and version that were supplied when the activity was registered.

- **name** (*string*) – [REQUIRED]

The name of this activity.

Note: The combination of activity type name and version must be unique within a domain.

- **version** (*string*) – [REQUIRED]

The version of this activity.

Note: The combination of activity type name and version must be unique with in a domain.

Return type dict

Returns

Response Syntax

```
{
    'typeInfo': {
        'activityType': {
            'name': 'string',
            'version': 'string'
        },
        'status': 'REGISTERED'|'DEPRECATED',
        'description': 'string',
        'creationDate': datetime(2015, 1, 1),
        'deprecationDate': datetime(2015, 1, 1)
    },
}
```

```
'configuration': {
    'defaultTaskStartToCloseTimeout': 'string',
    'defaultTaskHeartbeatTimeout': 'string',
    'defaultTaskList': {
        'name': 'string'
    },
    'defaultTaskPriority': 'string',
    'defaultTaskScheduleToStartTimeout': 'string',
    'defaultTaskScheduleToCloseTimeout': 'string'
}
```

Response Structure

- *(dict)* –

Detailed information about an activity type.

- **typeInfo** (*dict*) –

General information about the activity type.

The status of activity type (returned in the ActivityTypeInfo structure) can be one of the following.

- * **REGISTERED** : The type is registered and available. Workers supporting this type should be running.
- * **DEPRECATED** : The type was deprecated using `DeprecateActivityType` , but is still in use. You should keep workers supporting this type running. You cannot create new tasks of this type.

- * **activityType** (*dict*) –

The ActivityType type structure representing the activity type.

- **name** (*string*) –

The name of this activity.

Note: The combination of activity type name and version must be unique within a domain.

- **version** (*string*) –

The version of this activity.

Note: The combination of activity type name and version must be unique with in a domain.

- * **status** (*string*) –

The current status of the activity type.

- * **description** (*string*) –

The description of the activity type provided in `RegisterActivityType` .

- * **creationDate** (*datetime*) –

The date and time this activity type was created through `RegisterActivityType` .

- * **deprecationDate** (*datetime*) –

If **DEPRECATED**, the date and time `DeprecateActivityType` was called.

– **configuration** (*dict*) –

The configuration settings registered with the activity type.

* **defaultTaskStartToCloseTimeout** (*string*) –

Optional. The default maximum duration for tasks of an activity type specified when registering the activity type. You can override this default when scheduling a task through the `ScheduleActivityTask` decision.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

* **defaultTaskHeartbeatTimeout** (*string*) –

Optional. The default maximum time, in seconds, before which a worker processing a task must report progress by calling `RecordActivityTaskHeartbeat`.

You can specify this value only when *registering* an activity type. The registered default value can be overridden when you schedule a task through the `ScheduleActivityTask` decision. If the activity worker subsequently attempts to record a heartbeat or returns a result, the activity worker receives an `UnknownResource` fault. In this case, Amazon SWF no longer considers the activity task to be valid; the activity worker should clean up the activity task.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

* **defaultTaskList** (*dict*) –

Optional. The default task list specified for this activity type at registration. This default is used if a task list is not provided when a task is scheduled through the `ScheduleActivityTask` decision. You can override the default registered task list when scheduling a task through the `ScheduleActivityTask` decision.

· **name** (*string*) –

The name of the task list.

* **defaultTaskPriority** (*string*) –

Optional. The default task priority for tasks of this activity type, specified at registration. If not set, then “0” will be used as the default priority. This default can be overridden when scheduling an activity task.

Valid values are integers that range from Java’s `Integer.MIN_VALUE` (-2147483648) to `Integer.MAX_VALUE` (2147483647). Higher numbers indicate higher priority.

For more information about setting task priority, see [Setting Task Priority](#) in the *Amazon Simple Workflow Developer Guide*.

* **defaultTaskScheduleToStartTimeout** (*string*) –

Optional. The default maximum duration, specified when registering the activity type, that a task of an activity type can wait before being assigned to a worker. You can override this default when scheduling a task through the `ScheduleActivityTask` decision.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

* **defaultTaskScheduleToCloseTimeout** (*string*) –

Optional. The default maximum duration, specified when registering the activity type, for tasks of this activity type. You can override this default when scheduling a task through the `ScheduleActivityTask` decision.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

describe_domain (***kwargs*)

Returns information about the specified domain, including description and status.

Access Control

You can use IAM policies to control this action’s access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- You cannot use an IAM policy to constrain this action’s parameters.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute’s **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.describe_domain(
    name='string'
)
```

Parameters `name` (*string*) – [REQUIRED]

The name of the domain to describe.

Return type dict

Returns

Response Syntax

```
{
  'domainInfo': {
    'name': 'string',
    'status': 'REGISTERED'|'DEPRECATED',
    'description': 'string'
  },
  'configuration': {
    'workflowExecutionRetentionPeriodInDays': 'string'
  }
}
```

Response Structure

- (*dict*) –

Contains details of a domain.

– **domainInfo** (*dict*) –

Contains general information about a domain.

* **name** (*string*) –

The name of the domain. This name is unique within the account.

* **status** (*string*) –

The status of the domain:

- **REGISTERED** : The domain is properly registered and available. You can use this domain for registering types and creating new workflow executions.
- **DEPRECATED** : The domain was deprecated using `DeprecateDomain` , but is still in use. You should not create new workflow executions in this domain.

* **description** (*string*) –

The description of the domain provided through `RegisterDomain` .

– **configuration** (*dict*) –

Contains the configuration settings of a domain.

* **workflowExecutionRetentionPeriodInDays** (*string*) –

The retention period for workflow executions in this domain.

describe_workflow_execution (***kwargs*)

Returns information about the specified workflow execution including its type and some statistics.

Note: This operation is eventually consistent. The results are best effort and may not exactly reflect recent updates and changes.

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- You cannot use an IAM policy to constrain this action's parameters.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#) .

Request Syntax

```
response = client.describe_workflow_execution(
    domain='string',
    execution={
        'workflowId': 'string',
        'runId': 'string'
    }
)
```

Parameters

- **domain** (*string*) – [REQUIRED]

The name of the domain containing the workflow execution.

- **execution** (*dict*) – [REQUIRED]

The workflow execution to describe.

- **workflowId** (*string*) – [REQUIRED]

The user defined identifier associated with the workflow execution.

- **runId** (*string*) – [REQUIRED]

A system-generated unique identifier for the workflow execution.

Return type dict

Returns

Response Syntax

```
{
  'executionInfo': {
    'execution': {
      'workflowId': 'string',
      'runId': 'string'
    },
    'workflowType': {
      'name': 'string',
      'version': 'string'
    },
    'startTimestamp': datetime(2015, 1, 1),
    'closeTimestamp': datetime(2015, 1, 1),
    'executionStatus': 'OPEN'|'CLOSED',
    'closeStatus': 'COMPLETED'|'FAILED'|'CANCELED'|'TERMINATED'|'CONTINUED_AS_
    'parent': {
      'workflowId': 'string',
      'runId': 'string'
    },
    'tagList': [
      'string',
    ],
    'cancelRequested': True|False
  },
  'executionConfiguration': {
    'taskStartToCloseTimeout': 'string',
    'executionStartToCloseTimeout': 'string',
    'taskList': {
      'name': 'string'
    },
    'taskPriority': 'string',
    'childPolicy': 'TERMINATE'|'REQUEST_CANCEL'|'ABANDON',
    'lambdaRole': 'string'
  },
  'openCounts': {
    'openActivityTasks': 123,
    'openDecisionTasks': 123,
    'openTimers': 123,
    'openChildWorkflowExecutions': 123,
    'openLambdaFunctions': 123
  },
  'latestActivityTaskTimestamp': datetime(2015, 1, 1),
  'latestExecutionContext': 'string'
}
```

Response Structure

- (*dict*) –

Contains details about a workflow execution.

- **executionInfo** (*dict*) –

Information about the workflow execution.

- * **execution** (*dict*) –

The workflow execution this information is about.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- * **workflowType** (*dict*) –

The type of the workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- * **startTimestamp** (*datetime*) –

The time when the execution was started.

- * **closeTimestamp** (*datetime*) –

The time when the workflow execution was closed. Set only if the execution status is CLOSED.

- * **executionStatus** (*string*) –

The current status of the execution.

- * **closeStatus** (*string*) –

If the execution status is closed then this specifies how the execution was closed:

- **COMPLETED** : the execution was successfully completed.
- **CANCELED** : the execution was canceled. Cancellation allows the implementation to gracefully clean up before the execution is closed.
- **TERMINATED** : the execution was force terminated.
- **FAILED** : the execution failed to complete.
- **TIMED_OUT** : the execution did not complete in the allotted time and was automatically timed out.
- **CONTINUED_AS_NEW** : the execution is logically continued. This means the current execution was completed and a new execution was started to carry on the workflow.

- * **parent** (*dict*) –

If this workflow execution is a child of another execution then contains the workflow execution that started this execution.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- * **tagList** (*list*) –

The list of tags associated with the workflow execution. Tags can be used to identify and list workflow executions of interest through the visibility APIs. A workflow execution can have a maximum of 5 tags.

- (*string*) –

- * **cancelRequested** (*boolean*) –

Set to true if a cancellation is requested for this workflow execution.

- **executionConfiguration** (*dict*) –

The configuration settings for this workflow execution including timeout values, tasklist etc.

- * **taskStartToCloseTimeout** (*string*) –

The maximum duration allowed for decision tasks for this workflow execution.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- * **executionStartToCloseTimeout** (*string*) –

The total duration for this workflow execution.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- * **taskList** (*dict*) –

The task list used for the decision tasks generated for this workflow execution.

- **name** (*string*) –

The name of the task list.

- * **taskPriority** (*string*) –

The priority assigned to decision tasks for this workflow execution. Valid values are integers that range from Java’s `Integer.MIN_VALUE` (-2147483648) to `Integer.MAX_VALUE` (2147483647). Higher numbers indicate higher priority.

For more information about setting task priority, see [Setting Task Priority](#) in the *Amazon Simple Workflow Developer Guide*.

- * **childPolicy** (*string*) –

The policy to use for the child workflow executions if this workflow execution is terminated, by calling the `TerminateWorkflowExecution` action explicitly or due to an expired timeout.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.
- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON:** no action will be taken. The child executions will continue to run.

- * **lambdaRole** (*string*) –

The IAM role used by this workflow execution when invoking AWS Lambda functions.

– **openCounts** (*dict*) –

The number of tasks for this workflow execution. This includes open and closed tasks of all types.

* **openActivityTasks** (*integer*) –

The count of activity tasks whose status is OPEN.

* **openDecisionTasks** (*integer*) –

The count of decision tasks whose status is OPEN. A workflow execution can have at most one open decision task.

* **openTimers** (*integer*) –

The count of timers started by this workflow execution that have not fired yet.

* **openChildWorkflowExecutions** (*integer*) –

The count of child workflow executions whose status is OPEN.

* **openLambdaFunctions** (*integer*) –

The count of AWS Lambda functions that are currently executing.

– **latestActivityTaskTimestamp** (*datetime*) –

The time when the last activity task was scheduled for this workflow execution. You can use this information to determine if the workflow has not made progress for an unusually long period of time and might require a corrective action.

– **latestExecutionContext** (*string*) –

The latest executionContext provided by the decider for this workflow execution. A decider can provide an executionContext (a free-form string) when closing a decision task using RespondDecisionTaskCompleted .

describe_workflow_type (***kwargs*)

Returns information about the specified *workflow type* . This includes configuration settings specified when the type was registered and other information such as creation date, current status, and so on.

Access Control

You can use IAM policies to control this action’s access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- Constrain the following parameters by using a `Condition` element with the appropriate keys. *
`workflowType.name` : String constraint. The key is `swf:workflowType.name` .
- `workflowType.version` : String constraint. The key is `swf:workflowType.version` .

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute’s **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#) .

Request Syntax

```
response = client.describe_workflow_type(
    domain='string',
    workflowType={
        'name': 'string',
        'version': 'string'
```

```
}  
)
```

Parameters

- **domain** (*string*) – [REQUIRED]

The name of the domain in which this workflow type is registered.

- **workflowType** (*dict*) – [REQUIRED]

The workflow type to describe.

- **name** (*string*) – [REQUIRED]

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique within a domain.

- **version** (*string*) – [REQUIRED]

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique within a domain.

Return type dict

Returns**Response Syntax**

```
{  
  'typeInfo': {  
    'workflowType': {  
      'name': 'string',  
      'version': 'string'  
    },  
    'status': 'REGISTERED'|'DEPRECATED',  
    'description': 'string',  
    'creationDate': datetime(2015, 1, 1),  
    'deprecationDate': datetime(2015, 1, 1)  
  },  
  'configuration': {  
    'defaultTaskStartToCloseTimeout': 'string',  
    'defaultExecutionStartToCloseTimeout': 'string',  
    'defaultTaskList': {  
      'name': 'string'  
    },  
    'defaultTaskPriority': 'string',  
    'defaultChildPolicy': 'TERMINATE'|'REQUEST_CANCEL'|'ABANDON',  
    'defaultLambdaRole': 'string'  
  }  
}
```

Response Structure

- (*dict*) –

Contains details about a workflow type.

- **typeInfo** (*dict*) –

General information about the workflow type.

The status of the workflow type (returned in the `WorkflowTypeInfo` structure) can be one of the following.

- * **REGISTERED** : The type is registered and available. Workers supporting this type should be running.
- * **DEPRECATED** : The type was deprecated using `DeprecateWorkflowType` , but is still in use. You should keep workers supporting this type running. You cannot create new workflow executions of this type.

* **workflowType** (*dict*) –

The workflow type this information is about.

· **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

· **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

* **status** (*string*) –

The current status of the workflow type.

* **description** (*string*) –

The description of the type registered through `RegisterWorkflowType` .

* **creationDate** (*datetime*) –

The date when this type was registered.

* **deprecationDate** (*datetime*) –

If the type is in deprecated state, then it is set to the date when the type was deprecated.

– **configuration** (*dict*) –

Configuration settings of the workflow type registered through `RegisterWorkflowType`

* **defaultTaskStartToCloseTimeout** (*string*) –

Optional. The default maximum duration, specified when registering the workflow type, that a decision task for executions of this workflow type might take before returning completion or failure. If the task does not close in the specified time then the task is automatically timed out and rescheduled. If the decider eventually reports a completion or failure, it is ignored. This default can be overridden when starting a workflow execution using the `StartWorkflowExecution` action or the `StartChildWorkflowExecution` decision.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

* **defaultExecutionStartToCloseTimeout** (*string*) –

Optional. The default maximum duration, specified when registering the workflow type, for executions of this workflow

type. This default can be overridden when starting a workflow execution using the `StartWorkflowExecution` action or the `StartChildWorkflowExecution` decision.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

* **defaultTaskList** (*dict*) –

Optional. The default task list, specified when registering the workflow type, for decisions tasks scheduled for workflow executions of this type. This default can be overridden when starting a workflow execution using the `StartWorkflowExecution` action or the `StartChildWorkflowExecution` decision.

· **name** (*string*) –

The name of the task list.

* **defaultTaskPriority** (*string*) –

Optional. The default task priority, specified when registering the workflow type, for all decision tasks of this workflow type. This default can be overridden when starting a workflow execution using the `StartWorkflowExecution` action or the `StartChildWorkflowExecution` decision.

Valid values are integers that range from Java’s `Integer.MIN_VALUE` (-2147483648) to `Integer.MAX_VALUE` (2147483647). Higher numbers indicate higher priority.

For more information about setting task priority, see [Setting Task Priority](#) in the *Amazon Simple Workflow Developer Guide*.

* **defaultChildPolicy** (*string*) –

Optional. The default policy to use for the child workflow executions when a workflow execution of this type is terminated, by calling the `TerminateWorkflowExecution` action explicitly or due to an expired timeout. This default can be overridden when starting a workflow execution using the `StartWorkflowExecution` action or the `StartChildWorkflowExecution` decision.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.
- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON:** no action will be taken. The child executions will continue to run.

* **defaultLambdaRole** (*string*) –

The default IAM role to use when a workflow execution invokes a AWS Lambda function.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.

- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters **operation_name** (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_waiter (*waiter_name*)

get_workflow_execution_history (***kwargs*)

Returns the history of the specified workflow execution. The results may be split into multiple pages. To retrieve subsequent pages, make the call again using the `nextPageToken` returned by the initial call.

Note: This operation is eventually consistent. The results are best effort and may not exactly reflect recent updates and changes.

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- You cannot use an IAM policy to constrain this action's parameters.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.get_workflow_execution_history(
    domain='string',
    execution={
        'workflowId': 'string',
        'runId': 'string'
    },
    nextPageToken='string',
    maximumPageSize=123,
    reverseOrder=True|False
)
```

Parameters

- **domain** (*string*) – [REQUIRED]
The name of the domain containing the workflow execution.
- **execution** (*dict*) – [REQUIRED]
Specifies the workflow execution for which to return the history.

- **workflowId** (*string*) – [REQUIRED]

The user defined identifier associated with the workflow execution.

- **runId** (*string*) – [REQUIRED]

A system-generated unique identifier for the workflow execution.

- **nextPageToken** (*string*) – If a `NextPageToken` was returned by a previous call, there are more results available. To retrieve the next page of results, make the call again using the returned token in `nextPageToken`. Keep all other arguments unchanged.

The configured `maximumPageSize` determines how many results can be returned in a single call.

- **maximumPageSize** (*integer*) – The maximum number of results that will be returned per call. `nextPageToken` can be used to obtain further pages of results. The default is 1000, which is the maximum allowed page size. You can, however, specify a page size *smaller* than the maximum.

This is an upper limit only; the actual number of results returned per call may be fewer than the specified maximum.

- **reverseOrder** (*boolean*) – When set to `true`, returns the events in reverse order. By default the results are returned in ascending order of the `eventTimeStamp` of the events.

Return type dict

Returns

Response Syntax

```
{
  'events': [
    {
      'eventTimestamp': datetime(2015, 1, 1),
      'eventType': 'WorkflowExecutionStarted'|'WorkflowExecutionCancelRequest',
      'eventId': 123,
      'workflowExecutionStartedEventAttributes': {
        'input': 'string',
        'executionStartToCloseTimeout': 'string',
        'taskStartToCloseTimeout': 'string',
        'childPolicy': 'TERMINATE'|'REQUEST_CANCEL'|'ABANDON',
        'taskList': {
          'name': 'string'
        },
        'workflowType': {
          'name': 'string',
          'version': 'string'
        },
        'tagList': [
          'string',
        ],
        'taskPriority': 'string',
        'continuedExecutionRunId': 'string',
        'parentWorkflowExecution': {
          'workflowId': 'string',
          'runId': 'string'
        },
        'parentInitiatedEventId': 123,
        'lambdaRole': 'string'
      },
      'workflowExecutionCompletedEventAttributes': {
```

```

        'result': 'string',
        'decisionTaskCompletedEventId': 123
    },
    'completeWorkflowExecutionFailedEventAttributes': {
        'cause': 'UNHANDLED_DECISION'|'OPERATION_NOT_PERMITTED',
        'decisionTaskCompletedEventId': 123
    },
    'workflowExecutionFailedEventAttributes': {
        'reason': 'string',
        'details': 'string',
        'decisionTaskCompletedEventId': 123
    },
    'failWorkflowExecutionFailedEventAttributes': {
        'cause': 'UNHANDLED_DECISION'|'OPERATION_NOT_PERMITTED',
        'decisionTaskCompletedEventId': 123
    },
    'workflowExecutionTimedOutEventAttributes': {
        'timeoutType': 'START_TO_CLOSE',
        'childPolicy': 'TERMINATE'|'REQUEST_CANCEL'|'ABANDON'
    },
    'workflowExecutionCanceledEventAttributes': {
        'details': 'string',
        'decisionTaskCompletedEventId': 123
    },
    'cancelWorkflowExecutionFailedEventAttributes': {
        'cause': 'UNHANDLED_DECISION'|'OPERATION_NOT_PERMITTED',
        'decisionTaskCompletedEventId': 123
    },
    'workflowExecutionContinuedAsNewEventAttributes': {
        'input': 'string',
        'decisionTaskCompletedEventId': 123,
        'newExecutionRunId': 'string',
        'executionStartToCloseTimeout': 'string',
        'taskList': {
            'name': 'string'
        },
        'taskPriority': 'string',
        'taskStartToCloseTimeout': 'string',
        'childPolicy': 'TERMINATE'|'REQUEST_CANCEL'|'ABANDON',
        'tagList': [
            'string',
        ],
        'workflowType': {
            'name': 'string',
            'version': 'string'
        },
        'lambdaRole': 'string'
    },
    'continueAsNewWorkflowExecutionFailedEventAttributes': {
        'cause': 'UNHANDLED_DECISION'|'WORKFLOW_TYPE_DEPRECATED'|'WORKFLOW_
        'decisionTaskCompletedEventId': 123
    },
    'workflowExecutionTerminatedEventAttributes': {
        'reason': 'string',
        'details': 'string',
        'childPolicy': 'TERMINATE'|'REQUEST_CANCEL'|'ABANDON',
        'cause': 'CHILD_POLICY_APPLIED'|'EVENT_LIMIT_EXCEEDED'|'OPERATOR_I
    },

```

```
'workflowExecutionCancelRequestedEventAttributes': {
    'externalWorkflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'externalInitiatedEventId': 123,
    'cause': 'CHILD_POLICY_APPLIED'
},
'decisionTaskScheduledEventAttributes': {
    'taskList': {
        'name': 'string'
    },
    'taskPriority': 'string',
    'startToCloseTimeout': 'string'
},
'decisionTaskStartedEventAttributes': {
    'identity': 'string',
    'scheduledEventId': 123
},
'decisionTaskCompletedEventAttributes': {
    'executionContext': 'string',
    'scheduledEventId': 123,
    'startedEventId': 123
},
'decisionTaskTimedOutEventAttributes': {
    'timeoutType': 'START_TO_CLOSE',
    'scheduledEventId': 123,
    'startedEventId': 123
},
'activityTaskScheduledEventAttributes': {
    'activityType': {
        'name': 'string',
        'version': 'string'
    },
    'activityId': 'string',
    'input': 'string',
    'control': 'string',
    'scheduleToStartTimeout': 'string',
    'scheduleToCloseTimeout': 'string',
    'startToCloseTimeout': 'string',
    'taskList': {
        'name': 'string'
    },
    'taskPriority': 'string',
    'decisionTaskCompletedEventId': 123,
    'heartbeatTimeout': 'string'
},
'activityTaskStartedEventAttributes': {
    'identity': 'string',
    'scheduledEventId': 123
},
'activityTaskCompletedEventAttributes': {
    'result': 'string',
    'scheduledEventId': 123,
    'startedEventId': 123
},
'activityTaskFailedEventAttributes': {
    'reason': 'string',
```



```

        'details': 'string',
        'scheduledEventId': 123,
        'startedEventId': 123
    },
    'activityTaskTimedOutEventAttributes': {
        'timeoutType': 'START_TO_CLOSE'|'SCHEDULE_TO_START'|'SCHEDULE_TO_C
        'scheduledEventId': 123,
        'startedEventId': 123,
        'details': 'string'
    },
    'activityTaskCanceledEventAttributes': {
        'details': 'string',
        'scheduledEventId': 123,
        'startedEventId': 123,
        'latestCancelRequestedEventId': 123
    },
    'activityTaskCancelRequestedEventAttributes': {
        'decisionTaskCompletedEventId': 123,
        'activityId': 'string'
    },
    'workflowExecutionSignaledEventAttributes': {
        'signalName': 'string',
        'input': 'string',
        'externalWorkflowExecution': {
            'workflowId': 'string',
            'runId': 'string'
        },
        'externalInitiatedEventId': 123
    },
    'markerRecordedEventAttributes': {
        'markerName': 'string',
        'details': 'string',
        'decisionTaskCompletedEventId': 123
    },
    'recordMarkerFailedEventAttributes': {
        'markerName': 'string',
        'cause': 'OPERATION_NOT_PERMITTED',
        'decisionTaskCompletedEventId': 123
    },
    'timerStartedEventAttributes': {
        'timerId': 'string',
        'control': 'string',
        'startToFireTimeout': 'string',
        'decisionTaskCompletedEventId': 123
    },
    'timerFiredEventAttributes': {
        'timerId': 'string',
        'startedEventId': 123
    },
    'timerCanceledEventAttributes': {
        'timerId': 'string',
        'startedEventId': 123,
        'decisionTaskCompletedEventId': 123
    },
    'startChildWorkflowExecutionInitiatedEventAttributes': {
        'workflowId': 'string',
        'workflowType': {
            'name': 'string',

```

```
        'version': 'string'
    },
    'control': 'string',
    'input': 'string',
    'executionStartToCloseTimeout': 'string',
    'taskList': {
        'name': 'string'
    },
    'taskPriority': 'string',
    'decisionTaskCompletedEventId': 123,
    'childPolicy': 'TERMINATE'|'REQUEST_CANCEL'|'ABANDON',
    'taskStartToCloseTimeout': 'string',
    'tagList': [
        'string',
    ],
    'lambdaRole': 'string'
},
'childWorkflowExecutionStartedEventAttributes': {
    'workflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'initiatedEventId': 123
},
'childWorkflowExecutionCompletedEventAttributes': {
    'workflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'result': 'string',
    'initiatedEventId': 123,
    'startedEventId': 123
},
'childWorkflowExecutionFailedEventAttributes': {
    'workflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'reason': 'string',
    'details': 'string',
    'initiatedEventId': 123,
    'startedEventId': 123
},
'childWorkflowExecutionTimedOutEventAttributes': {
    'workflowExecution': {
        'workflowId': 'string',
```

```

        'runId': 'string'
    },
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'timeoutType': 'START_TO_CLOSE',
    'initiatedEventId': 123,
    'startedEventId': 123
},
'childWorkflowExecutionCanceledEventAttributes': {
    'workflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'details': 'string',
    'initiatedEventId': 123,
    'startedEventId': 123
},
'childWorkflowExecutionTerminatedEventAttributes': {
    'workflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'initiatedEventId': 123,
    'startedEventId': 123
},
'signalExternalWorkflowExecutionInitiatedEventAttributes': {
    'workflowId': 'string',
    'runId': 'string',
    'signalName': 'string',
    'input': 'string',
    'decisionTaskCompletedEventId': 123,
    'control': 'string'
},
'externalWorkflowExecutionSignaledEventAttributes': {
    'workflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'initiatedEventId': 123
},
'signalExternalWorkflowExecutionFailedEventAttributes': {
    'workflowId': 'string',
    'runId': 'string',
    'cause': 'UNKNOWN_EXTERNAL_WORKFLOW_EXECUTION'|'SIGNAL_EXTERNAL_WO
    'initiatedEventId': 123,
    'decisionTaskCompletedEventId': 123,
    'control': 'string'
},

```

```
'externalWorkflowExecutionCancelRequestedEventAttributes': {
    'workflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'initiatedEventId': 123
},
'requestCancelExternalWorkflowExecutionInitiatedEventAttributes': {
    'workflowId': 'string',
    'runId': 'string',
    'decisionTaskCompletedEventId': 123,
    'control': 'string'
},
'requestCancelExternalWorkflowExecutionFailedEventAttributes': {
    'workflowId': 'string',
    'runId': 'string',
    'cause': 'UNKNOWN_EXTERNAL_WORKFLOW_EXECUTION' | 'REQUEST_CANCEL_EXT
    'initiatedEventId': 123,
    'decisionTaskCompletedEventId': 123,
    'control': 'string'
},
'scheduleActivityTaskFailedEventAttributes': {
    'activityType': {
        'name': 'string',
        'version': 'string'
    },
    'activityId': 'string',
    'cause': 'ACTIVITY_TYPE_DEPRECATED' | 'ACTIVITY_TYPE_DOES_NOT_EXIST'
    'decisionTaskCompletedEventId': 123
},
'requestCancelActivityTaskFailedEventAttributes': {
    'activityId': 'string',
    'cause': 'ACTIVITY_ID_UNKNOWN' | 'OPERATION_NOT_PERMITTED',
    'decisionTaskCompletedEventId': 123
},
'startTimerFailedEventAttributes': {
    'timerId': 'string',
    'cause': 'TIMER_ID_ALREADY_IN_USE' | 'OPEN_TIMERS_LIMIT_EXCEEDED' | 'T
    'decisionTaskCompletedEventId': 123
},
'cancelTimerFailedEventAttributes': {
    'timerId': 'string',
    'cause': 'TIMER_ID_UNKNOWN' | 'OPERATION_NOT_PERMITTED',
    'decisionTaskCompletedEventId': 123
},
'startChildWorkflowExecutionFailedEventAttributes': {
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'cause': 'WORKFLOW_TYPE_DOES_NOT_EXIST' | 'WORKFLOW_TYPE_DEPRECATED'
    'workflowId': 'string',
    'initiatedEventId': 123,
    'decisionTaskCompletedEventId': 123,
    'control': 'string'
},
'lambdaFunctionScheduledEventAttributes': {
    'id': 'string',
```

```

        'name': 'string',
        'input': 'string',
        'startToCloseTimeout': 'string',
        'decisionTaskCompletedEventId': 123
    },
    'lambdaFunctionStartedEventAttributes': {
        'scheduledEventId': 123
    },
    'lambdaFunctionCompletedEventAttributes': {
        'scheduledEventId': 123,
        'startedEventId': 123,
        'result': 'string'
    },
    'lambdaFunctionFailedEventAttributes': {
        'scheduledEventId': 123,
        'startedEventId': 123,
        'reason': 'string',
        'details': 'string'
    },
    'lambdaFunctionTimedOutEventAttributes': {
        'scheduledEventId': 123,
        'startedEventId': 123,
        'timeoutType': 'START_TO_CLOSE'
    },
    'scheduleLambdaFunctionFailedEventAttributes': {
        'id': 'string',
        'name': 'string',
        'cause': 'ID_ALREADY_IN_USE'|'OPEN_LAMBDA_FUNCTIONS_LIMIT_EXCEEDED',
        'decisionTaskCompletedEventId': 123
    },
    'startLambdaFunctionFailedEventAttributes': {
        'scheduledEventId': 123,
        'cause': 'ASSUME_ROLE_FAILED',
        'message': 'string'
    }
},
'nextPageToken': 'string'
}

```

Response Structure

- *(dict)* –

Paginated representation of a workflow history for a workflow execution. This is the up to date, complete and authoritative record of the events related to all tasks and events in the life of the workflow execution.

- **events** (*list*) –

The list of history events.

- * *(dict)* –

Event within a workflow execution. A history event can be one of these types:

- **WorkflowExecutionStarted** : The workflow execution was started.
- **WorkflowExecutionCompleted** : The workflow execution was closed due to successful completion.
- **WorkflowExecutionFailed** : The workflow execution closed

due to a failure.

- **WorkflowExecutionTimedOut** : The workflow execution was closed because a time out was exceeded.
- **WorkflowExecutionCanceled** : The workflow execution was successfully canceled and closed.
- **WorkflowExecutionTerminated** : The workflow execution was terminated.
- **WorkflowExecutionContinuedAsNew** : The workflow execution was closed and a new execution of the same type was created with the same workflowId.
- **WorkflowExecutionCancelRequested** : A request to cancel this workflow execution was made.
- **DecisionTaskScheduled** : A decision task was scheduled for the workflow execution.
- **DecisionTaskStarted** : The decision task was dispatched to a decider.
- **DecisionTaskCompleted** : The decider successfully completed a decision task by calling `RespondDecisionTaskCompleted`.
- **DecisionTaskTimedOut** : The decision task timed out.
- **ActivityTaskScheduled** : An activity task was scheduled for execution.
- **ScheduleActivityTaskFailed** : Failed to process `ScheduleActivityTask` decision. This happens when the decision is not configured properly, for example the activity type specified is not registered.
- **ActivityTaskStarted** : The scheduled activity task was dispatched to a worker.
- **ActivityTaskCompleted** : An activity worker successfully completed an activity task by calling `RespondActivityTaskCompleted`.
- **ActivityTaskFailed** : An activity worker failed an activity task by calling `RespondActivityTaskFailed`.
- **ActivityTaskTimedOut** : The activity task timed out.
- **ActivityTaskCanceled** : The activity task was successfully canceled.
- **ActivityTaskCancelRequested** : A `RequestCancelActivityTask` decision was received by the system.
- **RequestCancelActivityTaskFailed** : Failed to process `RequestCancelActivityTask` decision. This happens when the decision is not configured properly.
- **WorkflowExecutionSignaled** : An external signal was received for the workflow execution.
- **MarkerRecorded** : A marker was recorded in the workflow history as the result of a `RecordMarker` decision.
- **TimerStarted** : A timer was started for the workflow execution due to a `StartTimer` decision.
- **StartTimerFailed** : Failed to process `StartTimer` decision. This happens when the decision is not configured properly, for example a timer already exists with the specified timer ID.
- **TimerFired** : A timer, previously started for this workflow execution, fired.
- **TimerCanceled** : A timer, previously started for this work-

flow execution, was successfully canceled.

- **CancelTimerFailed** : Failed to process CancelTimer decision. This happens when the decision is not configured properly, for example no timer exists with the specified timer ID.
- **StartChildWorkflowExecutionInitiated** : A request was made to start a child workflow execution.
- **StartChildWorkflowExecutionFailed** : Failed to process StartChildWorkflowExecution decision. This happens when the decision is not configured properly, for example the workflow type specified is not registered.
- **ChildWorkflowExecutionStarted** : A child workflow execution was successfully started.
- **ChildWorkflowExecutionCompleted** : A child workflow execution, started by this workflow execution, completed successfully and was closed.
- **ChildWorkflowExecutionFailed** : A child workflow execution, started by this workflow execution, failed to complete successfully and was closed.
- **ChildWorkflowExecutionTimedOut** : A child workflow execution, started by this workflow execution, timed out and was closed.
- **ChildWorkflowExecutionCanceled** : A child workflow execution, started by this workflow execution, was canceled and closed.
- **ChildWorkflowExecutionTerminated** : A child workflow execution, started by this workflow execution, was terminated.
- **SignalExternalWorkflowExecutionInitiated** : A request to signal an external workflow was made.
- **ExternalWorkflowExecutionSignaled** : A signal, requested by this workflow execution, was successfully delivered to the target external workflow execution.
- **SignalExternalWorkflowExecutionFailed** : The request to signal an external workflow execution failed.
- **RequestCancelExternalWorkflowExecutionInitiated** : A request was made to request the cancellation of an external workflow execution.
- **ExternalWorkflowExecutionCancelRequested** : Request to cancel an external workflow execution was successfully delivered to the target execution.
- **RequestCancelExternalWorkflowExecutionFailed** : Request to cancel an external workflow execution failed.
- **LambdaFunctionScheduled** : An AWS Lambda function was scheduled for execution.
- **LambdaFunctionStarted** : The scheduled function was invoked in the AWS Lambda service.
- **LambdaFunctionCompleted** : The AWS Lambda function successfully completed.
- **LambdaFunctionFailed** : The AWS Lambda function execution failed.
- **LambdaFunctionTimedOut** : The AWS Lambda function execution timed out.
- **ScheduleLambdaFunctionFailed** : Failed to process ScheduleLambdaFunction decision. This happens when the work-

flow execution does not have the proper IAM role attached to invoke AWS Lambda functions.

- **StartLambdaFunctionFailed** : Failed to invoke the scheduled function in the AWS Lambda service. This happens when the AWS Lambda service is not available in the current region, or received too many requests.

- **eventTimestamp** (*datetime*) –

The date and time when the event occurred.

- **eventType** (*string*) –

The type of the history event.

- **eventId** (*integer*) –

The system generated ID of the event. This ID uniquely identifies the event within the workflow execution history.

- **workflowExecutionStartedEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionStarted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **input** (*string*) –

The input provided to the workflow execution (if any).

- **executionStartToCloseTimeout** (*string*) –

The maximum duration for this workflow execution.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **taskStartToCloseTimeout** (*string*) –

The maximum duration of decision tasks for this workflow type.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **childPolicy** (*string*) –

The policy to use for the child workflow executions if this workflow execution is terminated, by calling the `TerminateWorkflowExecution` action explicitly or due to an expired timeout.

The supported child policies are:

- **TERMINATE**: the child executions will be terminated.
- **REQUEST_CANCEL**: a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON**: no action will be taken. The child executions will continue to run.
- **taskList** (*dict*) –

The name of the task list for scheduling the decision tasks for this workflow execution.

- **name** (*string*) –

The name of the task list.

- **workflowType** (*dict*) –

The workflow type of this execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **tagList** (*list*) –

The list of tags associated with this workflow execution. An execution can have up to 5 tags.

- (*string*) –

- **taskPriority** (*string*) –

- **continuedExecutionRunId** (*string*) –

If this workflow execution was started due to a `ContinueAsNewWorkflowExecution` decision, then it contains the `runId` of the previous workflow execution that was closed and continued as this execution.

- **parentWorkflowExecution** (*dict*) –

The source workflow execution that started this workflow execution. The member is not set if the workflow execution was not started by a workflow.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **parentInitiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this workflow execution. The source event with this ID can be found in the history of the source workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **lambdaRole** (*string*) –

The IAM role attached to this workflow execution to use when invoking AWS Lambda functions.

- **workflowExecutionCompletedEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionCompleted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **result** (*string*) –

The result produced by the workflow execution upon successful completion.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CompleteWorkflowExecution` decision to complete this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **completeWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `CompleteWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CompleteWorkflowExecution` decision to complete this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **workflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **reason** (*string*) –

The descriptive reason provided for the failure (if any).

- **details** (*string*) –

The details of the failure (if any).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `FailWorkflowExecution` decision to fail this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **failWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `FailWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `FailWorkflowExecution` decision to fail this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **workflowExecutionTimedOutEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionTimedOut` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timeoutType** (*string*) –

The type of timeout that caused this event.

- **childPolicy** (*string*) –

The policy used for the child workflow executions of this workflow execution.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.
- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON:** no action will be taken. The child executions will continue to run.
- **workflowExecutionCanceledEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionCanceled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **details** (*string*) –

Details for the cancellation (if any).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CancelWorkflowExecution` decision for this cancellation request. This information can be useful for diagnosing

problems by tracing back the chain of events leading up to this event.

- **cancelWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `CancelWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CancelWorkflowExecution` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **workflowExecutionContinuedAsNewEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionContinuedAsNew` then this member is set and provides detailed information about the event. It is not set for other event types.

- **input** (*string*) –

The input provided to the new workflow execution.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `ContinueAsNewWorkflowExecution` decision that started this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **newExecutionRunId** (*string*) –

The `runId` of the new workflow execution.

- **executionStartToCloseTimeout** (*string*) –

The total duration allowed for the new workflow execution.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **taskList** (*dict*) –

Represents a task list.

- **name** (*string*) –

The name of the task list.

- **taskPriority** (*string*) –
- **taskStartToCloseTimeout** (*string*) –

The maximum duration of decision tasks for the new workflow execution.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **childPolicy** (*string*) –

The policy to use for the child workflow executions of the new execution if it is terminated by calling the `TerminateWorkflowExecution` action explicitly or due to an expired timeout.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.
- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON:** no action will be taken. The child executions will continue to run.
- **tagList** (*list*) –

The list of tags associated with the new workflow execution.

- (*string*) –
- **workflowType** (*dict*) –

Represents a workflow type.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique within a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique within a domain.

- **lambdaRole** (*string*) –

The IAM role attached to this workflow execution to use when invoking AWS Lambda functions.

- **continueAsNewWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `ContinueAsNewWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `ContinueAsNewWorkflowExecution` decision that started this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **workflowExecutionTerminatedEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionTerminated` then this member is set and provides detailed information about the event. It is not set for other event types.

- **reason** (*string*) –

The reason provided for the termination (if any).

- **details** (*string*) –

The details provided for the termination (if any).

- **childPolicy** (*string*) –

The policy used for the child workflow executions of this workflow execution.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.
- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON:** no action will be taken. The child executions will continue to run.
- **cause** (*string*) –

If set, indicates that the workflow execution was automatically terminated, and specifies the cause. This happens if the parent workflow execution times out or is terminated and the child policy is set to terminate child executions.

- **workflowExecutionCancelRequestedEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionCancelRequested` then this member is set and provides detailed information about the event. It is not set for other event types.

- **externalWorkflowExecution** (*dict*) –

The external workflow execution for which the cancellation was requested.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **externalInitiatedEventId** (*integer*) –

The ID of the `RequestCancelExternalWorkflowExecutionInitiated` event corresponding to the `RequestCancelExternalWorkflowExecution` decision to cancel this workflow execution. The source event with this ID can be found in the history of the source workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **cause** (*string*) –

If set, indicates that the request to cancel the workflow execution was automatically generated, and specifies the cause. This happens if the parent workflow execution times out or is terminated, and the child policy is set to cancel child executions.

- **decisionTaskScheduledEventAttributes** (*dict*) –

If the event is of type `DecisionTaskScheduled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **taskList** (*dict*) –

The name of the task list in which the decision task was scheduled.

- **name** (*string*) –

The name of the task list.

- **taskPriority** (*string*) –

Optional. A task priority that, if set, specifies the priority for this decision task. Valid values are integers that range from Java's `Integer.MIN_VALUE` (-2147483648) to `Integer.MAX_VALUE` (2147483647). Higher numbers indicate higher priority.

For more information about setting task priority, see [Setting Task Priority](#) in the *Amazon Simple Workflow Developer Guide*.

- **startToCloseTimeout** (*string*) –

The maximum duration for this decision task. The task is considered timed out if it does not completed within this duration.

The duration is specified in seconds; an integer greater than or equal to 0. The value "NONE" can be used to specify unlimited duration.

- **decisionTaskStartedEventAttributes** (*dict*) –

If the event is of type `DecisionTaskStarted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **identity** (*string*) –

Identity of the decider making the request. This enables diagnostic tracing when problems arise. The form of this identity is user defined.

- **scheduledEventId** (*integer*) –

The ID of the `DecisionTaskScheduled` event that was recorded when this decision task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **decisionTaskCompletedEventAttributes** (*dict*) –

If the event is of type `DecisionTaskCompleted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **executionContext** (*string*) –

User defined context for the workflow execution.

- **scheduledEventId** (*integer*) –

The ID of the `DecisionTaskScheduled` event that was recorded when this decision task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `DecisionTaskStarted` event recorded when this decision task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **decisionTaskTimedOutEventAttributes** (*dict*) –

If the event is of type `DecisionTaskTimedOut` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timeoutType** (*string*) –

The type of timeout that expired before the decision task could be completed.

- **scheduledEventId** (*integer*) –

The ID of the `DecisionTaskScheduled` event that was recorded when this decision task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `DecisionTaskStarted` event recorded when this decision task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **activityTaskScheduledEventAttributes** (*dict*) –

If the event is of type `ActivityTaskScheduled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **activityType** (*dict*) –

The type of the activity task.

- **name** (*string*) –

The name of this activity.

Note: The combination of activity type name and version must be unique within a domain.

- **version** (*string*) –

The version of this activity.

Note: The combination of activity type name and version must be unique with in a domain.

- **activityId** (*string*) –

The unique ID of the activity task.

- **input** (*string*) –

The input provided to the activity task.

- **control** (*string*) –

Optional. Data attached to the event that can be used by the decider in subsequent workflow tasks. This data is not sent to the activity.

- **scheduleToStartTimeout** (*string*) –

The maximum amount of time the activity task can wait to be assigned to a worker.

- **scheduleToCloseTimeout** (*string*) –

The maximum amount of time for this activity task.

- **startToCloseTimeout** (*string*) –

The maximum amount of time a worker may take to process the activity task.

- **taskList** (*dict*) –

The task list in which the activity task has been scheduled.

- **name** (*string*) –

The name of the task list.

- **taskPriority** (*string*) –

Optional. The priority to assign to the scheduled activity task. If set, this will override any default priority value that was assigned when the activity type was registered.

Valid values are integers that range from Java's `Integer.MIN_VALUE` (-2147483648) to `Integer.MAX_VALUE` (2147483647). Higher numbers indicate higher priority.

For more information about setting task priority, see [Setting Task Priority](#) in the *Amazon Simple Workflow Developer Guide*.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision that resulted in the scheduling of this activity task. This information can be useful for diagnosing

problems by tracing back the chain of events leading up to this event.

- **heartbeatTimeout** (*string*) –

The maximum time before which the worker processing this task must report progress by calling `RecordActivityTaskHeartbeat`. If the timeout is exceeded, the activity task is automatically timed out. If the worker subsequently attempts to record a heartbeat or return a result, it will be ignored.

- **activityTaskStartedEventAttributes** (*dict*) –

If the event is of type `ActivityTaskStarted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **identity** (*string*) –

Identity of the worker that was assigned this task. This aids diagnostics when problems arise. The form of this identity is user defined.

- **scheduledEventId** (*integer*) –

The ID of the `ActivityTaskScheduled` event that was recorded when this activity task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **activityTaskCompletedEventAttributes** (*dict*) –

If the event is of type `ActivityTaskCompleted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **result** (*string*) –

The results of the activity task (if any).

- **scheduledEventId** (*integer*) –

The ID of the `ActivityTaskScheduled` event that was recorded when this activity task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ActivityTaskStarted` event recorded when this activity task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **activityTaskFailedEventAttributes** (*dict*) –

If the event is of type `ActivityTaskFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **reason** (*string*) –

The reason provided for the failure (if any).

- **details** (*string*) –

The details of the failure (if any).

- **scheduledEventId** (*integer*) –

The ID of the `ActivityTaskScheduled` event that was recorded when this activity task was scheduled. This infor-

mation can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ActivityTaskStarted` event recorded when this activity task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **activityTaskTimedOutEventAttributes** (*dict*) –

If the event is of type `ActivityTaskTimedOut` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timeoutType** (*string*) –

The type of the timeout that caused this event.

- **scheduledEventId** (*integer*) –

The ID of the `ActivityTaskScheduled` event that was recorded when this activity task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ActivityTaskStarted` event recorded when this activity task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **details** (*string*) –

Contains the content of the `details` parameter for the last call made by the activity to `RecordActivityTaskHeartbeat`.

- **activityTaskCanceledEventAttributes** (*dict*) –

If the event is of type `ActivityTaskCanceled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **details** (*string*) –

Details of the cancellation (if any).

- **scheduledEventId** (*integer*) –

The ID of the `ActivityTaskScheduled` event that was recorded when this activity task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ActivityTaskStarted` event recorded when this activity task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **latestCancelRequestedEventId** (*integer*) –

If set, contains the ID of the last `ActivityTaskCancelRequested` event recorded for this activity task. This information can be useful for

diagnosing problems by tracing back the chain of events leading up to this event.

- **activityTaskCancelRequestedEventAttributes** (*dict*) –

If the event is of type `ActivityTaskCancelRequested` then this member is set and provides detailed information about the event. It is not set for other event types.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RequestCancelActivityTask` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **activityId** (*string*) –

The unique ID of the task.

- **workflowExecutionSignaledEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionSignaled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **signalName** (*string*) –

The name of the signal received. The decider can use the signal name and inputs to determine how to process the signal.

- **input** (*string*) –

Inputs provided with the signal (if any). The decider can use the signal name and inputs to determine how to process the signal.

- **externalWorkflowExecution** (*dict*) –

The workflow execution that sent the signal. This is set only if the signal was sent by another workflow execution.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **externalInitiatedEventId** (*integer*) –

The ID of the `SignalExternalWorkflowExecutionInitiated` event corresponding to the `SignalExternalWorkflow` decision to signal this workflow execution. The source event with this ID can be found in the history of the source workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event. This field is set only if the signal was initiated by another workflow execution.

- **markerRecordedEventAttributes** (*dict*) –

If the event is of type `MarkerRecorded` then this member

is set and provides detailed information about the event. It is not set for other event types.

- **markerName** (*string*) –

The name of the marker.

- **details** (*string*) –

Details of the marker (if any).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RecordMarker` decision that requested this marker. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **recordMarkerFailedEventAttributes** (*dict*) –

If the event is of type `DecisionTaskFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **markerName** (*string*) –

The marker's name.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RecordMarkerFailed` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **timerStartedEventAttributes** (*dict*) –

If the event is of type `TimerStarted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timerId** (*string*) –

The unique ID of the timer that was started.

- **control** (*string*) –

Optional. Data attached to the event that can be used by the decider in subsequent workflow tasks.

- **startToFireTimeout** (*string*) –

The duration of time after which the timer will fire.

The duration is specified in seconds; an integer greater than or equal to 0.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `StartTimer` decision for this activity task. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **timerFiredEventAttributes** (*dict*) –

If the event is of type `TimerFired` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timerId** (*string*) –

The unique ID of the timer that fired.

- **startedEventId** (*integer*) –

The ID of the `TimerStarted` event that was recorded when this timer was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **timerCanceledEventAttributes** (*dict*) –

If the event is of type `TimerCanceled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timerId** (*string*) –

The unique ID of the timer that was canceled.

- **startedEventId** (*integer*) –

The ID of the `TimerStarted` event that was recorded when this timer was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CancelTimer` decision to cancel this timer. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startChildWorkflowExecutionInitiatedEventAttributes** (*dict*) –

If the event is of type `StartChildWorkflowExecutionInitiated` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowId** (*string*) –

The `workflowId` of the child workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique within a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique within a domain.

- **control** (*string*) –

Optional. Data attached to the event that can be used by the decider in subsequent decision tasks. This data is not sent to the activity.

- **input** (*string*) –

The inputs provided to the child workflow execution (if any).

- **executionStartToCloseTimeout** (*string*) –

The maximum duration for the child workflow execution. If the workflow execution is not closed within this duration, it will be timed out and force terminated.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **taskList** (*dict*) –

The name of the task list used for the decision tasks of the child workflow execution.

- **name** (*string*) –

The name of the task list.

- **taskPriority** (*string*) –

Optional. The priority assigned for the decision tasks for this workflow execution. Valid values are integers that range from Java’s `Integer.MIN_VALUE` (-2147483648) to `Integer.MAX_VALUE` (2147483647). Higher numbers indicate higher priority.

For more information about setting task priority, see [Setting Task Priority](#) in the *Amazon Simple Workflow Developer Guide*.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `StartChildWorkflowExecution` decision to request this child workflow execution. This information can be useful for diagnosing problems by tracing back the cause of events.

- **childPolicy** (*string*) –

The policy to use for the child workflow executions if this execution gets terminated by explicitly calling the `TerminateWorkflowExecution` action or due to an expired timeout.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.
- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in

its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.

- **ABANDON:** no action will be taken. The child executions will continue to run.
- **taskStartToCloseTimeout** (*string*) –

The maximum duration allowed for the decision tasks for this workflow execution.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **tagList** (*list*) –

The list of tags to associated with the child workflow execution.

- (*string*) –
- **lambdaRole** (*string*) –

The IAM role attached to this workflow execution to use when invoking AWS Lambda functions.

- **childWorkflowExecutionStartedEventAttributes** (*dict*) –

If the event is of type `ChildWorkflowExecutionStarted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that was started.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful

for diagnosing problems by tracing back the chain of events leading up to this event.

- **childWorkflowExecutionCompletedEventAttributes** (*dict*) –

–

If the event is of type `ChildWorkflowExecutionCompleted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that was completed.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **result** (*string*) –

The result of the child workflow execution (if any).

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ChildWorkflowExecutionStarted` event recorded when this child workflow execution was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **childWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `ChildWorkflowExecutionFailed` then this member

is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that failed.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **reason** (*string*) –

The reason for the failure (if provided).

- **details** (*string*) –

The details of the failure (if provided).

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ChildWorkflowExecutionStarted` event recorded when this child workflow execution was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **childWorkflowExecutionTimedOutEventAttributes** (*dict*) –

If the event is of type `ChildWorkflowExecutionTimedOut` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that timed out.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **timeoutType** (*string*) –

The type of the timeout that caused the child workflow execution to time out.

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ChildWorkflowExecutionStarted` event recorded when this child workflow execution was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **childWorkflowExecutionCanceledEventAttributes** (*dict*) –

If the event is of type `ChildWorkflowExecutionCanceled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that was canceled.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **details** (*string*) –

Details of the cancellation (if provided).

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ChildWorkflowExecutionStarted` event recorded when this child workflow execution was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **childWorkflowExecutionTerminatedEventAttributes** (*dict*) –

If the event is of type `ChildWorkflowExecutionTerminated` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that was terminated.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version

must be unique with in a domain.

-
- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

-
- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ChildWorkflowExecutionStarted` event recorded when this child workflow execution was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **signalExternalWorkflowExecutionInitiatedEventAttributes** (*dict*) –

If the event is of type `SignalExternalWorkflowExecutionInitiated` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowId** (*string*) –

The `workflowId` of the external workflow execution.

- **runId** (*string*) –

The `runId` of the external workflow execution to send the signal to.

- **signalName** (*string*) –

The name of the signal.

- **input** (*string*) –

Input provided to the signal (if any).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `SignalExternalWorkflowExecution` decision for this signal. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **control** (*string*) –

Optional. data attached to the event that can be used by the decider in subsequent decision tasks.

- **externalWorkflowExecutionSignaledEventAttributes** (*dict*) –

If the event is of type

`ExternalWorkflowExecutionSignaled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **`workflowExecution`** (*dict*) –

The external workflow execution that the signal was delivered to.

- **`workflowId`** (*string*) –

The user defined identifier associated with the workflow execution.

- **`runId`** (*string*) –

A system-generated unique identifier for the workflow execution.

- **`initiatedEventId`** (*integer*) –

The ID of the `SignalExternalWorkflowExecutionInitiated` event corresponding to the `SignalExternalWorkflowExecution` decision to request this signal. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **`signalExternalWorkflowExecutionFailedEventAttributes`** (*dict*) –

If the event is of type `SignalExternalWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **`workflowId`** (*string*) –

The `workflowId` of the external workflow execution that the signal was being delivered to.

- **`runId`** (*string*) –

The `runId` of the external workflow execution that the signal was being delivered to.

- **`cause`** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If `cause` is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **`initiatedEventId`** (*integer*) –

The ID of the `SignalExternalWorkflowExecutionInitiated` event corresponding to the `SignalExternalWorkflowExecution` decision to request this signal. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **`decisionTaskCompletedEventId`** (*integer*) –

The ID of the `DecisionTaskCompleted` event cor-

responding to the decision task that resulted in the `SignalExternalWorkflowExecution` decision for this signal. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **control** (*string*) –
- **externalWorkflowExecutionCancelRequestedEventAttributes** (*dict*) –

If the event is of type `ExternalWorkflowExecutionCancelRequested` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The external workflow execution to which the cancellation request was delivered.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **initiatedEventId** (*integer*) –

The ID of the `RequestCancelExternalWorkflowExecutionInitiated` event corresponding to the `RequestCancelExternalWorkflowExecution` decision to cancel this external workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **requestCancelExternalWorkflowExecutionInitiatedEventAttributes** (*dict*) –

If the event is of type `RequestCancelExternalWorkflowExecutionInitiated` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowId** (*string*) –

The `workflowId` of the external workflow execution to be canceled.

- **runId** (*string*) –

The `runId` of the external workflow execution to be canceled.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RequestCancelExternalWorkflowExecution` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **control** (*string*) –

Optional. Data attached to the event that can be used by the decider in subsequent workflow tasks.

- **requestCancelExternalWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `RequestCancelExternalWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowId** (*string*) –

The `workflowId` of the external workflow to which the cancel request was to be delivered.

- **runId** (*string*) –

The `runId` of the external workflow execution.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **initiatedEventId** (*integer*) –

The ID of the `RequestCancelExternalWorkflowExecutionInitiated` event corresponding to the `RequestCancelExternalWorkflowExecution` decision to cancel this external workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RequestCancelExternalWorkflowExecution` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **control** (*string*) –

- **scheduleActivityTaskFailedEventAttributes** (*dict*) –

If the event is of type `ScheduleActivityTaskFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **activityType** (*dict*) –

The activity type provided in the `ScheduleActivityTask` decision that failed.

- **name** (*string*) –

The name of this activity.

Note: The combination of activity type name and version must be unique within a domain.

- **version** (*string*) –

The version of this activity.

Note: The combination of activity type name and version must be unique within a domain.

- **activityId** (*string*) –

The activityId provided in the `ScheduleActivityTask` decision that failed.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision that resulted in the scheduling of this activity task. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **requestCancelActivityTaskFailedEventAttributes** (*dict*) –

If the event is of type `RequestCancelActivityTaskFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **activityId** (*string*) –

The activityId provided in the `RequestCancelActivityTask` decision that failed.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RequestCancelActivityTask` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startTimerFailedEventAttributes** (*dict*) –

If the event is of type `StartTimerFailed` then this mem-

ber is set and provides detailed information about the event. It is not set for other event types.

- **timerId** (*string*) –

The timerId provided in the `StartTimer` decision that failed.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `StartTimer` decision for this activity task. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **cancelTimerFailedEventAttributes** (*dict*) –

If the event is of type `CancelTimerFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timerId** (*string*) –

The timerId provided in the `CancelTimer` decision that failed.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CancelTimer` decision to cancel this timer. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startChildWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `StartChildWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowType** (*dict*) –

The workflow type provided in the

`StartChildWorkflowExecution` decision that failed.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **workflowId** (*string*) –

The `workflowId` of the child workflow execution.

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `StartChildWorkflowExecution` decision to request this child workflow execution. This information can be useful for diagnosing problems by tracing back the cause of events.

- **control** (*string*) –

- **lambdaFunctionScheduledEventAttributes** (*dict*) –

Provides details for the `LambdaFunctionScheduled` event.

- **id** (*string*) –

The unique Amazon SWF ID for the AWS Lambda task.

- **name** (*string*) –

The name of the scheduled AWS Lambda function.

- **input** (*string*) –

Input provided to the AWS Lambda function.

- **startToCloseTimeout** (*string*) –

The maximum time, in seconds, that the AWS Lambda function can take to execute from start to close before it is marked as failed.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event for the decision that resulted in the scheduling of this AWS Lambda function. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **lambdaFunctionStartedEventAttributes** (*dict*) –

Provides details for the `LambdaFunctionStarted` event.

- **scheduledEventId** (*integer*) –

The ID of the `LambdaFunctionScheduled` event that was recorded when this AWS Lambda function was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **lambdaFunctionCompletedEventAttributes** (*dict*) –

Provides details for the `LambdaFunctionCompleted` event.

- **scheduledEventId** (*integer*) –

The ID of the `LambdaFunctionScheduled` event that was recorded when this AWS Lambda function was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `LambdaFunctionStarted` event recorded in the history.

- **result** (*string*) –

The result of the function execution (if any).

- **lambdaFunctionFailedEventAttributes** (*dict*) –

Provides details for the `LambdaFunctionFailed` event.

- **scheduledEventId** (*integer*) –

The ID of the `LambdaFunctionScheduled` event that was recorded when this AWS Lambda function was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `LambdaFunctionStarted` event recorded in the history.

- **reason** (*string*) –

The reason provided for the failure (if any).

- **details** (*string*) –

The details of the failure (if any).

- **lambdaFunctionTimedOutEventAttributes** (*dict*) –

Provides details for the `LambdaFunctionTimedOut` event.

- **scheduledEventId** (*integer*) –

The ID of the `LambdaFunctionScheduled` event that was recorded when this AWS Lambda function was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `LambdaFunctionStarted` event recorded in the history.

- **timeoutType** (*string*) –

The type of the timeout that caused this event.

- **scheduleLambdaFunctionFailedEventAttributes** (*dict*) –

Provides details for the `ScheduleLambdaFunctionFailed` event.

- **id** (*string*) –

The unique Amazon SWF ID of the AWS Lambda task.

- **name** (*string*) –

The name of the scheduled AWS Lambda function.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision that resulted in the scheduling of this AWS Lambda function. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startLambdaFunctionFailedEventAttributes** (*dict*) –

Provides details for the `StartLambdaFunctionFailed` event.

- **scheduledEventId** (*integer*) –

The ID of the `LambdaFunctionScheduled` event that was recorded when this AWS Lambda function was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **message** (*string*) –

The error message (if any).

– **nextPageToken** (*string*) –

If a `NextPageToken` was returned by a previous call, there are more results available. To retrieve the next page of results, make the call again using the returned token in `nextPageToken`. Keep all other arguments unchanged.

The configured `maximumPageSize` determines how many results can be returned in a single call.

list_activity_types (***kwargs*)

Returns information about all activities registered in the specified domain that match the specified name and registration status. The result includes information like creation date, current status of the activity, etc. The results may be split into multiple pages. To retrieve subsequent pages, make the call again using the `nextPageToken` returned by the initial call.

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- You cannot use an IAM policy to constrain this action's parameters.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.list_activity_types(
    domain='string',
    name='string',
    registrationStatus='REGISTERED'|'DEPRECATED',
    nextPageToken='string',
    maximumPageSize=123,
    reverseOrder=True|False
)
```

Parameters

- **domain** (*string*) – [REQUIRED]

The name of the domain in which the activity types have been registered.

- **name** (*string*) – If specified, only lists the activity types that have this name.
- **registrationStatus** (*string*) – [REQUIRED]

Specifies the registration status of the activity types to list.

- **nextPageToken** (*string*) – If a `NextPageToken` was returned by a previous call, there are more results available. To retrieve the next page of results, make the call again using the returned token in `nextPageToken`. Keep all other arguments unchanged.

The configured `maximumPageSize` determines how many results can be returned in a single call.

- **maximumPageSize** (*integer*) – The maximum number of results that will be returned per call. `nextPageToken` can be used to obtain further pages of results. The default is 1000, which is the maximum allowed page size. You can, however, specify a page size *smaller* than the maximum.

This is an upper limit only; the actual number of results returned per call may be fewer than the specified maximum.

- **reverseOrder** (*boolean*) – When set to `true`, returns the results in reverse order. By default, the results are returned in ascending alphabetical order by name of the activity types.

Return type dict

Returns

Response Syntax

```
{
  'typeInfos': [
    {
      'activityType': {
        'name': 'string',
        'version': 'string'
      },
      'status': 'REGISTERED'|'DEPRECATED',
      'description': 'string',
      'creationDate': datetime(2015, 1, 1),
      'deprecationDate': datetime(2015, 1, 1)
    },
  ],
  'nextPageToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains a paginated list of activity type information structures.

- **typeInfos** (*list*) –

List of activity type information.

- * (*dict*) –

Detailed information about an activity type.

- **activityType** (*dict*) –

The ActivityType type structure representing the activity type.

- **name** (*string*) –

The name of this activity.

Note: The combination of activity type name and version must be unique within a domain.

- **version** (*string*) –

The version of this activity.

Note: The combination of activity type name and version must be unique with in a domain.

- **status** (*string*) –

The current status of the activity type.

- **description** (*string*) –

The description of the activity type provided in RegisterActivityType .

- **creationDate** (*datetime*) –

The date and time this activity type was created through RegisterActivityType .

- **deprecationDate** (*datetime*) –

If DEPRECATED, the date and time DeprecateActivityType was called.

- **nextPageToken** (*string*) –

If a NextPageToken was returned by a previous call, there are more results available. To retrieve the next page of results, make the call again using the returned token in nextPageToken . Keep all other arguments unchanged.

The configured maximumPageSize determines how many results can be returned in a single call.

list_closed_workflow_executions (***kwargs*)

Returns a list of closed workflow executions in the specified domain that meet the filtering criteria. The results may be split into multiple pages. To retrieve subsequent pages, make the call again using the nextPageToken returned by the initial call.

Note: This operation is eventually consistent. The results are best effort and may not exactly reflect recent updates and changes.

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a Resource element with the domain name to limit the action to only specified domains.
- Use an Action element to allow or deny permission to call this action.
- Constrain the following parameters by using a Condition element with the appropriate keys. *
 - tagFilter.tag : String constraint. The key is swf:tagFilter.tag .
 - typeFilter.name : String constraint. The key is swf:typeFilter.name .
 - typeFilter.version : String constraint. The key is swf:typeFilter.version .

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to OPERATION_NOT_PERMITTED. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#) .

Request Syntax

```
response = client.list_closed_workflow_executions(
    domain='string',
    startTimeFilter={
        'oldestDate': datetime(2015, 1, 1),
        'latestDate': datetime(2015, 1, 1)
    },
    closeTimeFilter={
        'oldestDate': datetime(2015, 1, 1),
        'latestDate': datetime(2015, 1, 1)
    },
    executionFilter={
        'workflowId': 'string'
    },
    closeStatusFilter={
        'status': 'COMPLETED' | 'FAILED' | 'CANCELED' | 'TERMINATED' | 'CONTINUED_AS_NEW' | 'TIMED_OUT'
    },
    )
```



```

typeFilter={
    'name': 'string',
    'version': 'string'
},
tagFilter={
    'tag': 'string'
},
nextPageToken='string',
maximumPageSize=123,
reverseOrder=True|False
)

```

Parameters

- **domain** (*string*) – [REQUIRED]

The name of the domain that contains the workflow executions to list.

- **startTimeFilter** (*dict*) – If specified, the workflow executions are included in the returned results based on whether their start times are within the range specified by this filter. Also, if this parameter is specified, the returned results are ordered by their start times.

Note: `startTimeFilter` and `closeTimeFilter` are mutually exclusive. You must specify one of these in a request but not both.

- **oldestDate** (*datetime*) – [REQUIRED]

Specifies the oldest start or close date and time to return.

- **latestDate** (*datetime*) –

Specifies the latest start or close date and time to return.

- **closeTimeFilter** (*dict*) – If specified, the workflow executions are included in the returned results based on whether their close times are within the range specified by this filter. Also, if this parameter is specified, the returned results are ordered by their close times.

Note: `startTimeFilter` and `closeTimeFilter` are mutually exclusive. You must specify one of these in a request but not both.

- **oldestDate** (*datetime*) – [REQUIRED]

Specifies the oldest start or close date and time to return.

- **latestDate** (*datetime*) –

Specifies the latest start or close date and time to return.

- **executionFilter** (*dict*) – If specified, only workflow executions matching the workflow ID specified in the filter are returned.

Note: `closeStatusFilter`, `executionFilter`, `typeFilter` and `tagFilter` are mutually exclusive. You can specify at most one of these in a request.

- **workflowId** (*string*) – [REQUIRED]

The workflowId to pass of match the criteria of this filter.

- **closeStatusFilter** (*dict*) – If specified, only workflow executions that match this *close status* are listed. For example, if `TERMINATED` is specified, then only `TERMINATED` workflow executions are listed.

Note: `closeStatusFilter`, `executionFilter`, `typeFilter` and `tagFilter` are mutually exclusive. You can specify at most one of these in a request.

– **status** (*string*) – [REQUIRED]

Required. The close status that must match the close status of an execution for it to meet the criteria of this filter.

- **typeFilter** (*dict*) – If specified, only executions of the type specified in the filter are returned.

Note: `closeStatusFilter`, `executionFilter`, `typeFilter` and `tagFilter` are mutually exclusive. You can specify at most one of these in a request.

– **name** (*string*) – [REQUIRED]

Required. Name of the workflow type.

– **version** (*string*) –

Version of the workflow type.

- **tagFilter** (*dict*) – If specified, only executions that have the matching tag are listed.

Note: `closeStatusFilter`, `executionFilter`, `typeFilter` and `tagFilter` are mutually exclusive. You can specify at most one of these in a request.

– **tag** (*string*) – [REQUIRED]

Required. Specifies the tag that must be associated with the execution for it to meet the filter criteria.

- **nextPageToken** (*string*) – If a `NextPageToken` was returned by a previous call, there are more results available. To retrieve the next page of results, make the call again using the returned token in `nextPageToken`. Keep all other arguments unchanged.

The configured `maximumPageSize` determines how many results can be returned in a single call.

- **maximumPageSize** (*integer*) – The maximum number of results that will be returned per call. `nextPageToken` can be used to obtain further pages of results. The default is 1000, which is the maximum allowed page size. You can, however, specify a page size *smaller* than the maximum.

This is an upper limit only; the actual number of results returned per call may be fewer than the specified maximum.

- **reverseOrder** (*boolean*) – When set to `true`, returns the results in reverse order. By default the results are returned in descending order of the start or the close time of the executions.

Return type `dict`

Returns

Response Syntax

```
{
    'executionInfos': [
```

```

{
    'execution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'startTimestamp': datetime(2015, 1, 1),
    'closeTimestamp': datetime(2015, 1, 1),
    'executionStatus': 'OPEN'|'CLOSED',
    'closeStatus': 'COMPLETED'|'FAILED'|'CANCELED'|'TERMINATED'|'CONTINUED',
    'parent': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'tagList': [
        'string',
    ],
    'cancelRequested': True|False
},
],
'nextPageToken': 'string'
}

```

Response Structure

- (*dict*) –

Contains a paginated list of information about workflow executions.

- **executionInfos** (*list*) –

The list of workflow information structures.

- * (*dict*) –

Contains information about a workflow execution.

- **execution** (*dict*) –

The workflow execution this information is about.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique within a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

· **startTimestamp** (*datetime*) –

The time when the execution was started.

· **closeTimestamp** (*datetime*) –

The time when the workflow execution was closed. Set only if the execution status is CLOSED.

· **executionStatus** (*string*) –

The current status of the execution.

· **closeStatus** (*string*) –

If the execution status is closed then this specifies how the execution was closed:

· COMPLETED : the execution was successfully completed.

· CANCELED : the execution was canceled. Cancellation allows the implementation to gracefully clean up before the execution is closed.

· TERMINATED : the execution was force terminated.

· FAILED : the execution failed to complete.

· TIMED_OUT : the execution did not complete in the allotted time and was automatically timed out.

· CONTINUED_AS_NEW : the execution is logically continued. This means the current execution was completed and a new execution was started to carry on the workflow.

· **parent** (*dict*) –

If this workflow execution is a child of another execution then contains the workflow execution that started this execution.

· **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

· **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

· **tagList** (*list*) –

The list of tags associated with the workflow execution. Tags can be used to identify and list workflow executions of interest through the visibility APIs. A workflow execution can have a maximum of 5 tags.

· (*string*) –

· **cancelRequested** (*boolean*) –

Set to true if a cancellation is requested for this workflow execution.

– **nextPageToken** (*string*) –

If a `NextPageToken` was returned by a previous call, there are more results available. To retrieve the next page of results, make the call again using the returned token in `nextPageToken`. Keep all other arguments unchanged.

The configured `maximumPageSize` determines how many results can be returned in a single call.

list_domains (**kwargs)

Returns the list of domains registered in the account. The results may be split into multiple pages. To retrieve subsequent pages, make the call again using the `nextPageToken` returned by the initial call.

Note: This operation is eventually consistent. The results are best effort and may not exactly reflect recent updates and changes.

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains. The element must be set to `arn:aws:swf::AccountID:domain/*`, where *AccountID* is the account ID, with no dashes.
- Use an `Action` element to allow or deny permission to call this action.
- You cannot use an IAM policy to constrain this action's parameters.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.list_domains(
    nextPageToken='string',
    registrationStatus='REGISTERED'|'DEPRECATED',
    maximumPageSize=123,
    reverseOrder=True|False
)
```

Parameters

- **nextPageToken** (*string*) – If a `NextPageToken` was returned by a previous call, there are more results available. To retrieve the next page of results, make the call again using the returned token in `nextPageToken`. Keep all other arguments unchanged.

The configured `maximumPageSize` determines how many results can be returned in a single call.

- **registrationStatus** (*string*) – **[REQUIRED]**

Specifies the registration status of the domains to list.

- **maximumPageSize** (*integer*) – The maximum number of results that will be returned per call. `nextPageToken` can be used to obtain further pages of results. The default is 1000, which is the maximum allowed page size. You can, however, specify a page size *smaller* than the maximum.

This is an upper limit only; the actual number of results returned per call may be fewer than the specified maximum.

- **reverseOrder** (*boolean*) – When set to `true`, returns the results in reverse order. By default, the results are returned in ascending alphabetical order by name of the domains.

Return type dict

Returns**Response Syntax**

```
{
    'domainInfos': [
```

```
{
    {
        'name': 'string',
        'status': 'REGISTERED'|'DEPRECATED',
        'description': 'string'
    },
    ],
    'nextPageToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains a paginated collection of DomainInfo structures.

- **domainInfos** (*list*) –

A list of DomainInfo structures.

- * (*dict*) –

Contains general information about a domain.

- **name** (*string*) –

The name of the domain. This name is unique within the account.

- **status** (*string*) –

The status of the domain:

- **REGISTERED** : The domain is properly registered and available. You can use this domain for registering types and creating new workflow executions.
- **DEPRECATED** : The domain was deprecated using `DeprecateDomain` , but is still in use. You should not create new workflow executions in this domain.
- **description** (*string*) –

The description of the domain provided through `RegisterDomain` .

- **nextPageToken** (*string*) –

If a `NextPageToken` was returned by a previous call, there are more results available. To retrieve the next page of results, make the call again using the returned token in `nextPageToken` . Keep all other arguments unchanged.

The configured `maximumPageSize` determines how many results can be returned in a single call.

list_open_workflow_executions (***kwargs*)

Returns a list of open workflow executions in the specified domain that meet the filtering criteria. The results may be split into multiple pages. To retrieve subsequent pages, make the call again using the `nextPageToken` returned by the initial call.

Note: This operation is eventually consistent. The results are best effort and may not exactly reflect recent updates and changes.

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.

- Constrain the following parameters by using a `Condition` element with the appropriate keys. *
`tagFilter.tag`: String constraint. The key is `swf:tagFilter.tag`.
- `typeFilter.name`: String constraint. The key is `swf:typeFilter.name`.
- `typeFilter.version`: String constraint. The key is `swf:typeFilter.version`.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.list_open_workflow_executions(
    domain='string',
    startTimeFilter={
        'oldestDate': datetime(2015, 1, 1),
        'latestDate': datetime(2015, 1, 1)
    },
    typeFilter={
        'name': 'string',
        'version': 'string'
    },
    tagFilter={
        'tag': 'string'
    },
    nextPageToken='string',
    maximumPageSize=123,
    reverseOrder=True|False,
    executionFilter={
        'workflowId': 'string'
    }
)
```

Parameters

- **domain** (*string*) – [REQUIRED]

The name of the domain that contains the workflow executions to list.

- **startTimeFilter** (*dict*) – [REQUIRED]

Workflow executions are included in the returned results based on whether their start times are within the range specified by this filter.

- **oldestDate** (*datetime*) – [REQUIRED]

Specifies the oldest start or close date and time to return.

- **latestDate** (*datetime*) –

Specifies the latest start or close date and time to return.

- **typeFilter** (*dict*) – If specified, only executions of the type specified in the filter are returned.

Note: `executionFilter`, `typeFilter` and `tagFilter` are mutually exclusive. You can specify at most one of these in a request.

- **name** (*string*) – [REQUIRED]

Required. Name of the workflow type.

- **version** (*string*) –

Version of the workflow type.

- **tagFilter** (*dict*) – If specified, only executions that have the matching tag are listed.

Note: `executionFilter`, `typeFilter` and `tagFilter` are mutually exclusive. You can specify at most one of these in a request.

– **tag** (*string*) – [REQUIRED]

Required. Specifies the tag that must be associated with the execution for it to meet the filter criteria.

- **nextPageToken** (*string*) – If a `NextPageToken` was returned by a previous call, there are more results available. To retrieve the next page of results, make the call again using the returned token in `nextPageToken`. Keep all other arguments unchanged.

The configured `maximumPageSize` determines how many results can be returned in a single call.

- **maximumPageSize** (*integer*) – The maximum number of results that will be returned per call. `nextPageToken` can be used to obtain further pages of results. The default is 1000, which is the maximum allowed page size. You can, however, specify a page size *smaller* than the maximum.

This is an upper limit only; the actual number of results returned per call may be fewer than the specified maximum.

- **reverseOrder** (*boolean*) – When set to `true`, returns the results in reverse order. By default the results are returned in descending order of the start time of the executions.
- **executionFilter** (*dict*) – If specified, only workflow executions matching the workflow ID specified in the filter are returned.

Note: `executionFilter`, `typeFilter` and `tagFilter` are mutually exclusive. You can specify at most one of these in a request.

– **workflowId** (*string*) – [REQUIRED]

The `workflowId` to pass of match the criteria of this filter.

Return type `dict`

Returns

Response Syntax

```
{
  'executionInfos': [
    {
      'execution': {
        'workflowId': 'string',
        'runId': 'string'
      },
      'workflowType': {
        'name': 'string',
        'version': 'string'
      },
      'startTimestamp': datetime(2015, 1, 1),
      'closeTimestamp': datetime(2015, 1, 1),
      'executionStatus': 'OPEN'|'CLOSED',
      'closeStatus': 'COMPLETED'|'FAILED'|'CANCELED'|'TERMINATED'|'CONTINUED',
      'parent': {
```



```

        'workflowId': 'string',
        'runId': 'string'
    },
    'tagList': [
        'string',
    ],
    'cancelRequested': True|False
},
],
'nextPageToken': 'string'
}

```

Response Structure

- (*dict*) –

Contains a paginated list of information about workflow executions.

- **executionInfos** (*list*) –

The list of workflow information structures.

- * (*dict*) –

Contains information about a workflow execution.

- **execution** (*dict*) –

The workflow execution this information is about.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **startTimestamp** (*datetime*) –

The time when the execution was started.

- **closeTimestamp** (*datetime*) –

The time when the workflow execution was closed. Set only if the execution status is CLOSED.

- **executionStatus** (*string*) –

The current status of the execution.

- **closeStatus** (*string*) –

If the execution status is closed then this specifies how the execution was closed:

- **COMPLETED** : the execution was successfully completed.
- **CANCELED** : the execution was canceled. Cancellation allows the implementation to gracefully clean up before the execution is closed.
- **TERMINATED** : the execution was force terminated.
- **FAILED** : the execution failed to complete.
- **TIMED_OUT** : the execution did not complete in the allotted time and was automatically timed out.
- **CONTINUED_AS_NEW** : the execution is logically continued. This means the current execution was completed and a new execution was started to carry on the workflow.

- **parent** (*dict*) –

If this workflow execution is a child of another execution then contains the workflow execution that started this execution.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **tagList** (*list*) –

The list of tags associated with the workflow execution. Tags can be used to identify and list workflow executions of interest through the visibility APIs. A workflow execution can have a maximum of 5 tags.

- (*string*) –

- **cancelRequested** (*boolean*) –

Set to true if a cancellation is requested for this workflow execution.

- **nextPageToken** (*string*) –

If a `NextPageToken` was returned by a previous call, there are more results available. To retrieve the next page of results, make the call again using the returned token in `nextPageToken`. Keep all other arguments unchanged.

The configured `maximumPageSize` determines how many results can be returned in a single call.

list_workflow_types (***kwargs*)

Returns information about workflow types in the specified domain. The results may be split into multiple pages that can be retrieved by making the call repeatedly.

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- You cannot use an IAM policy to constrain this action's parameters.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to

OPERATION_NOT_PERMITTED. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.list_workflow_types(
    domain='string',
    name='string',
    registrationStatus='REGISTERED'|'DEPRECATED',
    nextPageToken='string',
    maximumPageSize=123,
    reverseOrder=True|False
)
```

Parameters

- **domain** (*string*) – [REQUIRED]

The name of the domain in which the workflow types have been registered.

- **name** (*string*) – If specified, lists the workflow type with this name.
- **registrationStatus** (*string*) – [REQUIRED]

Specifies the registration status of the workflow types to list.

- **nextPageToken** (*string*) – If a `NextPageToken` was returned by a previous call, there are more results available. To retrieve the next page of results, make the call again using the returned token in `nextPageToken`. Keep all other arguments unchanged.

The configured `maximumPageSize` determines how many results can be returned in a single call.

- **maximumPageSize** (*integer*) – The maximum number of results that will be returned per call. `nextPageToken` can be used to obtain further pages of results. The default is 1000, which is the maximum allowed page size. You can, however, specify a page size *smaller* than the maximum.

This is an upper limit only; the actual number of results returned per call may be fewer than the specified maximum.

- **reverseOrder** (*boolean*) – When set to `true`, returns the results in reverse order. By default the results are returned in ascending alphabetical order of the name of the workflow types.

Return type dict

Returns

Response Syntax

```
{
  'typeInfos': [
    {
      'workflowType': {
        'name': 'string',
        'version': 'string'
      },
      'status': 'REGISTERED'|'DEPRECATED',
      'description': 'string',
      'creationDate': datetime(2015, 1, 1),
      'deprecationDate': datetime(2015, 1, 1)
    },
  ],
  'nextPageToken': 'string'
}
```

Response Structure

- *(dict)* –

Contains a paginated list of information structures about workflow types.

- **typeInfos** (*list*) –

The list of workflow type information.

- * *(dict)* –

Contains information about a workflow type.

- **workflowType** (*dict*) –

The workflow type this information is about.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **status** (*string*) –

The current status of the workflow type.

- **description** (*string*) –

The description of the type registered through RegisterWorkflowType .

- **creationDate** (*datetime*) –

The date when this type was registered.

- **deprecationDate** (*datetime*) –

If the type is in deprecated state, then it is set to the date when the type was deprecated.

- **nextPageToken** (*string*) –

If a NextPageToken was returned by a previous call, there are more results available. To retrieve the next page of results, make the call again using the returned token in nextPageToken . Keep all other arguments unchanged.

The configured maximumPageSize determines how many results can be returned in a single call.

poll_for_activity_task (***kwargs*)

Used by workers to get an ActivityTask from the specified activity taskList . This initiates a long poll, where the service holds the HTTP connection open and responds as soon as a task becomes available. The maximum time the service holds on to the request before responding is 60 seconds. If no task is available within 60 seconds, the poll will return an empty result. An empty result, in this context, means that an ActivityTask is returned, but that the value of taskToken is an empty string. If a task is returned, the worker should use its type to identify and process it correctly.

Warning: Workers should set their client side socket timeout to at least 70 seconds (10 seconds higher than the maximum time service may hold the poll request).

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- Constrain the `taskList.name` parameter by using a **Condition** element with the `swf:taskList.name` key to allow the action to access only certain task lists.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.poll_for_activity_task(
    domain='string',
    taskList={
        'name': 'string'
    },
    identity='string'
)
```

Parameters

- **domain** (*string*) – [REQUIRED]

The name of the domain that contains the task lists being polled.

- **taskList** (*dict*) – [REQUIRED]

Specifies the task list to poll for activity tasks.

The specified string must not start or end with whitespace. It must not contain a : (colon), / (slash), | (vertical bar), or any control characters (u0000-u001f | u007f - u009f). Also, it must not contain the literal string quotarnquot.

- **name** (*string*) – [REQUIRED]

The name of the task list.

- **identity** (*string*) – Identity of the worker making the request, recorded in the `ActivityTaskStarted` event in the workflow history. This enables diagnostic tracing when problems arise. The form of this identity is user defined.

Return type dict

Returns

Response Syntax

```
{
    'taskToken': 'string',
    'activityId': 'string',
    'startedEventId': 123,
    'workflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'activityType': {
        'name': 'string',
        'version': 'string'
    }
}
```

```
    },  
    'input': 'string'  
}
```

Response Structure

- (*dict*) –

Unit of work sent to an activity worker.

- **taskToken** (*string*) –

The opaque string used as a handle on the task. This token is used by workers to communicate progress and response information back to the system about the task.

- **activityId** (*string*) –

The unique ID of the task.

- **startedEventId** (*integer*) –

The ID of the `ActivityTaskStarted` event recorded in the history.

- **workflowExecution** (*dict*) –

The workflow execution that started this activity task.

- * **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- * **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **activityType** (*dict*) –

The type of this activity task.

- * **name** (*string*) –

The name of this activity.

Note: The combination of activity type name and version must be unique within a domain.

- * **version** (*string*) –

The version of this activity.

Note: The combination of activity type name and version must be unique with in a domain.

- **input** (*string*) –

The inputs provided when the activity task was scheduled. The form of the input is user defined and should be meaningful to the activity implementation.

poll_for_decision_task (***kwargs*)

Used by deciders to get a `DecisionTask` from the specified decision `taskList`. A decision task may be returned for any open workflow execution that is using the specified task list. The task includes a paginated view of the history of the workflow execution. The decider should use the workflow type and the history to determine how to properly handle the task.

This action initiates a long poll, where the service holds the HTTP connection open and responds as soon a task becomes available. If no decision task is available in the specified task list before the timeout of 60 seconds expires, an empty result is returned. An empty result, in this context, means that a `DecisionTask` is returned, but that the value of `taskToken` is an empty string.

Warning: Deciders should set their client-side socket timeout to at least 70 seconds (10 seconds higher than the timeout).

Warning: Because the number of workflow history events for a single workflow execution might be very large, the result returned might be split up across a number of pages. To retrieve subsequent pages, make additional calls to `PollForDecisionTask` using the `nextPageToken` returned by the initial call. Note that you do **not** call `GetWorkflowExecutionHistory` with this `nextPageToken`. Instead, call `PollForDecisionTask` again.

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- Constrain the `taskList.name` parameter by using a **Condition** element with the `swf:taskList.name` key to allow the action to access only certain task lists.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.poll_for_decision_task(
    domain='string',
    taskList={
        'name': 'string'
    },
    identity='string',
    nextPageToken='string',
    maximumPageSize=123,
    reverseOrder=True|False
)
```

Parameters

- **domain** (*string*) – [REQUIRED]

The name of the domain containing the task lists to poll.

- **taskList** (*dict*) – [REQUIRED]

Specifies the task list to poll for decision tasks.

The specified string must not start or end with whitespace. It must not contain a `:` (colon), `/` (slash), `|` (vertical bar), or any control characters (u0000-u001f | u007f - u009f). Also, it must not contain the literal string `quotarnquot`.

- **name** (*string*) – [REQUIRED]

The name of the task list.

- **identity** (*string*) – Identity of the decider making the request, which is recorded in the `DecisionTaskStarted` event in the workflow history. This enables diagnostic tracing when problems arise. The form of this identity is user defined.
- **nextPageToken** (*string*) – If a `NextPageToken` was returned by a previous call, there are more results available. To retrieve the next page of results, make the call again using the returned token in `nextPageToken`. Keep all other arguments unchanged.

The configured `maximumPageSize` determines how many results can be returned in a single call.

Note: The `nextPageToken` returned by this action cannot be used with `GetWorkflowExecutionHistory` to get the next page. You must call `PollForDecisionTask` again (with the `nextPageToken`) to retrieve the next page of history records. Calling `PollForDecisionTask` with a `nextPageToken` will not return a new decision task.

- **maximumPageSize** (*integer*) – The maximum number of results that will be returned per call. `nextPageToken` can be used to obtain further pages of results. The default is 1000, which is the maximum allowed page size. You can, however, specify a page size *smaller* than the maximum.

This is an upper limit only; the actual number of results returned per call may be fewer than the specified maximum.

- **reverseOrder** (*boolean*) – When set to `true`, returns the events in reverse order. By default the results are returned in ascending order of the `eventTimestamp` of the events.

Return type dict

Returns

Response Syntax

```
{
  'taskToken': 'string',
  'startedEventId': 123,
  'workflowExecution': {
    'workflowId': 'string',
    'runId': 'string'
  },
  'workflowType': {
    'name': 'string',
    'version': 'string'
  },
  'events': [
    {
      'eventTimestamp': datetime(2015, 1, 1),
      'eventType': 'WorkflowExecutionStarted'|'WorkflowExecutionCancelRequest',
      'eventId': 123,
      'workflowExecutionStartedEventAttributes': {
        'input': 'string',
        'executionStartToCloseTimeout': 'string',
        'taskStartToCloseTimeout': 'string',
        'childPolicy': 'TERMINATE'|'REQUEST_CANCEL'|'ABANDON',
        'taskList': {
          'name': 'string'
        },
        'workflowType': {
          'name': 'string',
          'version': 'string'
        },
        'tagList': [
          'string',
        ],
        'taskPriority': 'string',
      },
    },
  ],
}
```



```

        'continuedExecutionRunId': 'string',
        'parentWorkflowExecution': {
            'workflowId': 'string',
            'runId': 'string'
        },
        'parentInitiatedEventId': 123,
        'lambdaRole': 'string'
    },
    'workflowExecutionCompletedEventAttributes': {
        'result': 'string',
        'decisionTaskCompletedEventId': 123
    },
    'completeWorkflowExecutionFailedEventAttributes': {
        'cause': 'UNHANDLED_DECISION'|'OPERATION_NOT_PERMITTED',
        'decisionTaskCompletedEventId': 123
    },
    'workflowExecutionFailedEventAttributes': {
        'reason': 'string',
        'details': 'string',
        'decisionTaskCompletedEventId': 123
    },
    'failWorkflowExecutionFailedEventAttributes': {
        'cause': 'UNHANDLED_DECISION'|'OPERATION_NOT_PERMITTED',
        'decisionTaskCompletedEventId': 123
    },
    'workflowExecutionTimedOutEventAttributes': {
        'timeoutType': 'START_TO_CLOSE',
        'childPolicy': 'TERMINATE'|'REQUEST_CANCEL'|'ABANDON'
    },
    'workflowExecutionCanceledEventAttributes': {
        'details': 'string',
        'decisionTaskCompletedEventId': 123
    },
    'cancelWorkflowExecutionFailedEventAttributes': {
        'cause': 'UNHANDLED_DECISION'|'OPERATION_NOT_PERMITTED',
        'decisionTaskCompletedEventId': 123
    },
    'workflowExecutionContinuedAsNewEventAttributes': {
        'input': 'string',
        'decisionTaskCompletedEventId': 123,
        'newExecutionRunId': 'string',
        'executionStartToCloseTimeout': 'string',
        'taskList': {
            'name': 'string'
        },
        'taskPriority': 'string',
        'taskStartToCloseTimeout': 'string',
        'childPolicy': 'TERMINATE'|'REQUEST_CANCEL'|'ABANDON',
        'tagList': [
            'string',
        ],
        'workflowType': {
            'name': 'string',
            'version': 'string'
        },
        'lambdaRole': 'string'
    },
    'continueAsNewWorkflowExecutionFailedEventAttributes': {

```

```
        'cause': 'UNHANDLED_DECISION'|'WORKFLOW_TYPE_DEPRECATED'|'WORKFLOW_TYPE_DEPRECATED',
        'decisionTaskCompletedEventId': 123
    },
    'workflowExecutionTerminatedEventAttributes': {
        'reason': 'string',
        'details': 'string',
        'childPolicy': 'TERMINATE'|'REQUEST_CANCEL'|'ABANDON',
        'cause': 'CHILD_POLICY_APPLIED'|'EVENT_LIMIT_EXCEEDED'|'OPERATOR_TIMEOUT'
    },
    'workflowExecutionCancelRequestedEventAttributes': {
        'externalWorkflowExecution': {
            'workflowId': 'string',
            'runId': 'string'
        },
        'externalInitiatedEventId': 123,
        'cause': 'CHILD_POLICY_APPLIED'
    },
    'decisionTaskScheduledEventAttributes': {
        'taskList': {
            'name': 'string'
        },
        'taskPriority': 'string',
        'startToCloseTimeout': 'string'
    },
    'decisionTaskStartedEventAttributes': {
        'identity': 'string',
        'scheduledEventId': 123
    },
    'decisionTaskCompletedEventAttributes': {
        'executionContext': 'string',
        'scheduledEventId': 123,
        'startedEventId': 123
    },
    'decisionTaskTimedOutEventAttributes': {
        'timeoutType': 'START_TO_CLOSE',
        'scheduledEventId': 123,
        'startedEventId': 123
    },
    'activityTaskScheduledEventAttributes': {
        'activityType': {
            'name': 'string',
            'version': 'string'
        },
        'activityId': 'string',
        'input': 'string',
        'control': 'string',
        'scheduleToStartTimeout': 'string',
        'scheduleToCloseTimeout': 'string',
        'startToCloseTimeout': 'string',
        'taskList': {
            'name': 'string'
        },
        'taskPriority': 'string',
        'decisionTaskCompletedEventId': 123,
        'heartbeatTimeout': 'string'
    },
    'activityTaskStartedEventAttributes': {
        'identity': 'string',
```

```

        'scheduledEventId': 123
    },
    'activityTaskCompletedEventAttributes': {
        'result': 'string',
        'scheduledEventId': 123,
        'startedEventId': 123
    },
    'activityTaskFailedEventAttributes': {
        'reason': 'string',
        'details': 'string',
        'scheduledEventId': 123,
        'startedEventId': 123
    },
    'activityTaskTimedOutEventAttributes': {
        'timeoutType': 'START_TO_CLOSE'|'SCHEDULE_TO_START'|'SCHEDULE_TO_C
        'scheduledEventId': 123,
        'startedEventId': 123,
        'details': 'string'
    },
    'activityTaskCanceledEventAttributes': {
        'details': 'string',
        'scheduledEventId': 123,
        'startedEventId': 123,
        'latestCancelRequestedEventId': 123
    },
    'activityTaskCancelRequestedEventAttributes': {
        'decisionTaskCompletedEventId': 123,
        'activityId': 'string'
    },
    'workflowExecutionSignaledEventAttributes': {
        'signalName': 'string',
        'input': 'string',
        'externalWorkflowExecution': {
            'workflowId': 'string',
            'runId': 'string'
        },
        'externalInitiatedEventId': 123
    },
    'markerRecordedEventAttributes': {
        'markerName': 'string',
        'details': 'string',
        'decisionTaskCompletedEventId': 123
    },
    'recordMarkerFailedEventAttributes': {
        'markerName': 'string',
        'cause': 'OPERATION_NOT_PERMITTED',
        'decisionTaskCompletedEventId': 123
    },
    'timerStartedEventAttributes': {
        'timerId': 'string',
        'control': 'string',
        'startToFireTimeout': 'string',
        'decisionTaskCompletedEventId': 123
    },
    'timerFiredEventAttributes': {
        'timerId': 'string',
        'startedEventId': 123
    },

```

```
'timerCanceledEventAttributes': {
    'timerId': 'string',
    'startedEventId': 123,
    'decisionTaskCompletedEventId': 123
},
'startChildWorkflowExecutionInitiatedEventAttributes': {
    'workflowId': 'string',
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'control': 'string',
    'input': 'string',
    'executionStartToCloseTimeout': 'string',
    'taskList': {
        'name': 'string'
    },
    'taskPriority': 'string',
    'decisionTaskCompletedEventId': 123,
    'childPolicy': 'TERMINATE'|'REQUEST_CANCEL'|'ABANDON',
    'taskStartToCloseTimeout': 'string',
    'tagList': [
        'string',
    ],
    'lambdaRole': 'string'
},
'childWorkflowExecutionStartedEventAttributes': {
    'workflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'initiatedEventId': 123
},
'childWorkflowExecutionCompletedEventAttributes': {
    'workflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'result': 'string',
    'initiatedEventId': 123,
    'startedEventId': 123
},
'childWorkflowExecutionFailedEventAttributes': {
    'workflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'workflowType': {
        'name': 'string',
        'version': 'string'
    }
```

```

    },
    'reason': 'string',
    'details': 'string',
    'initiatedEventId': 123,
    'startedEventId': 123
  },
  'childWorkflowExecutionTimedOutEventAttributes': {
    'workflowExecution': {
      'workflowId': 'string',
      'runId': 'string'
    },
    'workflowType': {
      'name': 'string',
      'version': 'string'
    },
    'timeoutType': 'START_TO_CLOSE',
    'initiatedEventId': 123,
    'startedEventId': 123
  },
  'childWorkflowExecutionCanceledEventAttributes': {
    'workflowExecution': {
      'workflowId': 'string',
      'runId': 'string'
    },
    'workflowType': {
      'name': 'string',
      'version': 'string'
    },
    'details': 'string',
    'initiatedEventId': 123,
    'startedEventId': 123
  },
  'childWorkflowExecutionTerminatedEventAttributes': {
    'workflowExecution': {
      'workflowId': 'string',
      'runId': 'string'
    },
    'workflowType': {
      'name': 'string',
      'version': 'string'
    },
    'initiatedEventId': 123,
    'startedEventId': 123
  },
  'signalExternalWorkflowExecutionInitiatedEventAttributes': {
    'workflowId': 'string',
    'runId': 'string',
    'signalName': 'string',
    'input': 'string',
    'decisionTaskCompletedEventId': 123,
    'control': 'string'
  },
  'externalWorkflowExecutionSignaledEventAttributes': {
    'workflowExecution': {
      'workflowId': 'string',
      'runId': 'string'
    },
    'initiatedEventId': 123
  }
}

```

```
    },
    'signalExternalWorkflowExecutionFailedEventAttributes': {
        'workflowId': 'string',
        'runId': 'string',
        'cause': 'UNKNOWN_EXTERNAL_WORKFLOW_EXECUTION' | 'SIGNAL_EXTERNAL_WO
        'initiatedEventId': 123,
        'decisionTaskCompletedEventId': 123,
        'control': 'string'
    },
    'externalWorkflowExecutionCancelRequestedEventAttributes': {
        'workflowExecution': {
            'workflowId': 'string',
            'runId': 'string'
        },
        'initiatedEventId': 123
    },
    'requestCancelExternalWorkflowExecutionInitiatedEventAttributes': {
        'workflowId': 'string',
        'runId': 'string',
        'decisionTaskCompletedEventId': 123,
        'control': 'string'
    },
    'requestCancelExternalWorkflowExecutionFailedEventAttributes': {
        'workflowId': 'string',
        'runId': 'string',
        'cause': 'UNKNOWN_EXTERNAL_WORKFLOW_EXECUTION' | 'REQUEST_CANCEL_EXT
        'initiatedEventId': 123,
        'decisionTaskCompletedEventId': 123,
        'control': 'string'
    },
    'scheduleActivityTaskFailedEventAttributes': {
        'activityType': {
            'name': 'string',
            'version': 'string'
        },
        'activityId': 'string',
        'cause': 'ACTIVITY_TYPE_DEPRECATED' | 'ACTIVITY_TYPE_DOES_NOT_EXIST'
        'decisionTaskCompletedEventId': 123
    },
    'requestCancelActivityTaskFailedEventAttributes': {
        'activityId': 'string',
        'cause': 'ACTIVITY_ID_UNKNOWN' | 'OPERATION_NOT_PERMITTED',
        'decisionTaskCompletedEventId': 123
    },
    'startTimerFailedEventAttributes': {
        'timerId': 'string',
        'cause': 'TIMER_ID_ALREADY_IN_USE' | 'OPEN_TIMERS_LIMIT_EXCEEDED' | 'T
        'decisionTaskCompletedEventId': 123
    },
    'cancelTimerFailedEventAttributes': {
        'timerId': 'string',
        'cause': 'TIMER_ID_UNKNOWN' | 'OPERATION_NOT_PERMITTED',
        'decisionTaskCompletedEventId': 123
    },
    'startChildWorkflowExecutionFailedEventAttributes': {
        'workflowType': {
            'name': 'string',
            'version': 'string'
        },
```

```

        },
        'cause': 'WORKFLOW_TYPE_DOES_NOT_EXIST'|'WORKFLOW_TYPE_DEPRECATED',
        'workflowId': 'string',
        'initiatedEventId': 123,
        'decisionTaskCompletedEventId': 123,
        'control': 'string'
    },
    'lambdaFunctionScheduledEventAttributes': {
        'id': 'string',
        'name': 'string',
        'input': 'string',
        'startToCloseTimeout': 'string',
        'decisionTaskCompletedEventId': 123
    },
    'lambdaFunctionStartedEventAttributes': {
        'scheduledEventId': 123
    },
    'lambdaFunctionCompletedEventAttributes': {
        'scheduledEventId': 123,
        'startedEventId': 123,
        'result': 'string'
    },
    'lambdaFunctionFailedEventAttributes': {
        'scheduledEventId': 123,
        'startedEventId': 123,
        'reason': 'string',
        'details': 'string'
    },
    'lambdaFunctionTimedOutEventAttributes': {
        'scheduledEventId': 123,
        'startedEventId': 123,
        'timeoutType': 'START_TO_CLOSE'
    },
    'scheduleLambdaFunctionFailedEventAttributes': {
        'id': 'string',
        'name': 'string',
        'cause': 'ID_ALREADY_IN_USE'|'OPEN_LAMBDA_FUNCTIONS_LIMIT_EXCEEDED',
        'decisionTaskCompletedEventId': 123
    },
    'startLambdaFunctionFailedEventAttributes': {
        'scheduledEventId': 123,
        'cause': 'ASSUME_ROLE_FAILED',
        'message': 'string'
    }
},
'nextPageToken': 'string',
'previousStartedEventId': 123
}

```

Response Structure

- (*dict*) –

A structure that represents a decision task. Decision tasks are sent to deciders in order for them to make decisions.

- **taskToken** (*string*) –

The opaque string used as a handle on the task. This token is used by workers to communicate progress and response information back to the

system about the task.

– **startedEventId** (*integer*) –

The ID of the `DecisionTaskStarted` event recorded in the history.

– **workflowExecution** (*dict*) –

The workflow execution for which this decision task was created.

* **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

* **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

– **workflowType** (*dict*) –

The type of the workflow execution for which this decision task was created.

* **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

* **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

– **events** (*list*) –

A paginated list of history events of the workflow execution. The decider uses this during the processing of the decision task.

* (*dict*) –

Event within a workflow execution. A history event can be one of these types:

- **WorkflowExecutionStarted** : The workflow execution was started.
- **WorkflowExecutionCompleted** : The workflow execution was closed due to successful completion.
- **WorkflowExecutionFailed** : The workflow execution closed due to a failure.
- **WorkflowExecutionTimedOut** : The workflow execution was closed because a time out was exceeded.
- **WorkflowExecutionCanceled** : The workflow execution was successfully canceled and closed.
- **WorkflowExecutionTerminated** : The workflow execution was terminated.
- **WorkflowExecutionContinuedAsNew** : The workflow execution was closed and a new execution of the same type was created with the same workflowId.
- **WorkflowExecutionCancelRequested** : A request to cancel this workflow execution was made.
- **DecisionTaskScheduled** : A decision task was scheduled for the workflow execution.

- **DecisionTaskStarted** : The decision task was dispatched to a decider.
- **DecisionTaskCompleted** : The decider successfully completed a decision task by calling `RespondDecisionTaskCompleted` .
- **DecisionTaskTimedOut** : The decision task timed out.
- **ActivityTaskScheduled** : An activity task was scheduled for execution.
- **ScheduleActivityTaskFailed** : Failed to process `ScheduleActivityTask` decision. This happens when the decision is not configured properly, for example the activity type specified is not registered.
- **ActivityTaskStarted** : The scheduled activity task was dispatched to a worker.
- **ActivityTaskCompleted** : An activity worker successfully completed an activity task by calling `RespondActivityTaskCompleted` .
- **ActivityTaskFailed** : An activity worker failed an activity task by calling `RespondActivityTaskFailed` .
- **ActivityTaskTimedOut** : The activity task timed out.
- **ActivityTaskCanceled** : The activity task was successfully canceled.
- **ActivityTaskCancelRequested** : A `RequestCancelActivityTask` decision was received by the system.
- **RequestCancelActivityTaskFailed** : Failed to process `RequestCancelActivityTask` decision. This happens when the decision is not configured properly.
- **WorkflowExecutionSignaled** : An external signal was received for the workflow execution.
- **MarkerRecorded** : A marker was recorded in the workflow history as the result of a `RecordMarker` decision.
- **TimerStarted** : A timer was started for the workflow execution due to a `StartTimer` decision.
- **StartTimerFailed** : Failed to process `StartTimer` decision. This happens when the decision is not configured properly, for example a timer already exists with the specified timer ID.
- **TimerFired** : A timer, previously started for this workflow execution, fired.
- **TimerCanceled** : A timer, previously started for this workflow execution, was successfully canceled.
- **CancelTimerFailed** : Failed to process `CancelTimer` decision. This happens when the decision is not configured properly, for example no timer exists with the specified timer ID.
- **StartChildWorkflowExecutionInitiated** : A request was made to start a child workflow execution.
- **StartChildWorkflowExecutionFailed** : Failed to process `StartChildWorkflowExecution` decision. This happens when the decision is not configured properly, for example the workflow type specified is not registered.
- **ChildWorkflowExecutionStarted** : A child workflow execution was successfully started.
- **ChildWorkflowExecutionCompleted** : A child workflow execution, started by this workflow execution, completed suc-

cessfully and was closed.

- **ChildWorkflowExecutionFailed** : A child workflow execution, started by this workflow execution, failed to complete successfully and was closed.
- **ChildWorkflowExecutionTimedOut** : A child workflow execution, started by this workflow execution, timed out and was closed.
- **ChildWorkflowExecutionCanceled** : A child workflow execution, started by this workflow execution, was canceled and closed.
- **ChildWorkflowExecutionTerminated** : A child workflow execution, started by this workflow execution, was terminated.
- **SignalExternalWorkflowExecutionInitiated** : A request to signal an external workflow was made.
- **ExternalWorkflowExecutionSignaled** : A signal, requested by this workflow execution, was successfully delivered to the target external workflow execution.
- **SignalExternalWorkflowExecutionFailed** : The request to signal an external workflow execution failed.
- **RequestCancelExternalWorkflowExecutionInitiated** : A request was made to request the cancellation of an external workflow execution.
- **ExternalWorkflowExecutionCancelRequested** : Request to cancel an external workflow execution was successfully delivered to the target execution.
- **RequestCancelExternalWorkflowExecutionFailed** : Request to cancel an external workflow execution failed.
- **LambdaFunctionScheduled** : An AWS Lambda function was scheduled for execution.
- **LambdaFunctionStarted** : The scheduled function was invoked in the AWS Lambda service.
- **LambdaFunctionCompleted** : The AWS Lambda function successfully completed.
- **LambdaFunctionFailed** : The AWS Lambda function execution failed.
- **LambdaFunctionTimedOut** : The AWS Lambda function execution timed out.
- **ScheduleLambdaFunctionFailed** : Failed to process `ScheduleLambdaFunction` decision. This happens when the workflow execution does not have the proper IAM role attached to invoke AWS Lambda functions.
- **StartLambdaFunctionFailed** : Failed to invoke the scheduled function in the AWS Lambda service. This happens when the AWS Lambda service is not available in the current region, or received too many requests.
- **eventTimestamp** (*datetime*) –
The date and time when the event occurred.
- **eventType** (*string*) –
The type of the history event.
- **eventId** (*integer*) –
The system generated ID of the event. This ID uniquely iden-

tifies the event with in the workflow execution history.

- **workflowExecutionStartedEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionStarted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **input** (*string*) –

The input provided to the workflow execution (if any).

- **executionStartToCloseTimeout** (*string*) –

The maximum duration for this workflow execution.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **taskStartToCloseTimeout** (*string*) –

The maximum duration of decision tasks for this workflow type.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **childPolicy** (*string*) –

The policy to use for the child workflow executions if this workflow execution is terminated, by calling the `TerminateWorkflowExecution` action explicitly or due to an expired timeout.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.
- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON:** no action will be taken. The child executions will continue to run.
- **taskList** (*dict*) –

The name of the task list for scheduling the decision tasks for this workflow execution.

- **name** (*string*) –

The name of the task list.

- **workflowType** (*dict*) –

The workflow type of this execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique within a domain.

- **tagList** (*list*) –

The list of tags associated with this workflow execution. An execution can have up to 5 tags.

- (*string*) –

- **taskPriority** (*string*) –

- **continuedExecutionRunId** (*string*) –

If this workflow execution was started due to a `ContinueAsNewWorkflowExecution` decision, then it contains the `runId` of the previous workflow execution that was closed and continued as this execution.

- **parentWorkflowExecution** (*dict*) –

The source workflow execution that started this workflow execution. The member is not set if the workflow execution was not started by a workflow.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **parentInitiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this workflow execution. The source event with this ID can be found in the history of the source workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **lambdaRole** (*string*) –

The IAM role attached to this workflow execution to use when invoking AWS Lambda functions.

- **workflowExecutionCompletedEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionCompleted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **result** (*string*) –

The result produced by the workflow execution upon successful completion.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CompleteWorkflowExecution` decision to complete this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **completeWorkflowExecutionFailedEventAttributes** (*dict*)

–

If the event is of type `CompleteWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CompleteWorkflowExecution` decision to complete this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **workflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **reason** (*string*) –

The descriptive reason provided for the failure (if any).

- **details** (*string*) –

The details of the failure (if any).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `FailWorkflowExecution` decision to fail this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **failWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `FailWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `FailWorkflowExecution` decision to fail this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **workflowExecutionTimedOutEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionTimedOut` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timeoutType** (*string*) –

The type of timeout that caused this event.

- **childPolicy** (*string*) –

The policy used for the child workflow executions of this workflow execution.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.
- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON:** no action will be taken. The child executions will continue to run.
- **workflowExecutionCanceledEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionCanceled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **details** (*string*) –

Details for the cancellation (if any).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CancelWorkflowExecution` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **cancelWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `CancelWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions.

For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

· **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CancelWorkflowExecution` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

· **workflowExecutionContinuedAsNewEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionContinuedAsNew` then this member is set and provides detailed information about the event. It is not set for other event types.

· **input** (*string*) –

The input provided to the new workflow execution.

· **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `ContinueAsNewWorkflowExecution` decision that started this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

· **newExecutionRunId** (*string*) –

The `runId` of the new workflow execution.

· **executionStartToCloseTimeout** (*string*) –

The total duration allowed for the new workflow execution.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

· **taskList** (*dict*) –

Represents a task list.

· **name** (*string*) –

The name of the task list.

· **taskPriority** (*string*) –

· **taskStartToCloseTimeout** (*string*) –

The maximum duration of decision tasks for the new workflow execution.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

· **childPolicy** (*string*) –

The policy to use for the child workflow executions of the new execution if it is terminated by calling the `TerminateWorkflowExecution` action explicitly or due to an expired timeout.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.
- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON:** no action will be taken. The child executions will continue to run.
- **tagList** (*list*) –

The list of tags associated with the new workflow execution.

- (*string*) –
- **workflowType** (*dict*) –

Represents a workflow type.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **lambdaRole** (*string*) –

The IAM role attached to this workflow execution to use when invoking AWS Lambda functions.

- **continueAsNewWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `ContinueAsNewWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `ContinueAsNewWorkflowExecution` decision that started this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **workflowExecutionTerminatedEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionTerminated` then this member is set and provides detailed information about the event. It is not set for other event types.

- **reason** (*string*) –

The reason provided for the termination (if any).

- **details** (*string*) –

The details provided for the termination (if any).

- **childPolicy** (*string*) –

The policy used for the child workflow executions of this workflow execution.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.
- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON:** no action will be taken. The child executions will continue to run.
- **cause** (*string*) –

If set, indicates that the workflow execution was automatically terminated, and specifies the cause. This happens if the parent workflow execution times out or is terminated and the child policy is set to terminate child executions.

- **workflowExecutionCancelRequestedEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionCancelRequested` then this member is set and provides detailed information about the event. It is not set for other event types.

- **externalWorkflowExecution** (*dict*) –

The external workflow execution for which the cancellation was requested.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **externalInitiatedEventId** (*integer*) –

The ID of the `RequestCancelExternalWorkflowExecutionInitiated` event corresponding to the `RequestCancelExternalWorkflowExecution` decision to cancel this workflow execution. The source event with this ID can be found in the history of the source workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **cause** (*string*) –

If set, indicates that the request to cancel the workflow execution was automatically generated, and specifies the cause. This happens if the parent workflow execution times out or is terminated, and the child policy is set to cancel child executions.

- **decisionTaskScheduledEventAttributes** (*dict*) –

If the event is of type `DecisionTaskScheduled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **taskList** (*dict*) –

The name of the task list in which the decision task was scheduled.

- **name** (*string*) –

The name of the task list.

- **taskPriority** (*string*) –

Optional. A task priority that, if set, specifies the priority for this decision task. Valid values are integers that range from Java's `Integer.MIN_VALUE` (-2147483648) to `Integer.MAX_VALUE` (2147483647). Higher numbers indicate higher priority.

For more information about setting task priority, see [Setting Task Priority](#) in the *Amazon Simple Workflow Developer Guide*.

- **startToCloseTimeout** (*string*) –

The maximum duration for this decision task. The task is considered timed out if it does not completed within this duration.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **decisionTaskStartedEventAttributes** (*dict*) –

If the event is of type `DecisionTaskStarted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **identity** (*string*) –

Identity of the decider making the request. This enables diagnostic tracing when problems arise. The form of this identity is user defined.

- **scheduledEventId** (*integer*) –

The ID of the `DecisionTaskScheduled` event that was recorded when this decision task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **decisionTaskCompletedEventAttributes** (*dict*) –

If the event is of type `DecisionTaskCompleted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **executionContext** (*string*) –

User defined context for the workflow execution.

- **scheduledEventId** (*integer*) –

The ID of the `DecisionTaskScheduled` event that was recorded when this decision task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `DecisionTaskStarted` event recorded when this decision task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **decisionTaskTimedOutEventAttributes** (*dict*) –

If the event is of type `DecisionTaskTimedOut` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timeoutType** (*string*) –

The type of timeout that expired before the decision task could be completed.

- **scheduledEventId** (*integer*) –

The ID of the `DecisionTaskScheduled` event that was recorded when this decision task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `DecisionTaskStarted` event recorded when this decision task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **activityTaskScheduledEventAttributes** (*dict*) –

If the event is of type `ActivityTaskScheduled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **activityType** (*dict*) –

The type of the activity task.

- **name** (*string*) –

The name of this activity.

Note: The combination of activity type name and version must be unique within a domain.

- **version** (*string*) –

The version of this activity.

Note: The combination of activity type name and version must be unique with in a domain.

- **activityId** (*string*) –

The unique ID of the activity task.

- **input** (*string*) –
The input provided to the activity task.
- **control** (*string*) –
Optional. Data attached to the event that can be used by the decider in subsequent workflow tasks. This data is not sent to the activity.
- **scheduleToStartTimeout** (*string*) –
The maximum amount of time the activity task can wait to be assigned to a worker.
- **scheduleToCloseTimeout** (*string*) –
The maximum amount of time for this activity task.
- **startToCloseTimeout** (*string*) –
The maximum amount of time a worker may take to process the activity task.
- **taskList** (*dict*) –
The task list in which the activity task has been scheduled.
- **name** (*string*) –
The name of the task list.
- **taskPriority** (*string*) –
Optional. The priority to assign to the scheduled activity task. If set, this will override any default priority value that was assigned when the activity type was registered.

Valid values are integers that range from Java's `Integer.MIN_VALUE` (-2147483648) to `Integer.MAX_VALUE` (2147483647). Higher numbers indicate higher priority.

For more information about setting task priority, see [Setting Task Priority](#) in the *Amazon Simple Workflow Developer Guide*.
- **decisionTaskCompletedEventId** (*integer*) –
The ID of the `DecisionTaskCompleted` event corresponding to the decision that resulted in the scheduling of this activity task. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.
- **heartbeatTimeout** (*string*) –
The maximum time before which the worker processing this task must report progress by calling `RecordActivityTaskHeartbeat`. If the timeout is exceeded, the activity task is automatically timed out. If the worker subsequently attempts to record a heartbeat or return a result, it will be ignored.
- **activityTaskStartedEventAttributes** (*dict*) –
If the event is of type `ActivityTaskStarted` then this member is set and provides detailed information about the event. It is not set for other event types.
- **identity** (*string*) –

Identity of the worker that was assigned this task. This aids diagnostics when problems arise. The form of this identity is user defined.

- **scheduledEventId** (*integer*) –

The ID of the `ActivityTaskScheduled` event that was recorded when this activity task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **activityTaskCompletedEventAttributes** (*dict*) –

If the event is of type `ActivityTaskCompleted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **result** (*string*) –

The results of the activity task (if any).

- **scheduledEventId** (*integer*) –

The ID of the `ActivityTaskScheduled` event that was recorded when this activity task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ActivityTaskStarted` event recorded when this activity task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **activityTaskFailedEventAttributes** (*dict*) –

If the event is of type `ActivityTaskFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **reason** (*string*) –

The reason provided for the failure (if any).

- **details** (*string*) –

The details of the failure (if any).

- **scheduledEventId** (*integer*) –

The ID of the `ActivityTaskScheduled` event that was recorded when this activity task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ActivityTaskStarted` event recorded when this activity task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **activityTaskTimedOutEventAttributes** (*dict*) –

If the event is of type `ActivityTaskTimedOut` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timeoutType** (*string*) –

The type of the timeout that caused this event.

- **scheduledEventId** (*integer*) –

The ID of the `ActivityTaskScheduled` event that was recorded when this activity task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ActivityTaskStarted` event recorded when this activity task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **details** (*string*) –

Contains the content of the `details` parameter for the last call made by the activity to `RecordActivityTaskHeartbeat`.

- **activityTaskCanceledEventAttributes** (*dict*) –

If the event is of type `ActivityTaskCanceled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **details** (*string*) –

Details of the cancellation (if any).

- **scheduledEventId** (*integer*) –

The ID of the `ActivityTaskScheduled` event that was recorded when this activity task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ActivityTaskStarted` event recorded when this activity task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **latestCancelRequestedEventId** (*integer*) –

If set, contains the ID of the last `ActivityTaskCancelRequested` event recorded for this activity task. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **activityTaskCancelRequestedEventAttributes** (*dict*) –

If the event is of type `ActivityTaskCancelRequested` then this member is set and provides detailed information about the event. It is not set for other event types.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RequestCancelActivityTask` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **activityId** (*string*) –

The unique ID of the task.

- **workflowExecutionSignaledEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionSignaled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **signalName** (*string*) –

The name of the signal received. The decider can use the signal name and inputs to determine how to process the signal.

- **input** (*string*) –

Inputs provided with the signal (if any). The decider can use the signal name and inputs to determine how to process the signal.

- **externalWorkflowExecution** (*dict*) –

The workflow execution that sent the signal. This is set only if the signal was sent by another workflow execution.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **externalInitiatedEventId** (*integer*) –

The ID of the `SignalExternalWorkflowExecutionInitiated` event corresponding to the `SignalExternalWorkflow` decision to signal this workflow execution. The source event with this ID can be found in the history of the source workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event. This field is set only if the signal was initiated by another workflow execution.

- **markerRecordedEventAttributes** (*dict*) –

If the event is of type `MarkerRecorded` then this member is set and provides detailed information about the event. It is not set for other event types.

- **markerName** (*string*) –

The name of the marker.

- **details** (*string*) –

Details of the marker (if any).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RecordMarker` decision that requested this marker. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **recordMarkerFailedEventAttributes** (*dict*) –

If the event is of type `DecisionTaskFailed` then this member is set and provides detailed information about the

event. It is not set for other event types.

- **markerName** (*string*) –

The marker's name.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RecordMarkerFailed` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **timerStartedEventAttributes** (*dict*) –

If the event is of type `TimerStarted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timerId** (*string*) –

The unique ID of the timer that was started.

- **control** (*string*) –

Optional. Data attached to the event that can be used by the decider in subsequent workflow tasks.

- **startToFireTimeout** (*string*) –

The duration of time after which the timer will fire.

The duration is specified in seconds; an integer greater than or equal to 0.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `StartTimer` decision for this activity task. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **timerFiredEventAttributes** (*dict*) –

If the event is of type `TimerFired` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timerId** (*string*) –

The unique ID of the timer that fired.

- **startedEventId** (*integer*) –

The ID of the `TimerStarted` event that was recorded when this timer was started. This information can be useful for diagnosing problems by tracing back the chain of events leading

up to this event.

- **timerCanceledEventAttributes** (*dict*) –

If the event is of type `TimerCanceled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timerId** (*string*) –

The unique ID of the timer that was canceled.

- **startedEventId** (*integer*) –

The ID of the `TimerStarted` event that was recorded when this timer was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CancelTimer` decision to cancel this timer. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startChildWorkflowExecutionInitiatedEventAttributes** (*dict*) –

If the event is of type `StartChildWorkflowExecutionInitiated` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowId** (*string*) –

The `workflowId` of the child workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **control** (*string*) –

Optional. Data attached to the event that can be used by the decider in subsequent decision tasks. This data is not sent to the activity.

- **input** (*string*) –

The inputs provided to the child workflow execution (if any).

- **executionStartToCloseTimeout** (*string*) –

The maximum duration for the child workflow execution. If the workflow execution is not closed within this duration, it

will be timed out and force terminated.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **taskList** (*dict*) –

The name of the task list used for the decision tasks of the child workflow execution.

- **name** (*string*) –

The name of the task list.

- **taskPriority** (*string*) –

Optional. The priority assigned for the decision tasks for this workflow execution. Valid values are integers that range from Java’s `Integer.MIN_VALUE` (-2147483648) to `Integer.MAX_VALUE` (2147483647). Higher numbers indicate higher priority.

For more information about setting task priority, see [Setting Task Priority](#) in the *Amazon Simple Workflow Developer Guide*.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `StartChildWorkflowExecution` decision to request this child workflow execution. This information can be useful for diagnosing problems by tracing back the cause of events.

- **childPolicy** (*string*) –

The policy to use for the child workflow executions if this execution gets terminated by explicitly calling the `TerminateWorkflowExecution` action or due to an expired timeout.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.
- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON:** no action will be taken. The child executions will continue to run.
- **taskStartToCloseTimeout** (*string*) –

The maximum duration allowed for the decision tasks for this workflow execution.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **tagList** (*list*) –

The list of tags to associated with the child workflow execution.

- (*string*) –

- **lambdaRole** (*string*) –

The IAM role attached to this workflow execution to use when invoking AWS Lambda functions.

- **childWorkflowExecutionStartedEventAttributes** (*dict*) –

If the event is of type `ChildWorkflowExecutionStarted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that was started.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **childWorkflowExecutionCompletedEventAttributes** (*dict*) –

If the event is of type `ChildWorkflowExecutionCompleted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that was completed.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **result** (*string*) –

The result of the child workflow execution (if any).

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ChildWorkflowExecutionStarted` event recorded when this child workflow execution was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **childWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `ChildWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that failed.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **reason** (*string*) –

The reason for the failure (if provided).

- **details** (*string*) –

The details of the failure (if provided).

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ChildWorkflowExecutionStarted` event recorded when this child workflow execution was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **childWorkflowExecutionTimedOutEventAttributes** (*dict*) –

If the event is of type `ChildWorkflowExecutionTimedOut` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that timed out.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **timeoutType** (*string*) –

The type of the timeout that caused the child workflow execution to time out.

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ChildWorkflowExecutionStarted` event recorded when this child workflow execution was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **childWorkflowExecutionCanceledEventAttributes** (*dict*) –

If the event is of type `ChildWorkflowExecutionCanceled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that was canceled.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **details** (*string*) –

Details of the cancellation (if provided).

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ChildWorkflowExecutionStarted` event recorded when this child workflow execution was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **childWorkflowExecutionTerminatedEventAttributes** (*dict*) –

If the event is of type `ChildWorkflowExecutionTerminated` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that was terminated.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ChildWorkflowExecutionStarted` event recorded when this child workflow execution was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **signalExternalWorkflowExecutionInitiatedEventAttributes** (*dict*) –

If the event is of type `SignalExternalWorkflowExecutionInitiated` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowId** (*string*) –

The `workflowId` of the external workflow execution.

- **runId** (*string*) –

The `runId` of the external workflow execution to send the signal to.

- **signalName** (*string*) –

The name of the signal.

- **input** (*string*) –

Input provided to the signal (if any).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `SignalExternalWorkflowExecution` decision for this signal. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **control** (*string*) –

Optional. data attached to the event that can be used by the decider in subsequent decision tasks.

- **externalWorkflowExecutionSignaledEventAttributes** (*dict*) –

If the event is of type `ExternalWorkflowExecutionSignaled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The external workflow execution that the signal was delivered to.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **initiatedEventId** (*integer*) –

The ID of the `SignalExternalWorkflowExecutionInitiated` event corresponding to the `SignalExternalWorkflowExecution` decision to request this signal. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **signalExternalWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `SignalExternalWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowId** (*string*) –

The `workflowId` of the external workflow execution that the signal was being delivered to.

- **runId** (*string*) –

The `runId` of the external workflow execution that the signal was being delivered to.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **initiatedEventId** (*integer*) –

The ID of the `SignalExternalWorkflowExecutionInitiated` event corresponding to the `SignalExternalWorkflowExecution` decision to request this signal. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `SignalExternalWorkflowExecution` decision for this signal. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **control** (*string*) –

- **externalWorkflowExecutionCancelRequestedEventAttributes** (*dict*) –

If the event is of type `ExternalWorkflowExecutionCancelRequested` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The external workflow execution to which the cancellation

request was delivered.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **initiatedEventId** (*integer*) –

The ID of the `RequestCancelExternalWorkflowExecutionInitiated` event corresponding to the `RequestCancelExternalWorkflowExecution` decision to cancel this external workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **requestCancelExternalWorkflowExecutionInitiatedEventAttributes** (*dict*) –

If the event is of type `RequestCancelExternalWorkflowExecutionInitiated` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowId** (*string*) –

The `workflowId` of the external workflow execution to be canceled.

- **runId** (*string*) –

The `runId` of the external workflow execution to be canceled.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RequestCancelExternalWorkflowExecution` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **control** (*string*) –

Optional. Data attached to the event that can be used by the decider in subsequent workflow tasks.

- **requestCancelExternalWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `RequestCancelExternalWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowId** (*string*) –

The `workflowId` of the external workflow to which the cancel request was to be delivered.

- **runId** (*string*) –

The `runId` of the external workflow execution.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **initiatedEventId** (*integer*) –

The ID of the `RequestCancelExternalWorkflowExecutionInitiated` event corresponding to the `RequestCancelExternalWorkflowExecution` decision to cancel this external workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RequestCancelExternalWorkflowExecution` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **control** (*string*) –

- **scheduleActivityTaskFailedEventAttributes** (*dict*) –

If the event is of type `ScheduleActivityTaskFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **activityType** (*dict*) –

The activity type provided in the `ScheduleActivityTask` decision that failed.

- **name** (*string*) –

The name of this activity.

Note: The combination of activity type name and version must be unique within a domain.

- **version** (*string*) –

The version of this activity.

Note: The combination of activity type name and version must be unique with in a domain.

- **activityId** (*string*) –

The activityId provided in the `ScheduleActivityTask` decision that failed.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions.

For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

· **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision that resulted in the scheduling of this activity task. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

· **requestCancelActivityTaskFailedEventAttributes** (*dict*) –

If the event is of type `RequestCancelActivityTaskFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

· **activityId** (*string*) –

The `activityId` provided in the `RequestCancelActivityTask` decision that failed.

· **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

· **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RequestCancelActivityTask` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

· **startTimerFailedEventAttributes** (*dict*) –

If the event is of type `StartTimerFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

· **timerId** (*string*) –

The `timerId` provided in the `StartTimer` decision that failed.

· **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

· **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `StartTimer` decision for this activity task. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **cancelTimerFailedEventAttributes** (*dict*) –

If the event is of type `CancelTimerFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timerId** (*string*) –

The `timerId` provided in the `CancelTimer` decision that failed.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CancelTimer` decision to cancel this timer. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startChildWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `StartChildWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowType** (*dict*) –

The workflow type provided in the `StartChildWorkflowExecution` decision that failed.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique within a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique within a domain.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **workflowId** (*string*) –

The workflowId of the child workflow execution.

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `StartChildWorkflowExecution` decision to request this child workflow execution. This information can be useful for diagnosing problems by tracing back the cause of events.

- **control** (*string*) –

- **lambdaFunctionScheduledEventAttributes** (*dict*) –

Provides details for the `LambdaFunctionScheduled` event.

- **id** (*string*) –

The unique Amazon SWF ID for the AWS Lambda task.

- **name** (*string*) –

The name of the scheduled AWS Lambda function.

- **input** (*string*) –

Input provided to the AWS Lambda function.

- **startToCloseTimeout** (*string*) –

The maximum time, in seconds, that the AWS Lambda function can take to execute from start to close before it is marked as failed.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event for the decision that resulted in the scheduling of this AWS Lambda function. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **lambdaFunctionStartedEventAttributes** (*dict*) –

Provides details for the `LambdaFunctionStarted` event.

- **scheduledEventId** (*integer*) –

The ID of the `LambdaFunctionScheduled` event that was recorded when this AWS Lambda function was sched-

uled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **lambdaFunctionCompletedEventAttributes** (*dict*) –

Provides details for the `LambdaFunctionCompleted` event.

- **scheduledEventId** (*integer*) –

The ID of the `LambdaFunctionScheduled` event that was recorded when this AWS Lambda function was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `LambdaFunctionStarted` event recorded in the history.

- **result** (*string*) –

The result of the function execution (if any).

- **lambdaFunctionFailedEventAttributes** (*dict*) –

Provides details for the `LambdaFunctionFailed` event.

- **scheduledEventId** (*integer*) –

The ID of the `LambdaFunctionScheduled` event that was recorded when this AWS Lambda function was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `LambdaFunctionStarted` event recorded in the history.

- **reason** (*string*) –

The reason provided for the failure (if any).

- **details** (*string*) –

The details of the failure (if any).

- **lambdaFunctionTimedOutEventAttributes** (*dict*) –

Provides details for the `LambdaFunctionTimedOut` event.

- **scheduledEventId** (*integer*) –

The ID of the `LambdaFunctionScheduled` event that was recorded when this AWS Lambda function was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `LambdaFunctionStarted` event recorded in the history.

- **timeoutType** (*string*) –

The type of the timeout that caused this event.

- **scheduleLambdaFunctionFailedEventAttributes** (*dict*) –

Provides details for the `ScheduleLambdaFunctionFailed` event.

- **id** (*string*) –

The unique Amazon SWF ID of the AWS Lambda task.

- **name** (*string*) –

The name of the scheduled AWS Lambda function.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision that resulted in the scheduling of this AWS Lambda function. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startLambdaFunctionFailedEventAttributes** (*dict*) –

Provides details for the `StartLambdaFunctionFailed` event.

- **scheduledEventId** (*integer*) –

The ID of the `LambdaFunctionScheduled` event that was recorded when this AWS Lambda function was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **message** (*string*) –

The error message (if any).

- **nextPageToken** (*string*) –

If a `NextPageToken` was returned by a previous call, there are more results available. To retrieve the next page of results, make the call again using the returned token in `nextPageToken`. Keep all other arguments unchanged.

The configured `maximumPageSize` determines how many results can be returned in a single call.

- **previousStartedEventId** (*integer*) –

The ID of the `DecisionTaskStarted` event of the previous decision task of this workflow execution that was processed by the decider. This can be used to determine the events in the history new since the last decision task received by the decider.

record_activity_task_heartbeat (***kwargs*)

Used by activity workers to report to the service that the `ActivityTask` represented by the specified `taskToken` is still making progress. The worker can also (optionally) specify details of the progress, for example percent complete, using the `details` parameter. This action can also be used by the worker as a mechanism to check if cancellation is being requested for the activity task. If a cancellation is being attempted for the specified task, then the boolean `cancelRequested` flag returned by the service is set to `true`.

This action resets the `taskHeartbeatTimeout` clock. The `taskHeartbeatTimeout` is specified in `RegisterActivityType`.

This action does not in itself create an event in the workflow execution history. However, if the task times out, the workflow execution history will contain a `ActivityTaskTimedOut` event that contains the information from the last heartbeat generated by the activity worker.

Note: The `taskStartToCloseTimeout` of an activity type is the maximum duration of an activity task, regardless of the number of `RecordActivityTaskHeartbeat` requests received. The `taskStartToCloseTimeout` is also specified in `RegisterActivityType`.

Note: This operation is only useful for long-lived activities to report liveliness of the task and to determine if a cancellation is being attempted.

Warning: If the `cancelRequested` flag returns `true`, a cancellation is being attempted. If the worker can cancel the activity, it should respond with `RespondActivityTaskCanceled`. Otherwise, it should ignore the cancellation request.

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- You cannot use an IAM policy to constrain this action's parameters.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.record_activity_task_heartbeat(
    taskToken='string',
    details='string'
)
```

Parameters

- **taskToken** (*string*) – [REQUIRED]

The `taskToken` of the `ActivityTask`.

Warning: `taskToken` is generated by the service and should be treated as an opaque value. If the task is passed to another process, its `taskToken` must also be passed. This enables it to provide its progress and respond with results.

- **details** (*string*) – If specified, contains details about the progress of the task.

Return type dict

Returns

Response Syntax

```
{
    'cancelRequested': True|False
}
```

Response Structure

- (*dict*) –

Status information about an activity task.

- **cancelRequested** (*boolean*) –

Set to `true` if cancellation of the task is requested.

`register_activity_type` (***kwargs*)

Registers a new *activity type* along with its configuration settings in the specified domain.

Warning: A `TypeAlreadyExists` fault is returned if the type already exists in the domain. You cannot change any configuration settings of the type after its registration, and it must be registered as a new version.

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- Constrain the following parameters by using a `Condition` element with the appropriate keys. *
 `defaultTaskList.name`: String constraint. The key is `swf:defaultTaskList.name`.
- `name`: String constraint. The key is `swf:name`.
- `version`: String constraint. The key is `swf:version`.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.register_activity_type(
    domain='string',
    name='string',
    version='string',
    description='string',
    defaultTaskStartToCloseTimeout='string',
    defaultTaskHeartbeatTimeout='string',
    defaultTaskList={
        'name': 'string'
    },
    defaultTaskPriority='string',
    defaultTaskScheduleToStartTimeout='string',
    defaultTaskScheduleToCloseTimeout='string'
)
```

Parameters

- **domain** (*string*) – [REQUIRED]

The name of the domain in which this activity is to be registered.

- **name** (*string*) – [REQUIRED]

The name of the activity type within the domain.

The specified string must not start or end with whitespace. It must not contain a : (colon), / (slash), | (vertical bar), or any control characters (u0000-u001f | u007f - u009f). Also, it must not contain the literal string quotarnquot.

- **version** (*string*) – [REQUIRED]

The version of the activity type.

Note: The activity type consists of the name and version, the combination of which must be unique within the domain.

The specified string must not start or end with whitespace. It must not contain a : (colon), / (slash), | (vertical bar), or any control characters (u0000-u001f | u007f - u009f). Also, it must not contain the literal string quotarnquot.

- **description** (*string*) – A textual description of the activity type.
- **defaultTaskStartToCloseTimeout** (*string*) – If set, specifies the default maximum duration that a worker can take to process tasks of this activity type. This default can be overridden when scheduling an activity task using the `ScheduleActivityTask` decision.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **defaultTaskHeartbeatTimeout** (*string*) – If set, specifies the default maximum time before which a worker processing a task of this type must report progress by calling `RecordActivityTaskHeartbeat`. If the timeout is exceeded, the activity task is automatically timed out. This default can be overridden when scheduling an activity task using the `ScheduleActivityTask` decision. If the activity worker subsequently attempts to record a heartbeat or returns a result, the activity worker receives an `UnknownResource` fault. In this case, Amazon SWF no longer considers the activity task to be valid; the activity worker should clean up the activity task.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **defaultTaskList** (*dict*) – If set, specifies the default task list to use for scheduling tasks of this activity type. This default task list is used if a task list is not provided when a task is scheduled through the `ScheduleActivityTask` decision.
 - **name** (*string*) – [REQUIRED]

The name of the task list.

- **defaultTaskPriority** (*string*) – The default task priority to assign to the activity type. If not assigned, then “0” will be used. Valid values are integers that range from Java’s `Integer.MIN_VALUE` (-2147483648) to `Integer.MAX_VALUE` (2147483647). Higher numbers indicate higher priority.

For more information about setting task priority, see [Setting Task Priority](#) in the *Amazon Simple Workflow Developer Guide*.

- **defaultTaskScheduleToStartTimeout** (*string*) – If set, specifies the default maximum duration that a task of this activity type can wait before being assigned to a worker. This default can be overridden when scheduling an activity task using the `ScheduleActivityTask` decision.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **defaultTaskScheduleToCloseTimeout** (*string*) – If set, specifies the default maximum duration for a task of this activity type. This default can be overridden when scheduling an activity task using the `ScheduleActivityTask` decision.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

Returns None

register_domain (**kwargs)

Registers a new domain.

Access Control

You can use IAM policies to control this action’s access to Amazon SWF resources as follows:

- You cannot use an IAM policy to control domain access for this action. The name of the domain being registered is available as the resource of this action.
- Use an `Action` element to allow or deny permission to call this action.
- You cannot use an IAM policy to constrain this action’s parameters.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute’s **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.register_domain(
    name='string',
    description='string',
    workflowExecutionRetentionPeriodInDays='string'
)
```

Parameters

- **name** (*string*) – [REQUIRED]

Name of the domain to register. The name must be unique in the region that the domain is registered in.

The specified string must not start or end with whitespace. It must not contain a : (colon), / (slash), | (vertical bar), or any control characters (u0000-u001f | u007f - u009f). Also, it must not contain the literal string quotation.

- **description** (*string*) – A text description of the domain.
- **workflowExecutionRetentionPeriodInDays** (*string*) – [REQUIRED]

The duration (in days) that records and histories of workflow executions on the domain should be kept by the service. After the retention period, the workflow execution is not available in the results of visibility calls.

If you pass the value `NONE` or `0` (zero), then the workflow execution history will not be retained. As soon as the workflow execution completes, the execution record and its history are deleted.

The maximum workflow execution retention period is 90 days. For more information about Amazon SWF service limits, see: [Amazon SWF Service Limits](#) in the *Amazon SWF Developer Guide*.

Returns None

register_workflow_type (**kwargs)

Registers a new *workflow type* and its configuration settings in the specified domain.

The retention period for the workflow history is set by the RegisterDomain action.

Warning: If the type already exists, then a `TypeAlreadyExists` fault is returned. You cannot change the configuration settings of a workflow type once it is registered and it must be registered as a new version.

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- Constrain the following parameters by using a `Condition` element with the appropriate keys. *
 - `defaultTaskList.name`: String constraint. The key is `swf:defaultTaskList.name`.
 - `name`: String constraint. The key is `swf:name`.
 - `version`: String constraint. The key is `swf:version`.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.register_workflow_type(
    domain='string',
    name='string',
    version='string',
    description='string',
    defaultTaskStartToCloseTimeout='string',
    defaultExecutionStartToCloseTimeout='string',
    defaultTaskList={
        'name': 'string'
    },
    defaultTaskPriority='string',
    defaultChildPolicy='TERMINATE' | 'REQUEST_CANCEL' | 'ABANDON',
    defaultLambdaRole='string'
)
```

Parameters

- **domain** (*string*) – [REQUIRED]

The name of the domain in which to register the workflow type.

- **name** (*string*) – [REQUIRED]

The name of the workflow type.

The specified string must not start or end with whitespace. It must not contain a `:` (colon), `/` (slash), `|` (vertical bar), or any control characters (u0000-u001f | u007f - u009f). Also, it must not contain the literal string `quotarnquot`.

- **version** (*string*) – [REQUIRED]

The version of the workflow type.

Note: The workflow type consists of the name and version, the combination of which must be unique within the domain. To get a list of all currently registered workflow types, use the `ListWorkflowTypes` action.

The specified string must not start or end with whitespace. It must not contain a `:` (colon), `/` (slash), `|` (vertical bar), or any control characters (u0000-u001f |

u007f - u009f). Also, it must not contain the literal string `quotarnquot`.

- **description** (*string*) – Textual description of the workflow type.
- **defaultTaskStartToCloseTimeout** (*string*) – If set, specifies the default maximum duration of decision tasks for this workflow type. This default can be overridden when starting a workflow execution using the `StartWorkflowExecution` action or the `StartChildWorkflowExecution` decision.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **defaultExecutionStartToCloseTimeout** (*string*) – If set, specifies the default maximum duration for executions of this workflow type. You can override this default when starting an execution through the `StartWorkflowExecution` action or `StartChildWorkflowExecution` decision.

The duration is specified in seconds; an integer greater than or equal to 0. Unlike some of the other timeout parameters in Amazon SWF, you cannot specify a value of “NONE” for `defaultExecutionStartToCloseTimeout`; there is a one-year max limit on the time that a workflow execution can run. Exceeding this limit will always cause the workflow execution to time out.

- **defaultTaskList** (*dict*) – If set, specifies the default task list to use for scheduling decision tasks for executions of this workflow type. This default is used only if a task list is not provided when starting the execution through the `StartWorkflowExecution` action or `StartChildWorkflowExecution` decision.
 - **name** (*string*) – [REQUIRED]

The name of the task list.

- **defaultTaskPriority** (*string*) – The default task priority to assign to the workflow type. If not assigned, then “0” will be used. Valid values are integers that range from Java’s `Integer.MIN_VALUE` (-2147483648) to `Integer.MAX_VALUE` (2147483647). Higher numbers indicate higher priority.

For more information about setting task priority, see [Setting Task Priority](#) in the *Amazon Simple Workflow Developer Guide*.

- **defaultChildPolicy** (*string*) – If set, specifies the default policy to use for the child workflow executions when a workflow execution of this type is terminated, by calling the `TerminateWorkflowExecution` action explicitly or due to an expired timeout. This default can be overridden when starting a workflow execution using the `StartWorkflowExecution` action or the `StartChildWorkflowExecution` decision.

The supported child policies are:

- **TERMINATE**: the child executions will be terminated.
- **REQUEST_CANCEL**: a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON**: no action will be taken. The child executions will continue to run.
- **defaultLambdaRole** (*string*) – The ARN of the default IAM role to use when a workflow execution of this type invokes AWS Lambda functions.

This default can be overridden when starting a workflow execution using the `StartWorkflowExecution` action or the `StartChildWorkflowExecution` and `ContinueAsNewWorkflowExecution` decision.

Returns None

request_cancel_workflow_execution (**kwargs)

Records a `WorkflowExecutionCancelRequested` event in the currently running workflow execution identified by the given domain, workflowId, and runId. This logically requests the cancellation of the workflow execution as a whole. It is up to the decider to take appropriate actions when it receives an execution history with this event.

Note: If the runId is not specified, the `WorkflowExecutionCancelRequested` event is recorded in the history of the current open workflow execution with the specified workflowId in the domain.

Note: Because this action allows the workflow to properly clean up and gracefully close, it should be used instead of `TerminateWorkflowExecution` when possible.

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- You cannot use an IAM policy to constrain this action's parameters.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.request_cancel_workflow_execution(
    domain='string',
    workflowId='string',
    runId='string'
)
```

Parameters

- **domain** (*string*) – [REQUIRED]
The name of the domain containing the workflow execution to cancel.
- **workflowId** (*string*) – [REQUIRED]
The workflowId of the workflow execution to cancel.
- **runId** (*string*) – The runId of the workflow execution to cancel.

Returns None

respond_activity_task_canceled (**kwargs)

Used by workers to tell the service that the `ActivityTask` identified by the `taskToken` was successfully canceled. Additional details can be optionally provided using the `details` argument.

These details (if provided) appear in the `ActivityTaskCanceled` event added to the workflow history.

Warning: Only use this operation if the `canceled` flag of a `RecordActivityTaskHeartbeat` request returns `true` and if the activity can be safely undone or abandoned.

A task is considered open from the time that it is scheduled until it is closed. Therefore a task is reported as open while a worker is processing it. A task is closed after it has been specified in a call to `RespondActivityTaskCompleted`, `RespondActivityTaskCanceled`, `RespondActivityTaskFailed`, or the task has [timed out](#).

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- You cannot use an IAM policy to constrain this action's parameters.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.respond_activity_task_canceled(
    taskToken='string',
    details='string'
)
```

Parameters

- **taskToken** (*string*) – [REQUIRED]

The taskToken of the ActivityTask.

Warning: taskToken is generated by the service and should be treated as an opaque value. If the task is passed to another process, its taskToken must also be passed. This enables it to provide its progress and respond with results.

- **details** (*string*) – *Optional.* Information about the cancellation.

Returns None

respond_activity_task_completed (**kwargs)

Used by workers to tell the service that the ActivityTask identified by the taskToken completed successfully with a result (if provided). The result appears in the ActivityTaskCompleted event in the workflow history.

Warning: If the requested task does not complete successfully, use `RespondActivityTaskFailed` instead. If the worker finds that the task is canceled through the `canceled` flag returned by `RecordActivityTaskHeartbeat`, it should cancel the task, clean up and then call `RespondActivityTaskCanceled`.

A task is considered open from the time that it is scheduled until it is closed. Therefore a task is reported as open while a worker is processing it. A task is closed after it has been specified in a call to `RespondActivityTaskCompleted`, `RespondActivityTaskCanceled`, `RespondActivityTaskFailed`, or the task has [timed out](#).

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- You cannot use an IAM policy to constrain this action's parameters.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax


```
response = client.respond_activity_task_completed(
    taskToken='string',
    result='string'
)
```

Parameters

- **taskToken** (*string*) – [REQUIRED]

The taskToken of the ActivityTask .

Warning: taskToken is generated by the service and should be treated as an opaque value. If the task is passed to another process, its taskToken must also be passed. This enables it to provide its progress and respond with results.

- **result** (*string*) – The result of the activity task. It is a free form string that is implementation specific.

Returns None

respond_activity_task_failed (**kwargs)

Used by workers to tell the service that the ActivityTask identified by the taskToken has failed with reason (if specified). The reason and details appear in the ActivityTaskFailed event added to the workflow history.

A task is considered open from the time that it is scheduled until it is closed. Therefore a task is reported as open while a worker is processing it. A task is closed after it has been specified in a call to RespondActivityTaskCompleted , RespondActivityTaskCanceled , RespondActivityTaskFailed, or the task has [timed out](#) .

Access Control

You can use IAM policies to control this action’s access to Amazon SWF resources as follows:

- Use a **Resource** element with the domain name to limit the action to only specified domains.
- Use an **Action** element to allow or deny permission to call this action.
- You cannot use an IAM policy to constrain this action’s parameters.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute’s **cause** parameter will be set to **OPERATION_NOT_PERMITTED**. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#) .

Request Syntax

```
response = client.respond_activity_task_failed(
    taskToken='string',
    reason='string',
    details='string'
)
```

Parameters

- **taskToken** (*string*) – [REQUIRED]

The taskToken of the ActivityTask .

Warning: taskToken is generated by the service and should be treated as an opaque value. If the task is passed to another process, its taskToken must also be passed. This enables it to provide its progress and respond with results.

- **reason** (*string*) – Description of the error that may assist in diagnostics.
- **details** (*string*) – *Optional*. Detailed information about the failure.

Returns None

respond_decision_task_completed (**kwargs)

Used by deciders to tell the service that the DecisionTask identified by the `taskToken` has successfully completed. The `decisions` argument specifies the list of decisions made while processing the task.

A `DecisionTaskCompleted` event is added to the workflow history. The `executionContext` specified is attached to the event in the workflow execution history.

Access Control

If an IAM policy grants permission to use `RespondDecisionTaskCompleted`, it can express permissions for the list of decisions in the `decisions` parameter. Each of the decisions has one or more parameters, much like a regular API call. To allow for policies to be as readable as possible, you can express permissions on decisions as if they were actual API calls, including applying conditions to some parameters. For more information, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.respond_decision_task_completed(
    taskToken='string',
    decisions=[
        {
            'decisionType': 'ScheduleActivityTask'|'RequestCancelActivityTask'|'CompleteWorkflowExecution',
            'scheduleActivityTaskDecisionAttributes': {
                'activityType': {
                    'name': 'string',
                    'version': 'string'
                },
                'activityId': 'string',
                'control': 'string',
                'input': 'string',
                'scheduleToCloseTimeout': 'string',
                'taskList': {
                    'name': 'string'
                },
                'taskPriority': 'string',
                'scheduleToStartTimeout': 'string',
                'startToCloseTimeout': 'string',
                'heartbeatTimeout': 'string'
            },
            'requestCancelActivityTaskDecisionAttributes': {
                'activityId': 'string'
            },
            'completeWorkflowExecutionDecisionAttributes': {
                'result': 'string'
            },
            'failWorkflowExecutionDecisionAttributes': {
                'reason': 'string',
                'details': 'string'
            },
            'cancelWorkflowExecutionDecisionAttributes': {
                'details': 'string'
            },
            'continueAsNewWorkflowExecutionDecisionAttributes': {
                'input': 'string',
                'executionStartToCloseTimeout': 'string',
```

```

        'taskList': {
            'name': 'string'
        },
        'taskPriority': 'string',
        'taskStartToCloseTimeout': 'string',
        'childPolicy': 'TERMINATE' | 'REQUEST_CANCEL' | 'ABANDON',
        'tagList': [
            'string',
        ],
        'workflowTypeVersion': 'string',
        'lambdaRole': 'string'
    },
    'recordMarkerDecisionAttributes': {
        'markerName': 'string',
        'details': 'string'
    },
    'startTimerDecisionAttributes': {
        'timerId': 'string',
        'control': 'string',
        'startToFireTimeout': 'string'
    },
    'cancelTimerDecisionAttributes': {
        'timerId': 'string'
    },
    'signalExternalWorkflowExecutionDecisionAttributes': {
        'workflowId': 'string',
        'runId': 'string',
        'signalName': 'string',
        'input': 'string',
        'control': 'string'
    },
    'requestCancelExternalWorkflowExecutionDecisionAttributes': {
        'workflowId': 'string',
        'runId': 'string',
        'control': 'string'
    },
    'startChildWorkflowExecutionDecisionAttributes': {
        'workflowType': {
            'name': 'string',
            'version': 'string'
        },
        'workflowId': 'string',
        'control': 'string',
        'input': 'string',
        'executionStartToCloseTimeout': 'string',
        'taskList': {
            'name': 'string'
        },
        'taskPriority': 'string',
        'taskStartToCloseTimeout': 'string',
        'childPolicy': 'TERMINATE' | 'REQUEST_CANCEL' | 'ABANDON',
        'tagList': [
            'string',
        ],
        'lambdaRole': 'string'
    },
    'scheduleLambdaFunctionDecisionAttributes': {
        'id': 'string',

```

```
        'name': 'string',
        'input': 'string',
        'startToCloseTimeout': 'string'
    },
    ],
    executionContext='string'
)
```

Parameters

- **taskToken** (*string*) – [REQUIRED]

The taskToken from the DecisionTask .

Warning: taskToken is generated by the service and should be treated as an opaque value. If the task is passed to another process, its taskToken must also be passed. This enables it to provide its progress and respond with results.

- **decisions** (*list*) – The list of decisions (possibly empty) made by the decider while processing this decision task. See the docs for the decision structure for details.
 - (*dict*) –

Specifies a decision made by the decider. A decision can be one of these types:

- * **CancelTimer** : cancels a previously started timer and records a `TimerCanceled` event in the history.
- * **CancelWorkflowExecution** : closes the workflow execution and records a `WorkflowExecutionCanceled` event in the history.
- * **CompleteWorkflowExecution** : closes the workflow execution and records a `WorkflowExecutionCompleted` event in the history .
- * **ContinueAsNewWorkflowExecution** : closes the workflow execution and starts a new workflow execution of the same type using the same workflow ID and a unique run ID. A `WorkflowExecutionContinuedAsNew` event is recorded in the history.
- * **FailWorkflowExecution** : closes the workflow execution and records a `WorkflowExecutionFailed` event in the history.
- * **RecordMarker** : records a `MarkerRecorded` event in the history. Markers can be used for adding custom information in the history for instance to let deciders know that they do not need to look at the history beyond the marker event.
- * **RequestCancelActivityTask** : attempts to cancel a previously scheduled activity task. If the activity task was scheduled but has not been assigned to a worker, then it will be canceled. If the activity task was already assigned to a worker, then the worker will be informed that cancellation has been requested in the response to `RecordActivityTaskHeartbeat` .
- * **RequestCancelExternalWorkflowExecution** : requests that a request be made to cancel the specified external workflow execution and records a `RequestCancelExternalWorkflowExecutionInitiated` event in the history.
- * **ScheduleActivityTask** : schedules an activity task.

- * **ScheduleLambdaFunction** : schedules a AWS Lambda function.
- * **SignalExternalWorkflowExecution** : requests a signal to be delivered to the specified external workflow execution and records a `SignalExternalWorkflowExecutionInitiated` event in the history.
- * **StartChildWorkflowExecution** : requests that a child workflow execution be started and records a `StartChildWorkflowExecutionInitiated` event in the history. The child workflow execution is a separate workflow execution with its own history.
- * **StartTimer** : starts a timer for this workflow execution and records a `TimerStarted` event in the history. This timer will fire after the specified delay and record a `TimerFired` event.

Access Control

If you grant permission to use `RespondDecisionTaskCompleted`, you can use IAM policies to express permissions for the list of decisions returned by this action as if they were members of the API. Treating decisions as a pseudo API maintains a uniform conceptual model and helps keep policies readable. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Decision Failure

Decisions can fail for several reasons

- * The ordering of decisions should follow a logical flow. Some decisions might not make sense in the current context of the workflow execution and will therefore fail.
- * A limit on your account was reached.
- * The decision lacks sufficient permissions.

One of the following events might be added to the history to indicate an error. The event attribute's **cause** parameter indicates the cause. If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- * **ScheduleActivityTaskFailed** : a `ScheduleActivityTask` decision failed. This could happen if the activity type specified in the decision is not registered, is in a deprecated state, or the decision is not properly configured.
- * **ScheduleLambdaFunctionFailed** : a `ScheduleLambdaFunction` decision failed. This could happen if the AWS Lambda function specified in the decision does not exist, or the AWS Lambda service's limits are exceeded.
- * **RequestCancelActivityTaskFailed** : a `RequestCancelActivityTask` decision failed. This could happen if there is no open activity task with the specified `activityId`.
- * **StartTimerFailed** : a `StartTimer` decision failed. This could happen if there is another open timer with the same `timerId`.
- * **CancelTimerFailed** : a `CancelTimer` decision failed. This could happen if there is no open timer with the specified `timerId`.
- * **StartChildWorkflowExecutionFailed** : a `StartChildWorkflowExecution` decision failed. This could happen if the workflow type specified is not registered, is deprecated, or the decision is not properly configured.
- * **SignalExternalWorkflowExecutionFailed** : a `SignalExternal`

WorkflowExecution decision failed. This could happen if the `workflowID` specified in the decision was incorrect.

- * **RequestCancelExternalWorkflowExecutionFailed** : a RequestCancelExternalWorkflowExecution decision failed. This could happen if the `workflowID` specified in the decision was incorrect.
- * **CancelWorkflowExecutionFailed** : a CancelWorkflowExecution decision failed. This could happen if there is an unhandled decision task pending in the workflow execution.
- * **CompleteWorkflowExecutionFailed** : a CompleteWorkflowExecution decision failed. This could happen if there is an unhandled decision task pending in the workflow execution.
- * **ContinueAsNewWorkflowExecutionFailed** : a ContinueAsNewWorkflowExecution decision failed. This could happen if there is an unhandled decision task pending in the workflow execution or the ContinueAsNewWorkflowExecution decision was not configured correctly.
- * **FailWorkflowExecutionFailed** : a FailWorkflowExecution decision failed. This could happen if there is an unhandled decision task pending in the workflow execution.

The preceding error events might occur due to an error in the decider logic, which might put the workflow execution in an unstable state. The `cause` field in the event structure for the error event indicates the cause of the error.

Note: A workflow execution may be closed by the decider by returning one of the following decisions when completing a decision task: `CompleteWorkflowExecution`, `FailWorkflowExecution`, `CancelWorkflowExecution` and `ContinueAsNewWorkflowExecution`. An `UnhandledDecision` fault will be returned if a workflow closing decision is specified and a signal or activity event had been added to the history while the decision task was being performed by the decider. Unlike the above situations which are logic issues, this fault is always possible because of race conditions in a distributed system. The right action here is to call `RespondDecisionTaskCompleted` without any decisions. This would result in another decision task with these new events included in the history. The decider should handle the new events and may decide to close the workflow execution.

How to code a decision

You code a decision by first setting the `decision_type` field to one of the above decision values, and then set the corresponding `attributes` field shown below:

- * `ScheduleActivityTaskDecisionAttributes`
- * `ScheduleLambdaFunctionDecisionAttributes`
- * `RequestCancelActivityTaskDecisionAttributes`
- * `CompleteWorkflowExecutionDecisionAttributes`
- * `FailWorkflowExecutionDecisionAttributes`
- * `CancelWorkflowExecutionDecisionAttributes`
- * `ContinueAsNewWorkflowExecutionDecisionAttributes`
- * `RecordMarkerDecisionAttributes`
- * `StartTimerDecisionAttributes`
- * `CancelTimerDecisionAttributes`

- * SignalExternalWorkflowExecutionDecisionAttributes
- * RequestCancelExternalWorkflowExecutionDecisionAttributes
- * StartChildWorkflowExecutionDecisionAttributes

* **decisionType** (*string*) – [REQUIRED]

Specifies the type of the decision.

* **scheduleActivityTaskDecisionAttributes** (*dict*) –

Provides details of the `ScheduleActivityTask` decision. It is not set for other decision types.

· **activityType** (*dict*) – [REQUIRED]

Required. The type of the activity task to schedule.

· **name** (*string*) – [REQUIRED]

The name of this activity.

Note: The combination of activity type name and version must be unique within a domain.

· **version** (*string*) – [REQUIRED]

The version of this activity.

Note: The combination of activity type name and version must be unique with in a domain.

· **activityId** (*string*) – [REQUIRED]

Required. The `activityId` of the activity task.

The specified string must not start or end with whitespace. It must not contain a `:` (colon), `/` (slash), `|` (vertical bar), or any control characters (`u0000-u001f|u007f-u009f`). Also, it must not contain the literal string `quotarnquot`.

· **control** (*string*) –

Optional. Data attached to the event that can be used by the decider in subsequent workflow tasks. This data is not sent to the activity.

· **input** (*string*) –

The input provided to the activity task.

· **scheduleToCloseTimeout** (*string*) –

The maximum duration for this activity task.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

Note: A schedule-to-close timeout for this activity task must be specified either as a default for the activity type or through this field. If neither this field is set nor a default schedule-to-close timeout was specified at registration time then a fault will be returned.

· **taskList** (*dict*) –

If set, specifies the name of the task list in which to schedule

the activity task. If not specified, the `defaultTaskList` registered with the activity type will be used.

Note: A task list for this activity task must be specified either as a default for the activity type or through this field. If neither this field is set nor a default task list was specified at registration time then a fault will be returned.

The specified string must not start or end with whitespace. It must not contain a `:` (colon), `/` (slash), `|` (vertical bar), or any control characters (`u0000-u001f|u007f-u009f`). Also, it must not contain the literal string `quotarnquot`.

- **name** (*string*) – **[REQUIRED]**

The name of the task list.

- **taskPriority** (*string*) –

Optional. If set, specifies the priority with which the activity task is to be assigned to a worker. This overrides the `defaultTaskPriority` specified when registering the activity type using `RegisterActivityType`. Valid values are integers that range from Java's `Integer.MIN_VALUE` (-2147483648) to `Integer.MAX_VALUE` (2147483647). Higher numbers indicate higher priority.

For more information about setting task priority, see [Setting Task Priority](#) in the *Amazon Simple Workflow Developer Guide*.

- **scheduleToStartTimeout** (*string*) –

Optional. If set, specifies the maximum duration the activity task can wait to be assigned to a worker. This overrides the default schedule-to-start timeout specified when registering the activity type using `RegisterActivityType`.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

Note: A schedule-to-start timeout for this activity task must be specified either as a default for the activity type or through this field. If neither this field is set nor a default schedule-to-start timeout was specified at registration time then a fault will be returned.

- **startToCloseTimeout** (*string*) –

If set, specifies the maximum duration a worker may take to process this activity task. This overrides the default start-to-close timeout specified when registering the activity type using `RegisterActivityType`.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

Note: A start-to-close timeout for this activity task must be specified either as a default for the activity type or through

this field. If neither this field is set nor a default start-to-close timeout was specified at registration time then a fault will be returned.

· **heartbeatTimeout** (*string*) –

If set, specifies the maximum time before which a worker processing a task of this type must report progress by calling `RecordActivityTaskHeartbeat`. If the timeout is exceeded, the activity task is automatically timed out. If the worker subsequently attempts to record a heartbeat or returns a result, it will be ignored. This overrides the default heartbeat timeout specified when registering the activity type using `RegisterActivityType`.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

* **requestCancelActivityTaskDecisionAttributes** (*dict*) –

Provides details of the `RequestCancelActivityTask` decision. It is not set for other decision types.

· **activityId** (*string*) – [REQUIRED]

The `activityId` of the activity task to be canceled.

* **completeWorkflowExecutionDecisionAttributes** (*dict*) –

Provides details of the `CompleteWorkflowExecution` decision. It is not set for other decision types.

· **result** (*string*) –

The result of the workflow execution. The form of the result is implementation defined.

* **failWorkflowExecutionDecisionAttributes** (*dict*) –

Provides details of the `FailWorkflowExecution` decision. It is not set for other decision types.

· **reason** (*string*) –

A descriptive reason for the failure that may help in diagnostics.

· **details** (*string*) –

Optional. Details of the failure.

* **cancelWorkflowExecutionDecisionAttributes** (*dict*) –

Provides details of the `CancelWorkflowExecution` decision. It is not set for other decision types.

· **details** (*string*) –

Optional. details of the cancellation.

* **continueAsNewWorkflowExecutionDecisionAttributes** (*dict*) –

Provides details of the `ContinueAsNewWorkflowExecution` decision. It is not set for other decision types.

· **input** (*string*) –

The input provided to the new workflow execution.

· **executionStartToCloseTimeout** (*string*) –

If set, specifies the total duration for this workflow execution. This overrides the `defaultExecutionStartToCloseTimeout` specified when registering the workflow type.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

Note: An execution start-to-close timeout for this workflow execution must be specified either as a default for the workflow type or through this field. If neither this field is set nor a default execution start-to-close timeout was specified at registration time then a fault will be returned.

- **taskList** (*dict*) –

Represents a task list.

- **name** (*string*) – **[REQUIRED]**

The name of the task list.

- **taskPriority** (*string*) –

Optional. The task priority that, if set, specifies the priority for the decision tasks for this workflow execution. This overrides the `defaultTaskPriority` specified when registering the workflow type. Valid values are integers that range from Java’s `Integer.MIN_VALUE` (-2147483648) to `Integer.MAX_VALUE` (2147483647). Higher numbers indicate higher priority.

For more information about setting task priority, see [Setting Task Priority](#) in the *Amazon Simple Workflow Developer Guide*.

- **taskStartToCloseTimeout** (*string*) –

Specifies the maximum duration of decision tasks for the new workflow execution. This parameter overrides the `defaultTaskStartToCloseTimeout` specified when registering the workflow type using `RegisterWorkflowType`.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

Note: A task start-to-close timeout for the new workflow execution must be specified either as a default for the workflow type or through this parameter. If neither this parameter is set nor a default task start-to-close timeout was specified at registration time then a fault will be returned.

- **childPolicy** (*string*) –

If set, specifies the policy to use for the child workflow executions of the new execution if it is terminated by calling the `TerminateWorkflowExecution` action explicitly or due to an expired timeout. This policy overrides the default child policy specified when registering the workflow type using `RegisterWorkflowType`.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.
- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON:** no action will be taken. The child executions will continue to run.

Note: A child policy for this workflow execution must be specified either as a default for the workflow type or through this parameter. If neither this parameter is set nor a default child policy was specified at registration time then a fault will be returned.

- **tagList** (*list*) –

The list of tags to associate with the new workflow execution. A maximum of 5 tags can be specified. You can list workflow executions with a specific tag by calling `ListOpenWorkflowExecutions` or `ListClosedWorkflowExecutions` and specifying a `TagFilter`.

- (*string*) –
- **workflowTypeVersion** (*string*) –
- **lambdaRole** (*string*) –

The ARN of an IAM role that authorizes Amazon SWF to invoke AWS Lambda functions.

Note: In order for this workflow execution to invoke AWS Lambda functions, an appropriate IAM role must be specified either as a default for the workflow type or through this field.

- * **recordMarkerDecisionAttributes** (*dict*) –

Provides details of the `RecordMarker` decision. It is not set for other decision types.

- **markerName** (*string*) – **[REQUIRED]**

Required. The name of the marker.

- **details** (*string*) –

Optional. details of the marker.

- * **startTimerDecisionAttributes** (*dict*) –

Provides details of the `StartTimer` decision. It is not set for other decision types.

- **timerId** (*string*) – **[REQUIRED]**

Required. The unique ID of the timer.

The specified string must not start or end with whitespace. It must not contain a `:` (colon), `/` (slash), `|` (vertical bar), or any control characters (`u0000-u001f` | `u007f - u009f`). Also, it must not contain the literal string `quotarnquot`.

- **control** (*string*) –

Optional. Data attached to the event that can be used by the decider in subsequent workflow tasks.

- **startToFireTimeout** (*string*) – [REQUIRED]

Required. The duration to wait before firing the timer.

The duration is specified in seconds; an integer greater than or equal to 0.

- * **cancelTimerDecisionAttributes** (*dict*) –

Provides details of the `CancelTimer` decision. It is not set for other decision types.

- **timerId** (*string*) – [REQUIRED]

Required. The unique ID of the timer to cancel.

- * **signalExternalWorkflowExecutionDecisionAttributes** (*dict*) –

Provides details of the `SignalExternalWorkflowExecution` decision. It is not set for other decision types.

- **workflowId** (*string*) – [REQUIRED]

Required. The `workflowId` of the workflow execution to be signaled.

- **runId** (*string*) –

The `runId` of the workflow execution to be signaled.

- **signalName** (*string*) – [REQUIRED]

Required. The name of the signal. The target workflow execution will use the signal name and input to process the signal.

- **input** (*string*) –

Optional. Input data to be provided with the signal. The target workflow execution will use the signal name and input data to process the signal.

- **control** (*string*) –

Optional. Data attached to the event that can be used by the decider in subsequent decision tasks.

- * **requestCancelExternalWorkflowExecutionDecisionAttributes** (*dict*) –

Provides details of the `RequestCancelExternalWorkflowExecution` decision. It is not set for other decision types.

- **workflowId** (*string*) – [REQUIRED]

Required. The `workflowId` of the external workflow execution to cancel.

- **runId** (*string*) –

The `runId` of the external workflow execution to cancel.

- **control** (*string*) –

Optional. Data attached to the event that can be used by the decider in subsequent workflow tasks.

- * **startChildWorkflowExecutionDecisionAttributes** (*dict*) –

Provides details of the `StartChildWorkflowExecution` decision. It is not set for other decision types.

- **workflowType** (*dict*) – [REQUIRED]

Required. The type of the workflow execution to be started.

- **name** (*string*) – [REQUIRED]

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) – [REQUIRED]

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **workflowId** (*string*) – [REQUIRED]

Required. The `workflowId` of the workflow execution.

The specified string must not start or end with whitespace. It must not contain a : (colon), / (slash), | (vertical bar), or any control characters (u0000-u001f | u007f - u009f). Also, it must not contain the literal string `quotarnquot`.

- **control** (*string*) –

Optional. Data attached to the event that can be used by the decider in subsequent workflow tasks. This data is not sent to the child workflow execution.

- **input** (*string*) –

The input to be provided to the workflow execution.

- **executionStartToCloseTimeout** (*string*) –

The total duration for this workflow execution. This overrides the `defaultExecutionStartToCloseTimeout` specified when registering the workflow type.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

Note: An execution start-to-close timeout for this workflow execution must be specified either as a default for the workflow type or through this parameter. If neither this parameter is set nor a default execution start-to-close timeout was specified at registration time then a fault will be returned.

- **taskList** (*dict*) –

The name of the task list to be used for decision tasks of the child workflow execution.

Note: A task list for this workflow execution must be specified either as a default for the workflow type or through this parameter. If neither this parameter is set nor a default task list was specified at registration time then a fault will be returned.

The specified string must not start or end with whitespace. It must not contain a : (colon), / (slash), | (vertical bar), or any control characters (u0000-u001f | u007f - u009f). Also, it must not contain the literal string `quotarnquot`.

- **name** (*string*) – [REQUIRED]

The name of the task list.

- **taskPriority** (*string*) –

Optional. A task priority that, if set, specifies the priority for a decision task of this workflow execution. This overrides the defaultTaskPriority specified when registering the workflow type. Valid values are integers that range from Java's Integer.MIN_VALUE (-2147483648) to Integer.MAX_VALUE (2147483647). Higher numbers indicate higher priority.

For more information about setting task priority, see [Setting Task Priority](#) in the *Amazon Simple Workflow Developer Guide*.

- **taskStartToCloseTimeout** (*string*) –

Specifies the maximum duration of decision tasks for this workflow execution. This parameter overrides the defaultTaskStartToCloseTimeout specified when registering the workflow type using RegisterWorkflowType.

The duration is specified in seconds; an integer greater than or equal to 0. The value "NONE" can be used to specify unlimited duration.

Note: A task start-to-close timeout for this workflow execution must be specified either as a default for the workflow type or through this parameter. If neither this parameter is set nor a default task start-to-close timeout was specified at registration time then a fault will be returned.

- **childPolicy** (*string*) –

Optional. If set, specifies the policy to use for the child workflow executions if the workflow execution being started is terminated by calling the TerminateWorkflowExecution action explicitly or due to an expired timeout. This policy overrides the default child policy specified when registering the workflow type using RegisterWorkflowType.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.
- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a WorkflowExecutionCancelRequested event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON:** no action will be taken. The child executions will continue to run.

Note: A child policy for this workflow execution must be specified either as a default for the workflow type or through this parameter. If neither this parameter is set nor a default child policy was specified at registration time then a fault will be returned.

- **tagList** (*list*) –

The list of tags to associate with the child workflow execu-

tion. A maximum of 5 tags can be specified. You can list workflow executions with a specific tag by calling `ListOpenWorkflowExecutions` or `ListClosedWorkflowExecutions` and specifying a `TagFilter`.

- *(string)* –
- **lambdaRole** *(string)* –

The ARN of an IAM role that authorizes Amazon SWF to invoke AWS Lambda functions.

Note: In order for this workflow execution to invoke AWS Lambda functions, an appropriate IAM role must be specified either as a default for the workflow type or through this field.

* **scheduleLambdaFunctionDecisionAttributes** *(dict)* –

Provides details of the `ScheduleLambdaFunction` decision.

Access Control

You can use IAM policies to control this decision's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- Constrain the following parameters by using a `Condition` element with the appropriate keys.
 - * `activityType.name` : String constraint. The key is `swf:activityType.name`.
 - * `activityType.version` : String constraint. The key is `swf:activityType.version`.
 - * `taskList` : String constraint. The key is `swf:taskList.name`.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **id** *(string)* – **[REQUIRED]**

Required. The SWF `id` of the AWS Lambda task.

The specified string must not start or end with whitespace. It must not contain a `:` (colon), `/` (slash), `|` (vertical bar), or any control characters (`u0000-u001f` | `u007f - u009f`). Also, it must not contain the literal string `quotarnquot`.

- **name** *(string)* – **[REQUIRED]**

Required. The name of the AWS Lambda function to invoke.

- **input** *(string)* –

The input provided to the AWS Lambda function.

- **startToCloseTimeout** *(string)* –

If set, specifies the maximum duration the function may take to execute.

- **executionContext** *(string)* – User defined context to add to workflow execution.

Returns None

signal_workflow_execution (***kwargs*)

Records a `WorkflowExecutionSignaled` event in the workflow execution history and creates a decision task for the workflow execution identified by the given domain, workflowId and runId. The event is recorded with the specified user defined signalName and input (if provided).

Note: If a runId is not specified, then the `WorkflowExecutionSignaled` event is recorded in the history of the current open workflow with the matching workflowId in the domain.

Note: If the specified workflow execution is not open, this method fails with `UnknownResource`.

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- You cannot use an IAM policy to constrain this action's parameters.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.signal_workflow_execution(
    domain='string',
    workflowId='string',
    runId='string',
    signalName='string',
    input='string'
)
```

Parameters

- **domain** (*string*) – **[REQUIRED]**
The name of the domain containing the workflow execution to signal.
- **workflowId** (*string*) – **[REQUIRED]**
The workflowId of the workflow execution to signal.
- **runId** (*string*) – The runId of the workflow execution to signal.
- **signalName** (*string*) – **[REQUIRED]**
The name of the signal. This name must be meaningful to the target workflow.
- **input** (*string*) – Data to attach to the `WorkflowExecutionSignaled` event in the target workflow execution's history.

Returns None

start_workflow_execution (***kwargs*)

Starts an execution of the workflow type in the specified domain using the provided workflowId and input data.

This action returns the newly started workflow execution.

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.

- Constrain the following parameters by using a `Condition` element with the appropriate keys. *
 - `tagList.member.0`: The key is `swf:tagList.member.0`.
 - `tagList.member.1`: The key is `swf:tagList.member.1`.
 - `tagList.member.2`: The key is `swf:tagList.member.2`.
 - `tagList.member.3`: The key is `swf:tagList.member.3`.
 - `tagList.member.4`: The key is `swf:tagList.member.4`.
 - `taskList`: String constraint. The key is `swf:taskList.name`.
 - `workflowType.name`: String constraint. The key is `swf:workflowType.name`.
 - `workflowType.version`: String constraint. The key is `swf:workflowType.version`.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.start_workflow_execution(
    domain='string',
    workflowId='string',
    workflowType={
        'name': 'string',
        'version': 'string'
    },
    taskList={
        'name': 'string'
    },
    taskPriority='string',
    input='string',
    executionStartToCloseTimeout='string',
    tagList=[
        'string',
    ],
    taskStartToCloseTimeout='string',
    childPolicy='TERMINATE' | 'REQUEST_CANCEL' | 'ABANDON',
    lambdaRole='string'
)
```

Parameters

- **domain** (*string*) – [REQUIRED]

The name of the domain in which the workflow execution is created.

- **workflowId** (*string*) – [REQUIRED]

The user defined identifier associated with the workflow execution. You can use this to associate a custom identifier with the workflow execution. You may specify the same identifier if a workflow execution is logically a *restart* of a previous execution. You cannot have two open workflow executions with the same `workflowId` at the same time.

The specified string must not start or end with whitespace. It must not contain a : (colon), / (slash), | (vertical bar), or any control characters (u0000-u001f | u007f - u009f). Also, it must not contain the literal string quotarnquot.

- **workflowType** (*dict*) – [REQUIRED]

The type of the workflow to start.

- **name** (*string*) – [REQUIRED]

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

– **version** (*string*) – **[REQUIRED]**

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

-
- **taskList** (*dict*) – The task list to use for the decision tasks generated for this workflow execution. This overrides the `defaultTaskList` specified when registering the workflow type.

Note: A task list for this workflow execution must be specified either as a default for the workflow type or through this parameter. If neither this parameter is set nor a default task list was specified at registration time then a fault will be returned.

The specified string must not start or end with whitespace. It must not contain a : (colon), / (slash), | (vertical bar), or any control characters (u0000-u001f | u007f - u009f). Also, it must not contain the literal string `quotarnquot`.

– **name** (*string*) – **[REQUIRED]**

The name of the task list.

- **taskPriority** (*string*) – The task priority to use for this workflow execution. This will override any default priority that was assigned when the workflow type was registered. If not set, then the default task priority for the workflow type will be used. Valid values are integers that range from Java’s `Integer.MIN_VALUE` (-2147483648) to `Integer.MAX_VALUE` (2147483647). Higher numbers indicate higher priority.

For more information about setting task priority, see [Setting Task Priority](#) in the *Amazon Simple Workflow Developer Guide*.

- **input** (*string*) – The input for the workflow execution. This is a free form string which should be meaningful to the workflow you are starting. This input is made available to the new workflow execution in the `WorkflowExecutionStarted` history event.
- **executionStartToCloseTimeout** (*string*) – The total duration for this workflow execution. This overrides the `defaultExecutionStartToCloseTimeout` specified when registering the workflow type.

The duration is specified in seconds; an integer greater than or equal to 0. Exceeding this limit will cause the workflow execution to time out. Unlike some of the other timeout parameters in Amazon SWF, you cannot specify a value of “NONE” for this timeout; there is a one-year max limit on the time that a workflow execution can run.

Note: An execution start-to-close timeout must be specified either through this parameter or as a default when the workflow type is registered. If neither this parameter nor a default execution start-to-close timeout is specified, a fault is returned.

-
- **tagList** (*list*) – The list of tags to associate with the workflow execution. You can specify a maximum of 5 tags. You can list workflow executions with a specific tag by calling `ListOpenWorkflowExecutions` or `ListClosedWorkflowExecutions` and specifying a `TagFilter`.

– (*string*) –

- **taskStartToCloseTimeout** (*string*) – Specifies the maximum duration of decision tasks for this workflow execution. This parameter overrides the `defaultTaskStartToCloseTimeout` specified when registering the workflow type using `RegisterWorkflowType`.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

Note: A task start-to-close timeout for this workflow execution must be specified either as a default for the workflow type or through this parameter. If neither this parameter is set nor a default task start-to-close timeout was specified at registration time then a fault will be returned.

- **childPolicy** (*string*) – If set, specifies the policy to use for the child workflow executions of this workflow execution if it is terminated, by calling the `TerminateWorkflowExecution` action explicitly or due to an expired timeout. This policy overrides the default child policy specified when registering the workflow type using `RegisterWorkflowType`.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.
- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON:** no action will be taken. The child executions will continue to run.

Note: A child policy for this workflow execution must be specified either as a default for the workflow type or through this parameter. If neither this parameter is set nor a default child policy was specified at registration time then a fault will be returned.

- **lambdaRole** (*string*) – The ARN of an IAM role that authorizes Amazon SWF to invoke AWS Lambda functions.

Note: In order for this workflow execution to invoke AWS Lambda functions, an appropriate IAM role must be specified either as a default for the workflow type or through this field.

Return type dict

Returns

Response Syntax

```
{
    'runId': 'string'
}
```

Response Structure

- (*dict*) –

Specifies the `runId` of a workflow execution.

- **runId** (*string*) –

The `runId` of a workflow execution. This ID is generated by the ser-

vice and can be used to uniquely identify the workflow execution within a domain.

`terminate_workflow_execution` (**kwargs)

Records a `WorkflowExecutionTerminated` event and forces closure of the workflow execution identified by the given domain, runId, and workflowId. The child policy, registered with the workflow type or specified when starting this execution, is applied to any open child workflow executions of this workflow execution.

Warning: If the identified workflow execution was in progress, it is terminated immediately.

Note: If a runId is not specified, then the `WorkflowExecutionTerminated` event is recorded in the history of the current open workflow with the matching workflowId in the domain.

Note: You should consider using `RequestCancelWorkflowExecution` action instead because it allows the workflow to gracefully close while `TerminateWorkflowExecution` does not.

Access Control

You can use IAM policies to control this action's access to Amazon SWF resources as follows:

- Use a `Resource` element with the domain name to limit the action to only specified domains.
- Use an `Action` element to allow or deny permission to call this action.
- You cannot use an IAM policy to constrain this action's parameters.

If the caller does not have sufficient permissions to invoke the action, or the parameter values fall outside the specified constraints, the action fails. The associated event attribute's **cause** parameter will be set to `OPERATION_NOT_PERMITTED`. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

Request Syntax

```
response = client.terminate_workflow_execution(
    domain='string',
    workflowId='string',
    runId='string',
    reason='string',
    details='string',
    childPolicy='TERMINATE' | 'REQUEST_CANCEL' | 'ABANDON'
)
```

Parameters

- **domain** (*string*) – [REQUIRED]

The domain of the workflow execution to terminate.

- **workflowId** (*string*) – [REQUIRED]

The workflowId of the workflow execution to terminate.

- **runId** (*string*) – The runId of the workflow execution to terminate.
- **reason** (*string*) – *Optional*. A descriptive reason for terminating the workflow execution.
- **details** (*string*) – *Optional*. Details for terminating the workflow execution.
- **childPolicy** (*string*) – If set, specifies the policy to use for the child workflow executions of the workflow execution being terminated. This policy overrides the child policy specified for the workflow execution at registration time or when starting the execution.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.
- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON:** no action will be taken. The child executions will continue to run.

Note: A child policy for this workflow execution must be specified either as a default for the workflow type or through this parameter. If neither this parameter is set nor a default child policy was specified at registration time then a fault will be returned.

Returns None

Paginators

The available paginators are:

- `SWF.Paginator.GetWorkflowExecutionHistory`
- `SWF.Paginator.ListActivityTypes`
- `SWF.Paginator.ListClosedWorkflowExecutions`
- `SWF.Paginator.ListDomains`
- `SWF.Paginator.ListOpenWorkflowExecutions`
- `SWF.Paginator.ListWorkflowTypes`
- `SWF.Paginator.PollForDecisionTask`

class `SWF.Paginator.GetWorkflowExecutionHistory`

```
paginator = client.get_paginator('get_workflow_execution_history')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `SWF.Client.get_workflow_execution_history()`.

Request Syntax

```
response_iterator = paginator.paginate(
    domain='string',
    execution={
        'workflowId': 'string',
        'runId': 'string'
    },
    reverseOrder=True|False,
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **domain** (*string*) – [REQUIRED]

The name of the domain containing the workflow execution.

- **execution** (*dict*) – [REQUIRED]

Specifies the workflow execution for which to return the history.

- **workflowId** (*string*) – [REQUIRED]

The user defined identifier associated with the workflow execution.

- **runId** (*string*) – [REQUIRED]

A system-generated unique identifier for the workflow execution.

- **reverseOrder** (*boolean*) – When set to `true`, returns the events in reverse order. By default the results are returned in ascending order of the `eventTimeStamp` of the events.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
  'events': [
    {
      'eventTimestamp': datetime(2015, 1, 1),
      'eventType': 'WorkflowExecutionStarted'|'WorkflowExecutionCancelRequest',
      'eventId': 123,
      'workflowExecutionStartedEventAttributes': {
        'input': 'string',
        'executionStartToCloseTimeout': 'string',
        'taskStartToCloseTimeout': 'string',
        'childPolicy': 'TERMINATE'|'REQUEST_CANCEL'|'ABANDON',
        'taskList': {
          'name': 'string'
        },
      },
      'workflowType': {
        'name': 'string',
        'version': 'string'
      },
      'tagList': [
        'string',
      ],
      'taskPriority': 'string',
      'continuedExecutionRunId': 'string',
      'parentWorkflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
      }
    }
  ]
}
```

```

    },
    'parentInitiatedEventId': 123,
    'lambdaRole': 'string'
  },
  'workflowExecutionCompletedEventAttributes': {
    'result': 'string',
    'decisionTaskCompletedEventId': 123
  },
  'completeWorkflowExecutionFailedEventAttributes': {
    'cause': 'UNHANDLED_DECISION'|'OPERATION_NOT_PERMITTED',
    'decisionTaskCompletedEventId': 123
  },
  'workflowExecutionFailedEventAttributes': {
    'reason': 'string',
    'details': 'string',
    'decisionTaskCompletedEventId': 123
  },
  'failWorkflowExecutionFailedEventAttributes': {
    'cause': 'UNHANDLED_DECISION'|'OPERATION_NOT_PERMITTED',
    'decisionTaskCompletedEventId': 123
  },
  'workflowExecutionTimedOutEventAttributes': {
    'timeoutType': 'START_TO_CLOSE',
    'childPolicy': 'TERMINATE'|'REQUEST_CANCEL'|'ABANDON'
  },
  'workflowExecutionCanceledEventAttributes': {
    'details': 'string',
    'decisionTaskCompletedEventId': 123
  },
  'cancelWorkflowExecutionFailedEventAttributes': {
    'cause': 'UNHANDLED_DECISION'|'OPERATION_NOT_PERMITTED',
    'decisionTaskCompletedEventId': 123
  },
  'workflowExecutionContinuedAsNewEventAttributes': {
    'input': 'string',
    'decisionTaskCompletedEventId': 123,
    'newExecutionRunId': 'string',
    'executionStartToCloseTimeout': 'string',
    'taskList': {
      'name': 'string'
    },
    'taskPriority': 'string',
    'taskStartToCloseTimeout': 'string',
    'childPolicy': 'TERMINATE'|'REQUEST_CANCEL'|'ABANDON',
    'tagList': [
      'string',
    ],
    'workflowType': {
      'name': 'string',
      'version': 'string'
    },
    'lambdaRole': 'string'
  },
  'continueAsNewWorkflowExecutionFailedEventAttributes': {
    'cause': 'UNHANDLED_DECISION'|'WORKFLOW_TYPE_DEPRECATED'|'WORKFLOW'
    'decisionTaskCompletedEventId': 123
  },
  'workflowExecutionTerminatedEventAttributes': {

```

```
        'reason': 'string',
        'details': 'string',
        'childPolicy': 'TERMINATE'|'REQUEST_CANCEL'|'ABANDON',
        'cause': 'CHILD_POLICY_APPLIED'|'EVENT_LIMIT_EXCEEDED'|'OPERATOR_I
    },
    'workflowExecutionCancelRequestedEventAttributes': {
        'externalWorkflowExecution': {
            'workflowId': 'string',
            'runId': 'string'
        },
        'externalInitiatedEventId': 123,
        'cause': 'CHILD_POLICY_APPLIED'
    },
    'decisionTaskScheduledEventAttributes': {
        'taskList': {
            'name': 'string'
        },
        'taskPriority': 'string',
        'startToCloseTimeout': 'string'
    },
    'decisionTaskStartedEventAttributes': {
        'identity': 'string',
        'scheduledEventId': 123
    },
    'decisionTaskCompletedEventAttributes': {
        'executionContext': 'string',
        'scheduledEventId': 123,
        'startedEventId': 123
    },
    'decisionTaskTimedOutEventAttributes': {
        'timeoutType': 'START_TO_CLOSE',
        'scheduledEventId': 123,
        'startedEventId': 123
    },
    'activityTaskScheduledEventAttributes': {
        'activityType': {
            'name': 'string',
            'version': 'string'
        },
        'activityId': 'string',
        'input': 'string',
        'control': 'string',
        'scheduleToStartTimeout': 'string',
        'scheduleToCloseTimeout': 'string',
        'startToCloseTimeout': 'string',
        'taskList': {
            'name': 'string'
        },
        'taskPriority': 'string',
        'decisionTaskCompletedEventId': 123,
        'heartbeatTimeout': 'string'
    },
    'activityTaskStartedEventAttributes': {
        'identity': 'string',
        'scheduledEventId': 123
    },
    'activityTaskCompletedEventAttributes': {
        'result': 'string',
```



```

        'scheduledEventId': 123,
        'startedEventId': 123
    },
    'activityTaskFailedEventAttributes': {
        'reason': 'string',
        'details': 'string',
        'scheduledEventId': 123,
        'startedEventId': 123
    },
    'activityTaskTimedOutEventAttributes': {
        'timeoutType': 'START_TO_CLOSE'|'SCHEDULE_TO_START'|'SCHEDULE_TO_C
        'scheduledEventId': 123,
        'startedEventId': 123,
        'details': 'string'
    },
    'activityTaskCanceledEventAttributes': {
        'details': 'string',
        'scheduledEventId': 123,
        'startedEventId': 123,
        'latestCancelRequestedEventId': 123
    },
    'activityTaskCancelRequestedEventAttributes': {
        'decisionTaskCompletedEventId': 123,
        'activityId': 'string'
    },
    'workflowExecutionSignaledEventAttributes': {
        'signalName': 'string',
        'input': 'string',
        'externalWorkflowExecution': {
            'workflowId': 'string',
            'runId': 'string'
        },
        'externalInitiatedEventId': 123
    },
    'markerRecordedEventAttributes': {
        'markerName': 'string',
        'details': 'string',
        'decisionTaskCompletedEventId': 123
    },
    'recordMarkerFailedEventAttributes': {
        'markerName': 'string',
        'cause': 'OPERATION_NOT_PERMITTED',
        'decisionTaskCompletedEventId': 123
    },
    'timerStartedEventAttributes': {
        'timerId': 'string',
        'control': 'string',
        'startToFireTimeout': 'string',
        'decisionTaskCompletedEventId': 123
    },
    'timerFiredEventAttributes': {
        'timerId': 'string',
        'startedEventId': 123
    },
    'timerCanceledEventAttributes': {
        'timerId': 'string',
        'startedEventId': 123,
        'decisionTaskCompletedEventId': 123
    }

```

```
},
'startChildWorkflowExecutionInitiatedEventAttributes': {
    'workflowId': 'string',
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'control': 'string',
    'input': 'string',
    'executionStartToCloseTimeout': 'string',
    'taskList': {
        'name': 'string'
    },
    'taskPriority': 'string',
    'decisionTaskCompletedEventId': 123,
    'childPolicy': 'TERMINATE'|'REQUEST_CANCEL'|'ABANDON',
    'taskStartToCloseTimeout': 'string',
    'tagList': [
        'string',
    ],
    'lambdaRole': 'string'
},
'childWorkflowExecutionStartedEventAttributes': {
    'workflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'initiatedEventId': 123
},
'childWorkflowExecutionCompletedEventAttributes': {
    'workflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'result': 'string',
    'initiatedEventId': 123,
    'startedEventId': 123
},
'childWorkflowExecutionFailedEventAttributes': {
    'workflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'reason': 'string',
    'details': 'string',
    'initiatedEventId': 123,
```

```

        'startedEventId': 123
    },
    'childWorkflowExecutionTimedOutEventAttributes': {
        'workflowExecution': {
            'workflowId': 'string',
            'runId': 'string'
        },
        'workflowType': {
            'name': 'string',
            'version': 'string'
        },
        'timeoutType': 'START_TO_CLOSE',
        'initiatedEventId': 123,
        'startedEventId': 123
    },
    'childWorkflowExecutionCanceledEventAttributes': {
        'workflowExecution': {
            'workflowId': 'string',
            'runId': 'string'
        },
        'workflowType': {
            'name': 'string',
            'version': 'string'
        },
        'details': 'string',
        'initiatedEventId': 123,
        'startedEventId': 123
    },
    'childWorkflowExecutionTerminatedEventAttributes': {
        'workflowExecution': {
            'workflowId': 'string',
            'runId': 'string'
        },
        'workflowType': {
            'name': 'string',
            'version': 'string'
        },
        'initiatedEventId': 123,
        'startedEventId': 123
    },
    'signalExternalWorkflowExecutionInitiatedEventAttributes': {
        'workflowId': 'string',
        'runId': 'string',
        'signalName': 'string',
        'input': 'string',
        'decisionTaskCompletedEventId': 123,
        'control': 'string'
    },
    'externalWorkflowExecutionSignaledEventAttributes': {
        'workflowExecution': {
            'workflowId': 'string',
            'runId': 'string'
        },
        'initiatedEventId': 123
    },
    'signalExternalWorkflowExecutionFailedEventAttributes': {
        'workflowId': 'string',
        'runId': 'string',

```

```
        'cause': 'UNKNOWN_EXTERNAL_WORKFLOW_EXECUTION' | 'SIGNAL_EXTERNAL_WO
        'initiatedEventId': 123,
        'decisionTaskCompletedEventId': 123,
        'control': 'string'
    },
    'externalWorkflowExecutionCancelRequestedEventAttributes': {
        'workflowExecution': {
            'workflowId': 'string',
            'runId': 'string'
        },
        'initiatedEventId': 123
    },
    'requestCancelExternalWorkflowExecutionInitiatedEventAttributes': {
        'workflowId': 'string',
        'runId': 'string',
        'decisionTaskCompletedEventId': 123,
        'control': 'string'
    },
    'requestCancelExternalWorkflowExecutionFailedEventAttributes': {
        'workflowId': 'string',
        'runId': 'string',
        'cause': 'UNKNOWN_EXTERNAL_WORKFLOW_EXECUTION' | 'REQUEST_CANCEL_EXT
        'initiatedEventId': 123,
        'decisionTaskCompletedEventId': 123,
        'control': 'string'
    },
    'scheduleActivityTaskFailedEventAttributes': {
        'activityType': {
            'name': 'string',
            'version': 'string'
        },
        'activityId': 'string',
        'cause': 'ACTIVITY_TYPE_DEPRECATED' | 'ACTIVITY_TYPE_DOES_NOT_EXIST'
        'decisionTaskCompletedEventId': 123
    },
    'requestCancelActivityTaskFailedEventAttributes': {
        'activityId': 'string',
        'cause': 'ACTIVITY_ID_UNKNOWN' | 'OPERATION_NOT_PERMITTED',
        'decisionTaskCompletedEventId': 123
    },
    'startTimerFailedEventAttributes': {
        'timerId': 'string',
        'cause': 'TIMER_ID_ALREADY_IN_USE' | 'OPEN_TIMERS_LIMIT_EXCEEDED' | 'T
        'decisionTaskCompletedEventId': 123
    },
    'cancelTimerFailedEventAttributes': {
        'timerId': 'string',
        'cause': 'TIMER_ID_UNKNOWN' | 'OPERATION_NOT_PERMITTED',
        'decisionTaskCompletedEventId': 123
    },
    'startChildWorkflowExecutionFailedEventAttributes': {
        'workflowType': {
            'name': 'string',
            'version': 'string'
        },
        'cause': 'WORKFLOW_TYPE_DOES_NOT_EXIST' | 'WORKFLOW_TYPE_DEPRECATED'
        'workflowId': 'string',
        'initiatedEventId': 123,
```

```

        'decisionTaskCompletedEventId': 123,
        'control': 'string'
    },
    'lambdaFunctionScheduledEventAttributes': {
        'id': 'string',
        'name': 'string',
        'input': 'string',
        'startToCloseTimeout': 'string',
        'decisionTaskCompletedEventId': 123
    },
    'lambdaFunctionStartedEventAttributes': {
        'scheduledEventId': 123
    },
    'lambdaFunctionCompletedEventAttributes': {
        'scheduledEventId': 123,
        'startedEventId': 123,
        'result': 'string'
    },
    'lambdaFunctionFailedEventAttributes': {
        'scheduledEventId': 123,
        'startedEventId': 123,
        'reason': 'string',
        'details': 'string'
    },
    'lambdaFunctionTimedOutEventAttributes': {
        'scheduledEventId': 123,
        'startedEventId': 123,
        'timeoutType': 'START_TO_CLOSE'
    },
    'scheduleLambdaFunctionFailedEventAttributes': {
        'id': 'string',
        'name': 'string',
        'cause': 'ID_ALREADY_IN_USE'|'OPEN_LAMBDA_FUNCTIONS_LIMIT_EXCEEDED',
        'decisionTaskCompletedEventId': 123
    },
    'startLambdaFunctionFailedEventAttributes': {
        'scheduledEventId': 123,
        'cause': 'ASSUME_ROLE_FAILED',
        'message': 'string'
    }
},
    'NextToken': 'string'
}

```

Response Structure

- (*dict*) –

Paginated representation of a workflow history for a workflow execution. This is the up to date, complete and authoritative record of the events related to all tasks and events in the life of the workflow execution.

- **events** (*list*) –

The list of history events.

- * (*dict*) –

Event within a workflow execution. A history event can be one of these types:

- **WorkflowExecutionStarted** : The workflow execution was

started.

- **WorkflowExecutionCompleted** : The workflow execution was closed due to successful completion.
- **WorkflowExecutionFailed** : The workflow execution closed due to a failure.
- **WorkflowExecutionTimedOut** : The workflow execution was closed because a time out was exceeded.
- **WorkflowExecutionCanceled** : The workflow execution was successfully canceled and closed.
- **WorkflowExecutionTerminated** : The workflow execution was terminated.
- **WorkflowExecutionContinuedAsNew** : The workflow execution was closed and a new execution of the same type was created with the same workflowId.
- **WorkflowExecutionCancelRequested** : A request to cancel this workflow execution was made.
- **DecisionTaskScheduled** : A decision task was scheduled for the workflow execution.
- **DecisionTaskStarted** : The decision task was dispatched to a decider.
- **DecisionTaskCompleted** : The decider successfully completed a decision task by calling `RespondDecisionTaskCompleted`.
- **DecisionTaskTimedOut** : The decision task timed out.
- **ActivityTaskScheduled** : An activity task was scheduled for execution.
- **ScheduleActivityTaskFailed** : Failed to process `ScheduleActivityTask` decision. This happens when the decision is not configured properly, for example the activity type specified is not registered.
- **ActivityTaskStarted** : The scheduled activity task was dispatched to a worker.
- **ActivityTaskCompleted** : An activity worker successfully completed an activity task by calling `RespondActivityTaskCompleted`.
- **ActivityTaskFailed** : An activity worker failed an activity task by calling `RespondActivityTaskFailed`.
- **ActivityTaskTimedOut** : The activity task timed out.
- **ActivityTaskCanceled** : The activity task was successfully canceled.
- **ActivityTaskCancelRequested** : A `RequestCancelActivityTask` decision was received by the system.
- **RequestCancelActivityTaskFailed** : Failed to process `RequestCancelActivityTask` decision. This happens when the decision is not configured properly.
- **WorkflowExecutionSignaled** : An external signal was received for the workflow execution.
- **MarkerRecorded** : A marker was recorded in the workflow history as the result of a `RecordMarker` decision.
- **TimerStarted** : A timer was started for the workflow execution due to a `StartTimer` decision.
- **StartTimerFailed** : Failed to process `StartTimer` decision. This happens when the decision is not configured properly,

- for example a timer already exists with the specified timer ID.
- **TimerFired** : A timer, previously started for this workflow execution, fired.
 - **TimerCanceled** : A timer, previously started for this workflow execution, was successfully canceled.
 - **CancelTimerFailed** : Failed to process CancelTimer decision. This happens when the decision is not configured properly, for example no timer exists with the specified timer ID.
 - **StartChildWorkflowExecutionInitiated** : A request was made to start a child workflow execution.
 - **StartChildWorkflowExecutionFailed** : Failed to process StartChildWorkflowExecution decision. This happens when the decision is not configured properly, for example the workflow type specified is not registered.
 - **ChildWorkflowExecutionStarted** : A child workflow execution was successfully started.
 - **ChildWorkflowExecutionCompleted** : A child workflow execution, started by this workflow execution, completed successfully and was closed.
 - **ChildWorkflowExecutionFailed** : A child workflow execution, started by this workflow execution, failed to complete successfully and was closed.
 - **ChildWorkflowExecutionTimedOut** : A child workflow execution, started by this workflow execution, timed out and was closed.
 - **ChildWorkflowExecutionCanceled** : A child workflow execution, started by this workflow execution, was canceled and closed.
 - **ChildWorkflowExecutionTerminated** : A child workflow execution, started by this workflow execution, was terminated.
 - **SignalExternalWorkflowExecutionInitiated** : A request to signal an external workflow was made.
 - **ExternalWorkflowExecutionSignaled** : A signal, requested by this workflow execution, was successfully delivered to the target external workflow execution.
 - **SignalExternalWorkflowExecutionFailed** : The request to signal an external workflow execution failed.
 - **RequestCancelExternalWorkflowExecutionInitiated** : A request was made to request the cancellation of an external workflow execution.
 - **ExternalWorkflowExecutionCancelRequested** : Request to cancel an external workflow execution was successfully delivered to the target execution.
 - **RequestCancelExternalWorkflowExecutionFailed** : Request to cancel an external workflow execution failed.
 - **LambdaFunctionScheduled** : An AWS Lambda function was scheduled for execution.
 - **LambdaFunctionStarted** : The scheduled function was invoked in the AWS Lambda service.
 - **LambdaFunctionCompleted** : The AWS Lambda function successfully completed.
 - **LambdaFunctionFailed** : The AWS Lambda function execution failed.

- **LambdaFunctionTimedOut** : The AWS Lambda function execution timed out.
- **ScheduleLambdaFunctionFailed** : Failed to process ScheduleLambdaFunction decision. This happens when the workflow execution does not have the proper IAM role attached to invoke AWS Lambda functions.
- **StartLambdaFunctionFailed** : Failed to invoke the scheduled function in the AWS Lambda service. This happens when the AWS Lambda service is not available in the current region, or received too many requests.
- **eventTimestamp** (*datetime*) –
The date and time when the event occurred.
- **eventType** (*string*) –
The type of the history event.
- **eventId** (*integer*) –
The system generated ID of the event. This ID uniquely identifies the event within the workflow execution history.
- **workflowExecutionStartedEventAttributes** (*dict*) –
If the event is of type `WorkflowExecutionStarted` then this member is set and provides detailed information about the event. It is not set for other event types.
- **input** (*string*) –
The input provided to the workflow execution (if any).
- **executionStartToCloseTimeout** (*string*) –
The maximum duration for this workflow execution.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.
- **taskStartToCloseTimeout** (*string*) –
The maximum duration of decision tasks for this workflow type.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.
- **childPolicy** (*string*) –
The policy to use for the child workflow executions if this workflow execution is terminated, by calling the `TerminateWorkflowExecution` action explicitly or due to an expired timeout.

The supported child policies are:
 - **TERMINATE**: the child executions will be terminated.
 - **REQUEST_CANCEL**: a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
 - **ABANDON**: no action will be taken. The child executions will continue to run.

- **taskList** (*dict*) –

The name of the task list for scheduling the decision tasks for this workflow execution.

- **name** (*string*) –

The name of the task list.

- **workflowType** (*dict*) –

The workflow type of this execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **tagList** (*list*) –

The list of tags associated with this workflow execution. An execution can have up to 5 tags.

- (*string*) –

- **taskPriority** (*string*) –

- **continuedExecutionRunId** (*string*) –

If this workflow execution was started due to a `ContinueAsNewWorkflowExecution` decision, then it contains the `runId` of the previous workflow execution that was closed and continued as this execution.

- **parentWorkflowExecution** (*dict*) –

The source workflow execution that started this workflow execution. The member is not set if the workflow execution was not started by a workflow.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **parentInitiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this workflow execution. The source event with this ID can be found in the history of the source workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **lambdaRole** (*string*) –

The IAM role attached to this workflow execution to use when invoking AWS Lambda functions.

- **workflowExecutionCompletedEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionCompleted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **result** (*string*) –

The result produced by the workflow execution upon successful completion.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CompleteWorkflowExecution` decision to complete this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **completeWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `CompleteWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CompleteWorkflowExecution` decision to complete this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **workflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **reason** (*string*) –

The descriptive reason provided for the failure (if any).

- **details** (*string*) –

The details of the failure (if any).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the

`FailWorkflowExecution` decision to fail this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **failWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `FailWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `FailWorkflowExecution` decision to fail this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **workflowExecutionTimedOutEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionTimedOut` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timeoutType** (*string*) –

The type of timeout that caused this event.

- **childPolicy** (*string*) –

The policy used for the child workflow executions of this workflow execution.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.
- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON:** no action will be taken. The child executions will continue to run.
- **workflowExecutionCanceledEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionCanceled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **details** (*string*) –

Details for the cancellation (if any).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event cor-

responding to the decision task that resulted in the `CancelWorkflowExecution` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **cancelWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `CancelWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CancelWorkflowExecution` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **workflowExecutionContinuedAsNewEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionContinuedAsNew` then this member is set and provides detailed information about the event. It is not set for other event types.

- **input** (*string*) –

The input provided to the new workflow execution.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `ContinueAsNewWorkflowExecution` decision that started this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **newExecutionRunId** (*string*) –

The `runId` of the new workflow execution.

- **executionStartToCloseTimeout** (*string*) –

The total duration allowed for the new workflow execution.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **taskList** (*dict*) –

Represents a task list.

- **name** (*string*) –

The name of the task list.

- **taskPriority** (*string*) –
- **taskStartToCloseTimeout** (*string*) –

The maximum duration of decision tasks for the new workflow execution.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **childPolicy** (*string*) –

The policy to use for the child workflow executions of the new execution if it is terminated by calling the `TerminateWorkflowExecution` action explicitly or due to an expired timeout.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.
- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON:** no action will be taken. The child executions will continue to run.
- **tagList** (*list*) –

The list of tags associated with the new workflow execution.

- (*string*) –
- **workflowType** (*dict*) –

Represents a workflow type.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique within a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique within a domain.

- **lambdaRole** (*string*) –

The IAM role attached to this workflow execution to use when invoking AWS Lambda functions.

- **continueAsNewWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `ContinueAsNewWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `ContinueAsNewWorkflowExecution` decision that started this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **workflowExecutionTerminatedEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionTerminated` then this member is set and provides detailed information about the event. It is not set for other event types.

- **reason** (*string*) –

The reason provided for the termination (if any).

- **details** (*string*) –

The details provided for the termination (if any).

- **childPolicy** (*string*) –

The policy used for the child workflow executions of this workflow execution.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.
- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON:** no action will be taken. The child executions will continue to run.
- **cause** (*string*) –

If set, indicates that the workflow execution was automatically terminated, and specifies the cause. This happens if the parent workflow execution times out or is terminated and the child policy is set to terminate child executions.

- **workflowExecutionCancelRequestedEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionCancelRequested` then this member is set and provides detailed information about the event. It is not set for other event types.

- **externalWorkflowExecution** (*dict*) –

The external workflow execution for which the cancellation was requested.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **externalInitiatedEventId** (*integer*) –

The ID of the `RequestCancelExternalWorkflowExecutionInitiated` event corresponding to the `RequestCancelExternalWorkflowExecution` decision to cancel this workflow execution. The source event with this ID can be found in the history of the source workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **cause** (*string*) –

If set, indicates that the request to cancel the workflow execution was automatically generated, and specifies the cause. This happens if the parent workflow execution times out or is terminated, and the child policy is set to cancel child executions.

- **decisionTaskScheduledEventAttributes** (*dict*) –

If the event is of type `DecisionTaskScheduled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **taskList** (*dict*) –

The name of the task list in which the decision task was scheduled.

- **name** (*string*) –

The name of the task list.

- **taskPriority** (*string*) –

Optional. A task priority that, if set, specifies the priority for this decision task. Valid values are integers that range from Java's `Integer.MIN_VALUE` (-2147483648) to `Integer.MAX_VALUE` (2147483647). Higher numbers indicate higher priority.

For more information about setting task priority, see [Setting Task Priority](#) in the *Amazon Simple Workflow Developer Guide*.

- **startToCloseTimeout** (*string*) –

The maximum duration for this decision task. The task is considered timed out if it does not completed within this duration.

The duration is specified in seconds; an integer greater than or equal to 0. The value "NONE" can be used to specify unlimited duration.

- **decisionTaskStartedEventAttributes** (*dict*) –

If the event is of type `DecisionTaskStarted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **identity** (*string*) –

Identity of the decider making the request. This enables diagnostic tracing when problems arise. The form of this identity is user defined.

- **scheduledEventId** (*integer*) –

The ID of the `DecisionTaskScheduled` event that was recorded when this decision task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **decisionTaskCompletedEventAttributes** (*dict*) –

If the event is of type `DecisionTaskCompleted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **executionContext** (*string*) –

User defined context for the workflow execution.

- **scheduledEventId** (*integer*) –

The ID of the `DecisionTaskScheduled` event that was recorded when this decision task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `DecisionTaskStarted` event recorded when this decision task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **decisionTaskTimedOutEventAttributes** (*dict*) –

If the event is of type `DecisionTaskTimedOut` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timeoutType** (*string*) –

The type of timeout that expired before the decision task could be completed.

- **scheduledEventId** (*integer*) –

The ID of the `DecisionTaskScheduled` event that was recorded when this decision task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `DecisionTaskStarted` event recorded when this decision task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **activityTaskScheduledEventAttributes** (*dict*) –

If the event is of type `ActivityTaskScheduled` then this member is set and provides detailed information about

the event. It is not set for other event types.

- **activityType** (*dict*) –

The type of the activity task.

- **name** (*string*) –

The name of this activity.

Note: The combination of activity type name and version must be unique within a domain.

- **version** (*string*) –

The version of this activity.

Note: The combination of activity type name and version must be unique with in a domain.

- **activityId** (*string*) –

The unique ID of the activity task.

- **input** (*string*) –

The input provided to the activity task.

- **control** (*string*) –

Optional. Data attached to the event that can be used by the decider in subsequent workflow tasks. This data is not sent to the activity.

- **scheduleToStartTimeout** (*string*) –

The maximum amount of time the activity task can wait to be assigned to a worker.

- **scheduleToCloseTimeout** (*string*) –

The maximum amount of time for this activity task.

- **startToCloseTimeout** (*string*) –

The maximum amount of time a worker may take to process the activity task.

- **taskList** (*dict*) –

The task list in which the activity task has been scheduled.

- **name** (*string*) –

The name of the task list.

- **taskPriority** (*string*) –

Optional. The priority to assign to the scheduled activity task. If set, this will override any default priority value that was assigned when the activity type was registered.

Valid values are integers that range from Java's `Integer.MIN_VALUE` (-2147483648) to `Integer.MAX_VALUE` (2147483647). Higher numbers indicate higher priority.

For more information about setting task priority, see [Setting Task Priority](#) in the *Amazon Simple Workflow Developer Guide*.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision that resulted in the scheduling of this activity task. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **heartbeatTimeout** (*string*) –

The maximum time before which the worker processing this task must report progress by calling `RecordActivityTaskHeartbeat`. If the timeout is exceeded, the activity task is automatically timed out. If the worker subsequently attempts to record a heartbeat or return a result, it will be ignored.

- **activityTaskStartedEventAttributes** (*dict*) –

If the event is of type `ActivityTaskStarted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **identity** (*string*) –

Identity of the worker that was assigned this task. This aids diagnostics when problems arise. The form of this identity is user defined.

- **scheduledEventId** (*integer*) –

The ID of the `ActivityTaskScheduled` event that was recorded when this activity task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **activityTaskCompletedEventAttributes** (*dict*) –

If the event is of type `ActivityTaskCompleted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **result** (*string*) –

The results of the activity task (if any).

- **scheduledEventId** (*integer*) –

The ID of the `ActivityTaskScheduled` event that was recorded when this activity task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ActivityTaskStarted` event recorded when this activity task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **activityTaskFailedEventAttributes** (*dict*) –

If the event is of type `ActivityTaskFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **reason** (*string*) –

The reason provided for the failure (if any).

- **details** (*string*) –

The details of the failure (if any).

- **scheduledEventId** (*integer*) –

The ID of the `ActivityTaskScheduled` event that was recorded when this activity task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ActivityTaskStarted` event recorded when this activity task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **activityTaskTimedOutEventAttributes** (*dict*) –

If the event is of type `ActivityTaskTimedOut` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timeoutType** (*string*) –

The type of the timeout that caused this event.

- **scheduledEventId** (*integer*) –

The ID of the `ActivityTaskScheduled` event that was recorded when this activity task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ActivityTaskStarted` event recorded when this activity task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **details** (*string*) –

Contains the content of the `details` parameter for the last call made by the activity to `RecordActivityTaskHeartbeat`.

- **activityTaskCanceledEventAttributes** (*dict*) –

If the event is of type `ActivityTaskCanceled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **details** (*string*) –

Details of the cancellation (if any).

- **scheduledEventId** (*integer*) –

The ID of the `ActivityTaskScheduled` event that was recorded when this activity task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ActivityTaskStarted` event recorded when this activity task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **latestCancelRequestedEventId** (*integer*) –

If set, contains the ID of the last `ActivityTaskCancelRequested` event recorded for this activity task. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **activityTaskCancelRequestedEventAttributes** (*dict*) –

If the event is of type `ActivityTaskCancelRequested` then this member is set and provides detailed information about the event. It is not set for other event types.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RequestCancelActivityTask` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **activityId** (*string*) –

The unique ID of the task.

- **workflowExecutionSignaledEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionSignaled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **signalName** (*string*) –

The name of the signal received. The decider can use the signal name and inputs to determine how to process the signal.

- **input** (*string*) –

Inputs provided with the signal (if any). The decider can use the signal name and inputs to determine how to process the signal.

- **externalWorkflowExecution** (*dict*) –

The workflow execution that sent the signal. This is set only if the signal was sent by another workflow execution.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **externalInitiatedEventId** (*integer*) –

The ID of the `SignalExternalWorkflowExecutionInitiated` event corresponding to the `SignalExternalWorkflow` decision to signal this workflow execution. The source event with this ID can be found in the history of the source workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event. This field is set only if the signal was initiated by another workflow execution.

- **markerRecordedEventAttributes** (*dict*) –

If the event is of type `MarkerRecorded` then this member is set and provides detailed information about the event. It is not set for other event types.

- **markerName** (*string*) –

The name of the marker.

- **details** (*string*) –

Details of the marker (if any).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RecordMarker` decision that requested this marker. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **recordMarkerFailedEventAttributes** (*dict*) –

If the event is of type `DecisionTaskFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **markerName** (*string*) –

The marker's name.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RecordMarkerFailed` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **timerStartedEventAttributes** (*dict*) –

If the event is of type `TimerStarted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timerId** (*string*) –

The unique ID of the timer that was started.

- **control** (*string*) –

Optional. Data attached to the event that can be used by the decider in subsequent workflow tasks.

- **startToFireTimeout** (*string*) –

The duration of time after which the timer will fire.

The duration is specified in seconds; an integer greater than or equal to 0.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `StartTimer` decision for this activity task. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **timerFiredEventAttributes** (*dict*) –

If the event is of type `TimerFired` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timerId** (*string*) –

The unique ID of the timer that fired.

- **startedEventId** (*integer*) –

The ID of the `TimerStarted` event that was recorded when this timer was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **timerCanceledEventAttributes** (*dict*) –

If the event is of type `TimerCanceled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timerId** (*string*) –

The unique ID of the timer that was canceled.

- **startedEventId** (*integer*) –

The ID of the `TimerStarted` event that was recorded when this timer was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CancelTimer` decision to cancel this timer. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startChildWorkflowExecutionInitiatedEventAttributes** (*dict*) –

If the event is of type `StartChildWorkflowExecutionInitiated` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowId** (*string*) –

The `workflowId` of the child workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **control** (*string*) –

Optional. Data attached to the event that can be used by the decider in subsequent decision tasks. This data is not sent to the activity.

- **input** (*string*) –

The inputs provided to the child workflow execution (if any).

- **executionStartToCloseTimeout** (*string*) –

The maximum duration for the child workflow execution. If the workflow execution is not closed within this duration, it will be timed out and force terminated.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **taskList** (*dict*) –

The name of the task list used for the decision tasks of the child workflow execution.

- **name** (*string*) –

The name of the task list.

- **taskPriority** (*string*) –

Optional. The priority assigned for the decision tasks for this workflow execution. Valid values are integers that range from Java’s `Integer.MIN_VALUE` (-2147483648) to `Integer.MAX_VALUE` (2147483647). Higher numbers indicate higher priority.

For more information about setting task priority, see [Setting Task Priority](#) in the *Amazon Simple Workflow Developer Guide*.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `StartChildWorkflowExecution` decision to request this child workflow execution. This information can be useful for diagnosing problems by tracing back the cause of events.

- **childPolicy** (*string*) –

The policy to use for the child workflow executions if this execution gets terminated by explicitly calling the `TerminateWorkflowExecution` action or due to an expired timeout.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.

- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.

- **ABANDON:** no action will be taken. The child executions will continue to run.

- **taskStartToCloseTimeout** (*string*) –

The maximum duration allowed for the decision tasks for this workflow execution.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **tagList** (*list*) –

The list of tags to associated with the child workflow execution.

- (*string*) –

- **lambdaRole** (*string*) –

The IAM role attached to this workflow execution to use when invoking AWS Lambda functions.

- **childWorkflowExecutionStartedEventAttributes** (*dict*) –

If the event is of type `ChildWorkflowExecutionStarted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that was started.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **childWorkflowExecutionCompletedEventAttributes** (*dict*) –

–

If the event is of type `ChildWorkflowExecutionCompleted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that was completed.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **result** (*string*) –

The result of the child workflow execution (if any).

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ChildWorkflowExecutionStarted` event recorded when this child workflow execution was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **childWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `ChildWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that failed.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **reason** (*string*) –

The reason for the failure (if provided).

- **details** (*string*) –

The details of the failure (if provided).

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ChildWorkflowExecutionStarted` event recorded when this child workflow execution was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **childWorkflowExecutionTimedOutEventAttributes** (*dict*) –

If the event is of type `ChildWorkflowExecutionTimedOut` then this

member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that timed out.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **timeoutType** (*string*) –

The type of the timeout that caused the child workflow execution to time out.

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ChildWorkflowExecutionStarted` event recorded when this child workflow execution was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **childWorkflowExecutionCanceledEventAttributes** (*dict*) –

If the event is of type `ChildWorkflowExecutionCanceled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that was canceled.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **details** (*string*) –

Details of the cancellation (if provided).

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ChildWorkflowExecutionStarted` event recorded when this child workflow execution was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **childWorkflowExecutionTerminatedEventAttributes** (*dict*) –

If the event is of type `ChildWorkflowExecutionTerminated` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that was terminated.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ChildWorkflowExecutionStarted` event recorded when this child workflow execution was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **signalExternalWorkflowExecutionInitiatedEventAttributes** (*dict*) –

If the event is of type `SignalExternalWorkflowExecutionInitiated` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowId** (*string*) –

The `workflowId` of the external workflow execution.

- **runId** (*string*) –

The `runId` of the external workflow execution to send the signal to.

- **signalName** (*string*) –

The name of the signal.

- **input** (*string*) –

Input provided to the signal (if any).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `SignalExternalWorkflowExecution` decision for this signal. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **control** (*string*) –

Optional. data attached to the event that can be used by the decider in subsequent decision tasks.

- **externalWorkflowExecutionSignaledEventAttributes** (*dict*) –

If the event is of type `ExternalWorkflowExecutionSignaled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The external workflow execution that the signal was delivered to.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **initiatedEventId** (*integer*) –

The ID of the `SignalExternalWorkflowExecutionInitiated` event corresponding to the `SignalExternalWorkflowExecution` decision to request this signal. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **signalExternalWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `SignalExternalWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowId** (*string*) –

The `workflowId` of the external workflow execution that the signal was being delivered to.

- **runId** (*string*) –

The `runId` of the external workflow execution that the signal was being delivered to.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **initiatedEventId** (*integer*) –

The ID of the `SignalExternalWorkflowExecutionInitiated` event corresponding to the `SignalExternalWorkflowExecution` decision

to request this signal. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `SignalExternalWorkflowExecution` decision for this signal. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **control** (*string*) –
- **externalWorkflowExecutionCancelRequestedEventAttributes** (*dict*) –

If the event is of type `ExternalWorkflowExecutionCancelRequested` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The external workflow execution to which the cancellation request was delivered.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **initiatedEventId** (*integer*) –

The ID of the `RequestCancelExternalWorkflowExecutionInitiated` event corresponding to the `RequestCancelExternalWorkflowExecution` decision to cancel this external workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **requestCancelExternalWorkflowExecutionInitiatedEventAttributes** (*dict*) –

If the event is of type `RequestCancelExternalWorkflowExecutionInitiated` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowId** (*string*) –

The `workflowId` of the external workflow execution to be canceled.

- **runId** (*string*) –

The `runId` of the external workflow execution to be canceled.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the

`RequestCancelExternalWorkflowExecution` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **control** (*string*) –

Optional. Data attached to the event that can be used by the decider in subsequent workflow tasks.

- **requestCancelExternalWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `RequestCancelExternalWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowId** (*string*) –

The `workflowId` of the external workflow to which the cancel request was to be delivered.

- **runId** (*string*) –

The `runId` of the external workflow execution.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **initiatedEventId** (*integer*) –

The ID of the `RequestCancelExternalWorkflowExecutionInitiated` event corresponding to the `RequestCancelExternalWorkflowExecution` decision to cancel this external workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RequestCancelExternalWorkflowExecution` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **control** (*string*) –

- **scheduleActivityTaskFailedEventAttributes** (*dict*) –

If the event is of type `ScheduleActivityTaskFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **activityType** (*dict*) –

The activity type provided in the `ScheduleActivityTask` decision that failed.

- **name** (*string*) –

The name of this activity.

Note: The combination of activity type name and version must be unique within a domain.

- **version** (*string*) –

The version of this activity.

Note: The combination of activity type name and version must be unique with in a domain.

- **activityId** (*string*) –

The activityId provided in the `ScheduleActivityTask` decision that failed.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision that resulted in the scheduling of this activity task. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **requestCancelActivityTaskFailedEventAttributes** (*dict*) –

If the event is of type `RequestCancelActivityTaskFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **activityId** (*string*) –

The activityId provided in the `RequestCancelActivityTask` decision that failed.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RequestCancelActivityTask` decision for this cancellation request. This information can be useful for diag-

nosong problems by tracing back the chain of events leading up to this event.

- **startTimerFailedEventAttributes** (*dict*) –

If the event is of type `StartTimerFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timerId** (*string*) –

The `timerId` provided in the `StartTimer` decision that failed.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `StartTimer` decision for this activity task. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **cancelTimerFailedEventAttributes** (*dict*) –

If the event is of type `CancelTimerFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timerId** (*string*) –

The `timerId` provided in the `CancelTimer` decision that failed.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CancelTimer` decision to cancel this timer. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startChildWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `StartChildWorkflowExecutionFailed` then

this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowType** (*dict*) –

The workflow type provided in the StartChildWorkflowExecution decision that failed.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to OPERATION_NOT_PERMITTED, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **workflowId** (*string*) –

The workflowId of the child workflow execution.

- **initiatedEventId** (*integer*) –

The ID of the StartChildWorkflowExecutionInitiated event corresponding to the StartChildWorkflowExecution decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the DecisionTaskCompleted event corresponding to the decision task that resulted in the StartChildWorkflowExecution decision to request this child workflow execution. This information can be useful for diagnosing problems by tracing back the cause of events.

- **control** (*string*) –

- **lambdaFunctionScheduledEventAttributes** (*dict*) –

Provides details for the LambdaFunctionScheduled event.

- **id** (*string*) –

The unique Amazon SWF ID for the AWS Lambda task.

- **name** (*string*) –

The name of the scheduled AWS Lambda function.

- **input** (*string*) –
Input provided to the AWS Lambda function.
- **startToCloseTimeout** (*string*) –
The maximum time, in seconds, that the AWS Lambda function can take to execute from start to close before it is marked as failed.
- **decisionTaskCompletedEventId** (*integer*) –
The ID of the `DecisionTaskCompleted` event for the decision that resulted in the scheduling of this AWS Lambda function. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.
- **lambdaFunctionStartedEventAttributes** (*dict*) –
Provides details for the `LambdaFunctionStarted` event.
- **scheduledEventId** (*integer*) –
The ID of the `LambdaFunctionScheduled` event that was recorded when this AWS Lambda function was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.
- **lambdaFunctionCompletedEventAttributes** (*dict*) –
Provides details for the `LambdaFunctionCompleted` event.
- **scheduledEventId** (*integer*) –
The ID of the `LambdaFunctionScheduled` event that was recorded when this AWS Lambda function was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.
- **startedEventId** (*integer*) –
The ID of the `LambdaFunctionStarted` event recorded in the history.
- **result** (*string*) –
The result of the function execution (if any).
- **lambdaFunctionFailedEventAttributes** (*dict*) –
Provides details for the `LambdaFunctionFailed` event.
- **scheduledEventId** (*integer*) –
The ID of the `LambdaFunctionScheduled` event that was recorded when this AWS Lambda function was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.
- **startedEventId** (*integer*) –
The ID of the `LambdaFunctionStarted` event recorded in the history.
- **reason** (*string*) –
The reason provided for the failure (if any).
- **details** (*string*) –
The details of the failure (if any).

- **lambdaFunctionTimedOutEventAttributes** (*dict*) –
Provides details for the `LambdaFunctionTimedOut` event.
- **scheduledEventId** (*integer*) –
The ID of the `LambdaFunctionScheduled` event that was recorded when this AWS Lambda function was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.
- **startedEventId** (*integer*) –
The ID of the `LambdaFunctionStarted` event recorded in the history.
- **timeoutType** (*string*) –
The type of the timeout that caused this event.
- **scheduleLambdaFunctionFailedEventAttributes** (*dict*) –
Provides details for the `ScheduleLambdaFunctionFailed` event.
- **id** (*string*) –
The unique Amazon SWF ID of the AWS Lambda task.
- **name** (*string*) –
The name of the scheduled AWS Lambda function.
- **cause** (*string*) –
The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –
The ID of the `DecisionTaskCompleted` event corresponding to the decision that resulted in the scheduling of this AWS Lambda function. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.
- **startLambdaFunctionFailedEventAttributes** (*dict*) –
Provides details for the `StartLambdaFunctionFailed` event.
- **scheduledEventId** (*integer*) –
The ID of the `LambdaFunctionScheduled` event that was recorded when this AWS Lambda function was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.
- **cause** (*string*) –
The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`,

the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **message** (*string*) –

The error message (if any).

- **NextToken** (*string*) –

A token to resume pagination.

class `SWF.Paginator.ListActivityTypes`

```
paginator = client.get_paginator('list_activity_types')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `SWF.Client.list_activity_types()`.

Request Syntax

```
response_iterator = paginator.paginate(
    domain='string',
    name='string',
    registrationStatus='REGISTERED' | 'DEPRECATED',
    reverseOrder=True|False,
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **domain** (*string*) – **[REQUIRED]**

The name of the domain in which the activity types have been registered.

- **name** (*string*) – If specified, only lists the activity types that have this name.

- **registrationStatus** (*string*) – **[REQUIRED]**

Specifies the registration status of the activity types to list.

- **reverseOrder** (*boolean*) – When set to `true`, returns the results in reverse order. By default, the results are returned in ascending alphabetical order by name of the activity types.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
  'typeInfos': [
    {
      'activityType': {
        'name': 'string',
        'version': 'string'
      },
      'status': 'REGISTERED'|'DEPRECATED',
      'description': 'string',
      'creationDate': datetime(2015, 1, 1),
      'deprecationDate': datetime(2015, 1, 1)
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains a paginated list of activity type information structures.

- **typeInfos** (*list*) –

List of activity type information.

- * (*dict*) –

Detailed information about an activity type.

- **activityType** (*dict*) –

The ActivityType type structure representing the activity type.

- **name** (*string*) –

The name of this activity.

Note: The combination of activity type name and version must be unique within a domain.

- **version** (*string*) –

The version of this activity.

Note: The combination of activity type name and version must be unique with in a domain.

- **status** (*string*) –

The current status of the activity type.

- **description** (*string*) –

The description of the activity type provided in RegisterActivityType .

- **creationDate** (*datetime*) –

The date and time this activity type was created through RegisterActivityType .

- **deprecationDate** (*datetime*) –

If DEPRECATED, the date and time DeprecateActivityType was called.

– **NextToken** (*string*) –

A token to resume pagination.

class `SWF.Paginator.ListClosedWorkflowExecutions`

```
paginator = client.get_paginator('list_closed_workflow_executions')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `SWF.Client.list_closed_workflow_executions()`.

Request Syntax

```
response_iterator = paginator.paginate(
    domain='string',
    startTimeFilter={
        'oldestDate': datetime(2015, 1, 1),
        'latestDate': datetime(2015, 1, 1)
    },
    closeTimeFilter={
        'oldestDate': datetime(2015, 1, 1),
        'latestDate': datetime(2015, 1, 1)
    },
    executionFilter={
        'workflowId': 'string'
    },
    closeStatusFilter={
        'status': 'COMPLETED' | 'FAILED' | 'CANCELED' | 'TERMINATED' | 'CONTINUED_AS_NEW' | 'TIMED_OUT'
    },
    typeFilter={
        'name': 'string',
        'version': 'string'
    },
    tagFilter={
        'tag': 'string'
    },
    reverseOrder=True|False,
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **domain** (*string*) – [REQUIRED]

The name of the domain that contains the workflow executions to list.

- **startTimeFilter** (*dict*) – If specified, the workflow executions are included in the returned results based on whether their start times are within the range specified by this filter. Also, if this parameter is specified, the returned results are ordered by their start times.

Note: `startTimeFilter` and `closeTimeFilter` are mutually exclusive. You must specify one of these in a request but not both.

– **oldestDate** (*datetime*) – [REQUIRED]

Specifies the oldest start or close date and time to return.

– **latestDate** (*datetime*) –

Specifies the latest start or close date and time to return.

- **closeTimeFilter** (*dict*) – If specified, the workflow executions are included in the returned results based on whether their close times are within the range specified by this filter. Also, if this parameter is specified, the returned results are ordered by their close times.

Note: `startTimeFilter` and `closeTimeFilter` are mutually exclusive. You must specify one of these in a request but not both.

– **oldestDate** (*datetime*) – [REQUIRED]

Specifies the oldest start or close date and time to return.

– **latestDate** (*datetime*) –

Specifies the latest start or close date and time to return.

- **executionFilter** (*dict*) – If specified, only workflow executions matching the workflow ID specified in the filter are returned.

Note: `closeStatusFilter`, `executionFilter`, `typeFilter` and `tagFilter` are mutually exclusive. You can specify at most one of these in a request.

– **workflowId** (*string*) – [REQUIRED]

The workflowId to pass of match the criteria of this filter.

- **closeStatusFilter** (*dict*) – If specified, only workflow executions that match this *close status* are listed. For example, if TERMINATED is specified, then only TERMINATED workflow executions are listed.

Note: `closeStatusFilter`, `executionFilter`, `typeFilter` and `tagFilter` are mutually exclusive. You can specify at most one of these in a request.

– **status** (*string*) – [REQUIRED]

Required. The close status that must match the close status of an execution for it to meet the criteria of this filter.

- **typeFilter** (*dict*) – If specified, only executions of the type specified in the filter are returned.

Note: `closeStatusFilter`, `executionFilter`, `typeFilter` and `tagFilter` are mutually exclusive. You can specify at most one of these in a request.

– **name** (*string*) – [REQUIRED]

Required. Name of the workflow type.

– **version** (*string*) –

Version of the workflow type.

- **tagFilter** (*dict*) – If specified, only executions that have the matching tag are listed.

Note: `closeStatusFilter`, `executionFilter`, `typeFilter` and `tagFilter` are mutually exclusive. You can specify at most one of these in a request.

- **tag** (*string*) – **[REQUIRED]**

Required. Specifies the tag that must be associated with the execution for it to meet the filter criteria.

- **reverseOrder** (*boolean*) – When set to `true`, returns the results in reverse order. By default the results are returned in descending order of the start or the close time of the executions.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
  'executionInfos': [
    {
      'execution': {
        'workflowId': 'string',
        'runId': 'string'
      },
      'workflowType': {
        'name': 'string',
        'version': 'string'
      },
      'startTimestamp': datetime(2015, 1, 1),
      'closeTimestamp': datetime(2015, 1, 1),
      'executionStatus': 'OPEN'|'CLOSED',
      'closeStatus': 'COMPLETED'|'FAILED'|'CANCELED'|'TERMINATED'|'CONTINUED',
      'parent': {
        'workflowId': 'string',
        'runId': 'string'
      },
      'tagList': [
        'string',
      ],
      'cancelRequested': True|False
    },
  ],
}
```

```

    'NextToken': 'string'
}

```

Response Structure

- (*dict*) –

Contains a paginated list of information about workflow executions.

- **executionInfos** (*list*) –

The list of workflow information structures.

- * (*dict*) –

Contains information about a workflow execution.

- **execution** (*dict*) –

The workflow execution this information is about.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **startTimestamp** (*datetime*) –

The time when the execution was started.

- **closeTimestamp** (*datetime*) –

The time when the workflow execution was closed. Set only if the execution status is CLOSED.

- **executionStatus** (*string*) –

The current status of the execution.

- **closeStatus** (*string*) –

If the execution status is closed then this specifies how the execution was closed:

- **COMPLETED** : the execution was successfully completed.
- **CANCELED** : the execution was canceled. Cancellation allows the implementation to gracefully clean up before the execution is closed.
- **TERMINATED** : the execution was force terminated.
- **FAILED** : the execution failed to complete.

- **TIMED_OUT** : the execution did not complete in the allotted time and was automatically timed out.
- **CONTINUED_AS_NEW** : the execution is logically continued. This means the current execution was completed and a new execution was started to carry on the workflow.

- **parent** (*dict*) –

If this workflow execution is a child of another execution then contains the workflow execution that started this execution.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **tagList** (*list*) –

The list of tags associated with the workflow execution. Tags can be used to identify and list workflow executions of interest through the visibility APIs. A workflow execution can have a maximum of 5 tags.

- (*string*) –

- **cancelRequested** (*boolean*) –

Set to true if a cancellation is requested for this workflow execution.

- **NextToken** (*string*) –

A token to resume pagination.

class `SWF.Paginator.ListDomains`

```
paginator = client.get_paginator('list_domains')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `SWF.Client.list_domains()`.

Request Syntax

```
response_iterator = paginator.paginate(
    registrationStatus='REGISTERED'|'DEPRECATED',
    reverseOrder=True|False,
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **registrationStatus** (*string*) – **[REQUIRED]**

Specifies the registration status of the domains to list.

- **reverseOrder** (*boolean*) – When set to `true`, returns the results in reverse order. By default, the results are returned in ascending alphabetical order by name of the domains.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.
 - **MaxItems** (*integer*) –
The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.
 - **PageSize** (*integer*) –
The size of each page.
 - **StartingToken** (*string*) –
A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
  'domainInfos': [
    {
      'name': 'string',
      'status': 'REGISTERED'|'DEPRECATED',
      'description': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –
Contains a paginated collection of `DomainInfo` structures.
 - **domainInfos** (*list*) –
A list of `DomainInfo` structures.
 - * (*dict*) –
Contains general information about a domain.
 - **name** (*string*) –
The name of the domain. This name is unique within the account.
 - **status** (*string*) –
The status of the domain:
 - **REGISTERED** : The domain is properly registered and available. You can use this domain for registering types and creating new workflow executions.
 - **DEPRECATED** : The domain was deprecated using `DeprecateDomain` , but is still in use. You should not create new workflow executions in this domain.
 - **description** (*string*) –
The description of the domain provided through `RegisterDomain` .
 - **NextToken** (*string*) –
A token to resume pagination.

class `SWF.Paginator.ListOpenWorkflowExecutions`

```
paginator = client.get_paginator('list_open_workflow_executions')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `SWF.Client.list_open_workflow_executions()`.

Request Syntax

```
response_iterator = paginator.paginate(
    domain='string',
    startTimeFilter={
        'oldestDate': datetime(2015, 1, 1),
        'latestDate': datetime(2015, 1, 1)
    },
    typeFilter={
        'name': 'string',
        'version': 'string'
    },
    tagFilter={
        'tag': 'string'
    },
    reverseOrder=True|False,
    executionFilter={
        'workflowId': 'string'
    },
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

Parameters

- **domain** (*string*) – [REQUIRED]

The name of the domain that contains the workflow executions to list.

- **startTimeFilter** (*dict*) – [REQUIRED]

Workflow executions are included in the returned results based on whether their start times are within the range specified by this filter.

- **oldestDate** (*datetime*) – [REQUIRED]

Specifies the oldest start or close date and time to return.

- **latestDate** (*datetime*) –

Specifies the latest start or close date and time to return.

- **typeFilter** (*dict*) – If specified, only executions of the type specified in the filter are returned.

Note: `executionFilter`, `typeFilter` and `tagFilter` are mutually exclusive. You can specify at most one of these in a request.

- **name** (*string*) – [REQUIRED]

Required. Name of the workflow type.

– **version** (*string*) –

Version of the workflow type.

- **tagFilter** (*dict*) – If specified, only executions that have the matching tag are listed.

Note: `executionFilter`, `typeFilter` and `tagFilter` are mutually exclusive. You can specify at most one of these in a request.

– **tag** (*string*) – **[REQUIRED]**

Required. Specifies the tag that must be associated with the execution for it to meet the filter criteria.

- **reverseOrder** (*boolean*) – When set to `true`, returns the results in reverse order. By default the results are returned in descending order of the start time of the executions.
- **executionFilter** (*dict*) – If specified, only workflow executions matching the workflow ID specified in the filter are returned.

Note: `executionFilter`, `typeFilter` and `tagFilter` are mutually exclusive. You can specify at most one of these in a request.

– **workflowId** (*string*) – **[REQUIRED]**

The workflowId to pass of match the criteria of this filter.

- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

– **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

– **PageSize** (*integer*) –

The size of each page.

– **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type `dict`

Returns

Response Syntax

```
{
  'executionInfos': [
    {
      'execution': {
        'workflowId': 'string',
        'runId': 'string'
      },
      'workflowType': {
        'name': 'string',
        'version': 'string'
      },
      'startTimestamp': datetime(2015, 1, 1),
      'closeTimestamp': datetime(2015, 1, 1),
      'executionStatus': 'OPEN'|'CLOSED',
    },
  ],
}
```

```
    'closeStatus': 'COMPLETED' | 'FAILED' | 'CANCELED' | 'TERMINATED' | 'CONTINUED',
    'parent': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'tagList': [
        'string',
    ],
    'cancelRequested': True|False
},
],
'NextToken': 'string'
}
```

Response Structure

- *(dict)* –

Contains a paginated list of information about workflow executions.

- **executionInfos** (*list*) –

The list of workflow information structures.

- * *(dict)* –

Contains information about a workflow execution.

- **execution** (*dict*) –

The workflow execution this information is about.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **startTimestamp** (*datetime*) –

The time when the execution was started.

- **closeTimestamp** (*datetime*) –

The time when the workflow execution was closed. Set only if the execution status is CLOSED.

- **executionStatus** (*string*) –

The current status of the execution.

- **closeStatus** (*string*) –

If the execution status is closed then this specifies how the execution was closed:

- **COMPLETED** : the execution was successfully completed.
- **CANCELED** : the execution was canceled. Cancellation allows the implementation to gracefully clean up before the execution is closed.
- **TERMINATED** : the execution was force terminated.
- **FAILED** : the execution failed to complete.
- **TIMED_OUT** : the execution did not complete in the allotted time and was automatically timed out.
- **CONTINUED_AS_NEW** : the execution is logically continued. This means the current execution was completed and a new execution was started to carry on the workflow.

- **parent** (*dict*) –

If this workflow execution is a child of another execution then contains the workflow execution that started this execution.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **tagList** (*list*) –

The list of tags associated with the workflow execution. Tags can be used to identify and list workflow executions of interest through the visibility APIs. A workflow execution can have a maximum of 5 tags.

- (*string*) –
- **cancelRequested** (*boolean*) –

Set to true if a cancellation is requested for this workflow execution.

- **NextToken** (*string*) –

A token to resume pagination.

class SWF.Paginator.**ListWorkflowTypes**

```
paginator = client.get_paginator('list_workflow_types')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `SWF.Client.list_workflow_types()`.

Request Syntax

```
response_iterator = paginator.paginate(
    domain='string',
    name='string',
    registrationStatus='REGISTERED' | 'DEPRECATED',
    reverseOrder=True | False,
```

```
PaginationConfig={
    'MaxItems': 123,
    'PageSize': 123,
    'StartingToken': 'string'
}
)
```

Parameters

- **domain** (*string*) – [REQUIRED]

The name of the domain in which the workflow types have been registered.

- **name** (*string*) – If specified, lists the workflow type with this name.
- **registrationStatus** (*string*) – [REQUIRED]

Specifies the registration status of the workflow types to list.

- **reverseOrder** (*boolean*) – When set to `true`, returns the results in reverse order. By default the results are returned in ascending alphabetical order of the name of the workflow types.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in max-items then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{
  'typeInfos': [
    {
      'workflowType': {
        'name': 'string',
        'version': 'string'
      },
      'status': 'REGISTERED'|'DEPRECATED',
      'description': 'string',
      'creationDate': datetime(2015, 1, 1),
      'deprecationDate': datetime(2015, 1, 1)
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains a paginated list of information structures about workflow types.

- **typeInfos** (*list*) –

The list of workflow type information.

* (*dict*) –

Contains information about a workflow type.

- **workflowType** (*dict*) –

The workflow type this information is about.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **status** (*string*) –

The current status of the workflow type.

- **description** (*string*) –

The description of the type registered through RegisterWorkflowType .

- **creationDate** (*datetime*) –

The date when this type was registered.

- **deprecationDate** (*datetime*) –

If the type is in deprecated state, then it is set to the date when the type was deprecated.

- **NextToken** (*string*) –

A token to resume pagination.

class `SWF.Paginator.PollForDecisionTask`

```
paginator = client.get_paginator('poll_for_decision_task')
```

paginate (***kwargs*)

Creates an iterator that will paginate through responses from `SWF.Client.poll_for_decision_task()`.

Request Syntax

```
response_iterator = paginator.paginate(
    domain='string',
    taskList={
        'name': 'string'
    },
    identity='string',
    reverseOrder=True|False,
    PaginationConfig={
        'MaxItems': 123,
        'PageSize': 123,
        'StartingToken': 'string'
    }
)
```

```
}  
)
```

Parameters

- **domain** (*string*) – [REQUIRED]

The name of the domain containing the task lists to poll.

- **taskList** (*dict*) – [REQUIRED]

Specifies the task list to poll for decision tasks.

The specified string must not start or end with whitespace. It must not contain a : (colon), / (slash), | (vertical bar), or any control characters (u0000-u001f | u007f - u009f). Also, it must not contain the literal string quotarnquot.

- **name** (*string*) – [REQUIRED]

The name of the task list.

- **identity** (*string*) – Identity of the decider making the request, which is recorded in the DecisionTaskStarted event in the workflow history. This enables diagnostic tracing when problems arise. The form of this identity is user defined.
- **reverseOrder** (*boolean*) – When set to `true`, returns the events in reverse order. By default the results are returned in ascending order of the `eventTimestamp` of the events.
- **PaginationConfig** (*dict*) – A dictionary that provides parameters to control pagination.

- **MaxItems** (*integer*) –

The total number of items to return. If the total number of items available is more than the value specified in `max-items` then a `NextToken` will be provided in the output that you can use to resume pagination.

- **PageSize** (*integer*) –

The size of each page.

- **StartingToken** (*string*) –

A token to specify where to start paginating. This is the `NextToken` from a previous response.

Return type dict

Returns

Response Syntax

```
{  
  'taskToken': 'string',  
  'startedEventId': 123,  
  'workflowExecution': {  
    'workflowId': 'string',  
    'runId': 'string'  
  },  
  'workflowType': {  
    'name': 'string',  
    'version': 'string'  
  },  
  'events': [  
    {  
      'eventTimestamp': datetime(2015, 1, 1),  
      'eventType': 'WorkflowExecutionStarted'|'WorkflowExecutionCancelRequest',  
      'eventId': 123,  
      'workflowExecutionStartedEventAttributes': {  
        'startTime': datetime(2015, 1, 1),  
        'endTime': datetime(2015, 1, 1),  
        'eventAttributes': {}  
      }  
    }  
  ]  
}
```

```

        'input': 'string',
        'executionStartToCloseTimeout': 'string',
        'taskStartToCloseTimeout': 'string',
        'childPolicy': 'TERMINATE'|'REQUEST_CANCEL'|'ABANDON',
        'taskList': {
            'name': 'string'
        },
        'workflowType': {
            'name': 'string',
            'version': 'string'
        },
        'tagList': [
            'string',
        ],
        'taskPriority': 'string',
        'continuedExecutionRunId': 'string',
        'parentWorkflowExecution': {
            'workflowId': 'string',
            'runId': 'string'
        },
        'parentInitiatedEventId': 123,
        'lambdaRole': 'string'
    },
    'workflowExecutionCompletedEventAttributes': {
        'result': 'string',
        'decisionTaskCompletedEventId': 123
    },
    'completeWorkflowExecutionFailedEventAttributes': {
        'cause': 'UNHANDLED_DECISION'|'OPERATION_NOT_PERMITTED',
        'decisionTaskCompletedEventId': 123
    },
    'workflowExecutionFailedEventAttributes': {
        'reason': 'string',
        'details': 'string',
        'decisionTaskCompletedEventId': 123
    },
    'failWorkflowExecutionFailedEventAttributes': {
        'cause': 'UNHANDLED_DECISION'|'OPERATION_NOT_PERMITTED',
        'decisionTaskCompletedEventId': 123
    },
    'workflowExecutionTimedOutEventAttributes': {
        'timeoutType': 'START_TO_CLOSE',
        'childPolicy': 'TERMINATE'|'REQUEST_CANCEL'|'ABANDON'
    },
    'workflowExecutionCanceledEventAttributes': {
        'details': 'string',
        'decisionTaskCompletedEventId': 123
    },
    'cancelWorkflowExecutionFailedEventAttributes': {
        'cause': 'UNHANDLED_DECISION'|'OPERATION_NOT_PERMITTED',
        'decisionTaskCompletedEventId': 123
    },
    'workflowExecutionContinuedAsNewEventAttributes': {
        'input': 'string',
        'decisionTaskCompletedEventId': 123,
        'newExecutionRunId': 'string',
        'executionStartToCloseTimeout': 'string',
        'taskList': {

```

```
        'name': 'string'
    },
    'taskPriority': 'string',
    'taskStartToCloseTimeout': 'string',
    'childPolicy': 'TERMINATE'|'REQUEST_CANCEL'|'ABANDON',
    'tagList': [
        'string',
    ],
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'lambdaRole': 'string'
},
'continueAsNewWorkflowExecutionFailedEventAttributes': {
    'cause': 'UNHANDLED_DECISION'|'WORKFLOW_TYPE_DEPRECATED'|'WORKFLOW_
    'decisionTaskCompletedEventId': 123
},
'workflowExecutionTerminatedEventAttributes': {
    'reason': 'string',
    'details': 'string',
    'childPolicy': 'TERMINATE'|'REQUEST_CANCEL'|'ABANDON',
    'cause': 'CHILD_POLICY_APPLIED'|'EVENT_LIMIT_EXCEEDED'|'OPERATOR_I
},
'workflowExecutionCancelRequestedEventAttributes': {
    'externalWorkflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'externalInitiatedEventId': 123,
    'cause': 'CHILD_POLICY_APPLIED'
},
'decisionTaskScheduledEventAttributes': {
    'taskList': {
        'name': 'string'
    },
    'taskPriority': 'string',
    'startToCloseTimeout': 'string'
},
'decisionTaskStartedEventAttributes': {
    'identity': 'string',
    'scheduledEventId': 123
},
'decisionTaskCompletedEventAttributes': {
    'executionContext': 'string',
    'scheduledEventId': 123,
    'startedEventId': 123
},
'decisionTaskTimedOutEventAttributes': {
    'timeoutType': 'START_TO_CLOSE',
    'scheduledEventId': 123,
    'startedEventId': 123
},
'activityTaskScheduledEventAttributes': {
    'activityType': {
        'name': 'string',
        'version': 'string'
    },
},
```

```

        'activityId': 'string',
        'input': 'string',
        'control': 'string',
        'scheduleToStartTimeout': 'string',
        'scheduleToCloseTimeout': 'string',
        'startToCloseTimeout': 'string',
        'taskList': {
            'name': 'string'
        },
        'taskPriority': 'string',
        'decisionTaskCompletedEventId': 123,
        'heartbeatTimeout': 'string'
    },
    'activityTaskStartedEventAttributes': {
        'identity': 'string',
        'scheduledEventId': 123
    },
    'activityTaskCompletedEventAttributes': {
        'result': 'string',
        'scheduledEventId': 123,
        'startedEventId': 123
    },
    'activityTaskFailedEventAttributes': {
        'reason': 'string',
        'details': 'string',
        'scheduledEventId': 123,
        'startedEventId': 123
    },
    'activityTaskTimedOutEventAttributes': {
        'timeoutType': 'START_TO_CLOSE'|'SCHEDULE_TO_START'|'SCHEDULE_TO_C
        'scheduledEventId': 123,
        'startedEventId': 123,
        'details': 'string'
    },
    'activityTaskCanceledEventAttributes': {
        'details': 'string',
        'scheduledEventId': 123,
        'startedEventId': 123,
        'latestCancelRequestedEventId': 123
    },
    'activityTaskCancelRequestedEventAttributes': {
        'decisionTaskCompletedEventId': 123,
        'activityId': 'string'
    },
    'workflowExecutionSignaledEventAttributes': {
        'signalName': 'string',
        'input': 'string',
        'externalWorkflowExecution': {
            'workflowId': 'string',
            'runId': 'string'
        },
        'externalInitiatedEventId': 123
    },
    'markerRecordedEventAttributes': {
        'markerName': 'string',
        'details': 'string',
        'decisionTaskCompletedEventId': 123
    },

```

```
'recordMarkerFailedEventAttributes': {
    'markerName': 'string',
    'cause': 'OPERATION_NOT_PERMITTED',
    'decisionTaskCompletedEventId': 123
},
'timerStartedEventAttributes': {
    'timerId': 'string',
    'control': 'string',
    'startToFireTimeout': 'string',
    'decisionTaskCompletedEventId': 123
},
'timerFiredEventAttributes': {
    'timerId': 'string',
    'startedEventId': 123
},
'timerCanceledEventAttributes': {
    'timerId': 'string',
    'startedEventId': 123,
    'decisionTaskCompletedEventId': 123
},
'startChildWorkflowExecutionInitiatedEventAttributes': {
    'workflowId': 'string',
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'control': 'string',
    'input': 'string',
    'executionStartToCloseTimeout': 'string',
    'taskList': {
        'name': 'string'
    },
    'taskPriority': 'string',
    'decisionTaskCompletedEventId': 123,
    'childPolicy': 'TERMINATE'|'REQUEST_CANCEL'|'ABANDON',
    'taskStartToCloseTimeout': 'string',
    'tagList': [
        'string',
    ],
    'lambdaRole': 'string'
},
'childWorkflowExecutionStartedEventAttributes': {
    'workflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'initiatedEventId': 123
},
'childWorkflowExecutionCompletedEventAttributes': {
    'workflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'workflowType': {
```



```

        'name': 'string',
        'version': 'string'
    },
    'result': 'string',
    'initiatedEventId': 123,
    'startedEventId': 123
},
'childWorkflowExecutionFailedEventAttributes': {
    'workflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'reason': 'string',
    'details': 'string',
    'initiatedEventId': 123,
    'startedEventId': 123
},
'childWorkflowExecutionTimedOutEventAttributes': {
    'workflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'timeoutType': 'START_TO_CLOSE',
    'initiatedEventId': 123,
    'startedEventId': 123
},
'childWorkflowExecutionCanceledEventAttributes': {
    'workflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'details': 'string',
    'initiatedEventId': 123,
    'startedEventId': 123
},
'childWorkflowExecutionTerminatedEventAttributes': {
    'workflowExecution': {
        'workflowId': 'string',
        'runId': 'string'
    },
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'initiatedEventId': 123,
    'startedEventId': 123
}

```

```
    },
    'signalExternalWorkflowExecutionInitiatedEventAttributes': {
        'workflowId': 'string',
        'runId': 'string',
        'signalName': 'string',
        'input': 'string',
        'decisionTaskCompletedEventId': 123,
        'control': 'string'
    },
    'externalWorkflowExecutionSignaledEventAttributes': {
        'workflowExecution': {
            'workflowId': 'string',
            'runId': 'string'
        },
        'initiatedEventId': 123
    },
    'signalExternalWorkflowExecutionFailedEventAttributes': {
        'workflowId': 'string',
        'runId': 'string',
        'cause': 'UNKNOWN_EXTERNAL_WORKFLOW_EXECUTION' | 'SIGNAL_EXTERNAL_WO',
        'initiatedEventId': 123,
        'decisionTaskCompletedEventId': 123,
        'control': 'string'
    },
    'externalWorkflowExecutionCancelRequestedEventAttributes': {
        'workflowExecution': {
            'workflowId': 'string',
            'runId': 'string'
        },
        'initiatedEventId': 123
    },
    'requestCancelExternalWorkflowExecutionInitiatedEventAttributes': {
        'workflowId': 'string',
        'runId': 'string',
        'decisionTaskCompletedEventId': 123,
        'control': 'string'
    },
    'requestCancelExternalWorkflowExecutionFailedEventAttributes': {
        'workflowId': 'string',
        'runId': 'string',
        'cause': 'UNKNOWN_EXTERNAL_WORKFLOW_EXECUTION' | 'REQUEST_CANCEL_EXT',
        'initiatedEventId': 123,
        'decisionTaskCompletedEventId': 123,
        'control': 'string'
    },
    'scheduleActivityTaskFailedEventAttributes': {
        'activityType': {
            'name': 'string',
            'version': 'string'
        },
        'activityId': 'string',
        'cause': 'ACTIVITY_TYPE_DEPRECATED' | 'ACTIVITY_TYPE_DOES_NOT_EXIST',
        'decisionTaskCompletedEventId': 123
    },
    'requestCancelActivityTaskFailedEventAttributes': {
        'activityId': 'string',
        'cause': 'ACTIVITY_ID_UNKNOWN' | 'OPERATION_NOT_PERMITTED',
        'decisionTaskCompletedEventId': 123
    }
```

```

},
'startTimerFailedEventAttributes': {
    'timerId': 'string',
    'cause': 'TIMER_ID_ALREADY_IN_USE' | 'OPEN_TIMERS_LIMIT_EXCEEDED' | 'T
    'decisionTaskCompletedEventId': 123
},
'cancelTimerFailedEventAttributes': {
    'timerId': 'string',
    'cause': 'TIMER_ID_UNKNOWN' | 'OPERATION_NOT_PERMITTED',
    'decisionTaskCompletedEventId': 123
},
'startChildWorkflowExecutionFailedEventAttributes': {
    'workflowType': {
        'name': 'string',
        'version': 'string'
    },
    'cause': 'WORKFLOW_TYPE_DOES_NOT_EXIST' | 'WORKFLOW_TYPE_DEPRECATED',
    'workflowId': 'string',
    'initiatedEventId': 123,
    'decisionTaskCompletedEventId': 123,
    'control': 'string'
},
'lambdaFunctionScheduledEventAttributes': {
    'id': 'string',
    'name': 'string',
    'input': 'string',
    'startToCloseTimeout': 'string',
    'decisionTaskCompletedEventId': 123
},
'lambdaFunctionStartedEventAttributes': {
    'scheduledEventId': 123
},
'lambdaFunctionCompletedEventAttributes': {
    'scheduledEventId': 123,
    'startedEventId': 123,
    'result': 'string'
},
'lambdaFunctionFailedEventAttributes': {
    'scheduledEventId': 123,
    'startedEventId': 123,
    'reason': 'string',
    'details': 'string'
},
'lambdaFunctionTimedOutEventAttributes': {
    'scheduledEventId': 123,
    'startedEventId': 123,
    'timeoutType': 'START_TO_CLOSE'
},
'scheduleLambdaFunctionFailedEventAttributes': {
    'id': 'string',
    'name': 'string',
    'cause': 'ID_ALREADY_IN_USE' | 'OPEN_LAMBDA_FUNCTIONS_LIMIT_EXCEEDED
    'decisionTaskCompletedEventId': 123
},
'startLambdaFunctionFailedEventAttributes': {
    'scheduledEventId': 123,
    'cause': 'ASSUME_ROLE_FAILED',
    'message': 'string'
}

```

```
    }
  },
],
'previousStartedEventId': 123,
'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

A structure that represents a decision task. Decision tasks are sent to deciders in order for them to make decisions.

- **taskToken** (*string*) –

The opaque string used as a handle on the task. This token is used by workers to communicate progress and response information back to the system about the task.

- **startedEventId** (*integer*) –

The ID of the `DecisionTaskStarted` event recorded in the history.

- **workflowExecution** (*dict*) –

The workflow execution for which this decision task was created.

- * **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- * **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the workflow execution for which this decision task was created.

- * **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique within a domain.

- * **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique within a domain.

- **events** (*list*) –

A paginated list of history events of the workflow execution. The decider uses this during the processing of the decision task.

- * (*dict*) –

Event within a workflow execution. A history event can be one of these types:

- **WorkflowExecutionStarted** : The workflow execution was started.
- **WorkflowExecutionCompleted** : The workflow execution was closed due to successful completion.

- **WorkflowExecutionFailed** : The workflow execution closed due to a failure.
- **WorkflowExecutionTimedOut** : The workflow execution was closed because a time out was exceeded.
- **WorkflowExecutionCanceled** : The workflow execution was successfully canceled and closed.
- **WorkflowExecutionTerminated** : The workflow execution was terminated.
- **WorkflowExecutionContinuedAsNew** : The workflow execution was closed and a new execution of the same type was created with the same workflowId.
- **WorkflowExecutionCancelRequested** : A request to cancel this workflow execution was made.
- **DecisionTaskScheduled** : A decision task was scheduled for the workflow execution.
- **DecisionTaskStarted** : The decision task was dispatched to a decider.
- **DecisionTaskCompleted** : The decider successfully completed a decision task by calling `RespondDecisionTaskCompleted`.
- **DecisionTaskTimedOut** : The decision task timed out.
- **ActivityTaskScheduled** : An activity task was scheduled for execution.
- **ScheduleActivityTaskFailed** : Failed to process `ScheduleActivityTask` decision. This happens when the decision is not configured properly, for example the activity type specified is not registered.
- **ActivityTaskStarted** : The scheduled activity task was dispatched to a worker.
- **ActivityTaskCompleted** : An activity worker successfully completed an activity task by calling `RespondActivityTaskCompleted`.
- **ActivityTaskFailed** : An activity worker failed an activity task by calling `RespondActivityTaskFailed`.
- **ActivityTaskTimedOut** : The activity task timed out.
- **ActivityTaskCanceled** : The activity task was successfully canceled.
- **ActivityTaskCancelRequested** : A `RequestCancelActivityTask` decision was received by the system.
- **RequestCancelActivityTaskFailed** : Failed to process `RequestCancelActivityTask` decision. This happens when the decision is not configured properly.
- **WorkflowExecutionSignaled** : An external signal was received for the workflow execution.
- **MarkerRecorded** : A marker was recorded in the workflow history as the result of a `RecordMarker` decision.
- **TimerStarted** : A timer was started for the workflow execution due to a `StartTimer` decision.
- **StartTimerFailed** : Failed to process `StartTimer` decision. This happens when the decision is not configured properly, for example a timer already exists with the specified timer ID.
- **TimerFired** : A timer, previously started for this workflow execution, fired.

- **TimerCanceled** : A timer, previously started for this workflow execution, was successfully canceled.
- **CancelTimerFailed** : Failed to process CancelTimer decision. This happens when the decision is not configured properly, for example no timer exists with the specified timer ID.
- **StartChildWorkflowExecutionInitiated** : A request was made to start a child workflow execution.
- **StartChildWorkflowExecutionFailed** : Failed to process StartChildWorkflowExecution decision. This happens when the decision is not configured properly, for example the workflow type specified is not registered.
- **ChildWorkflowExecutionStarted** : A child workflow execution was successfully started.
- **ChildWorkflowExecutionCompleted** : A child workflow execution, started by this workflow execution, completed successfully and was closed.
- **ChildWorkflowExecutionFailed** : A child workflow execution, started by this workflow execution, failed to complete successfully and was closed.
- **ChildWorkflowExecutionTimedOut** : A child workflow execution, started by this workflow execution, timed out and was closed.
- **ChildWorkflowExecutionCanceled** : A child workflow execution, started by this workflow execution, was canceled and closed.
- **ChildWorkflowExecutionTerminated** : A child workflow execution, started by this workflow execution, was terminated.
- **SignalExternalWorkflowExecutionInitiated** : A request to signal an external workflow was made.
- **ExternalWorkflowExecutionSignaled** : A signal, requested by this workflow execution, was successfully delivered to the target external workflow execution.
- **SignalExternalWorkflowExecutionFailed** : The request to signal an external workflow execution failed.
- **RequestCancelExternalWorkflowExecutionInitiated** : A request was made to request the cancellation of an external workflow execution.
- **ExternalWorkflowExecutionCancelRequested** : Request to cancel an external workflow execution was successfully delivered to the target execution.
- **RequestCancelExternalWorkflowExecutionFailed** : Request to cancel an external workflow execution failed.
- **LambdaFunctionScheduled** : An AWS Lambda function was scheduled for execution.
- **LambdaFunctionStarted** : The scheduled function was invoked in the AWS Lambda service.
- **LambdaFunctionCompleted** : The AWS Lambda function successfully completed.
- **LambdaFunctionFailed** : The AWS Lambda function execution failed.
- **LambdaFunctionTimedOut** : The AWS Lambda function execution timed out.
- **ScheduleLambdaFunctionFailed** : Failed to process Sched-

uleLambdaFunction decision. This happens when the workflow execution does not have the proper IAM role attached to invoke AWS Lambda functions.

- **StartLambdaFunctionFailed** : Failed to invoke the scheduled function in the AWS Lambda service. This happens when the AWS Lambda service is not available in the current region, or received too many requests.

- **eventTimestamp** (*datetime*) –

The date and time when the event occurred.

- **eventType** (*string*) –

The type of the history event.

- **eventId** (*integer*) –

The system generated ID of the event. This ID uniquely identifies the event within the workflow execution history.

- **workflowExecutionStartedEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionStarted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **input** (*string*) –

The input provided to the workflow execution (if any).

- **executionStartToCloseTimeout** (*string*) –

The maximum duration for this workflow execution.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **taskStartToCloseTimeout** (*string*) –

The maximum duration of decision tasks for this workflow type.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **childPolicy** (*string*) –

The policy to use for the child workflow executions if this workflow execution is terminated, by calling the `TerminateWorkflowExecution` action explicitly or due to an expired timeout.

The supported child policies are:

- **TERMINATE**: the child executions will be terminated.
- **REQUEST_CANCEL**: a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON**: no action will be taken. The child executions will continue to run.
- **taskList** (*dict*) –

The name of the task list for scheduling the decision tasks for this workflow execution.

- **name** (*string*) –

The name of the task list.

- **workflowType** (*dict*) –

The workflow type of this execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **tagList** (*list*) –

The list of tags associated with this workflow execution. An execution can have up to 5 tags.

- (*string*) –

- **taskPriority** (*string*) –

- **continuedExecutionRunId** (*string*) –

If this workflow execution was started due to a `ContinueAsNewWorkflowExecution` decision, then it contains the `runId` of the previous workflow execution that was closed and continued as this execution.

- **parentWorkflowExecution** (*dict*) –

The source workflow execution that started this workflow execution. The member is not set if the workflow execution was not started by a workflow.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **parentInitiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this workflow execution. The source event with this ID can be found in the history of the source workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **lambdaRole** (*string*) –

The IAM role attached to this workflow execution to use when invoking AWS Lambda functions.

- **workflowExecutionCompletedEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionCompleted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **result** (*string*) –

The result produced by the workflow execution upon successful completion.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CompleteWorkflowExecution` decision to complete this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **completeWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `CompleteWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CompleteWorkflowExecution` decision to complete this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **workflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **reason** (*string*) –

The descriptive reason provided for the failure (if any).

- **details** (*string*) –

The details of the failure (if any).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `FailWorkflowExecution` decision to fail this execution. This information can be useful for diagnosing problems

by tracing back the chain of events leading up to this event.

- **failWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `FailWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `FailWorkflowExecution` decision to fail this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **workflowExecutionTimedOutEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionTimedOut` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timeoutType** (*string*) –

The type of timeout that caused this event.

- **childPolicy** (*string*) –

The policy used for the child workflow executions of this workflow execution.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.
- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON:** no action will be taken. The child executions will continue to run.
- **workflowExecutionCanceledEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionCanceled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **details** (*string*) –

Details for the cancellation (if any).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CancelWorkflowExecution` decision for this cancella-

tion request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **cancelWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `CancelWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CancelWorkflowExecution` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **workflowExecutionContinuedAsNewEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionContinuedAsNew` then this member is set and provides detailed information about the event. It is not set for other event types.

- **input** (*string*) –

The input provided to the new workflow execution.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `ContinueAsNewWorkflowExecution` decision that started this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **newExecutionRunId** (*string*) –

The `runId` of the new workflow execution.

- **executionStartToCloseTimeout** (*string*) –

The total duration allowed for the new workflow execution.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **taskList** (*dict*) –

Represents a task list.

- **name** (*string*) –

The name of the task list.

- **taskPriority** (*string*) –
- **taskStartToCloseTimeout** (*string*) –

The maximum duration of decision tasks for the new workflow execution.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **childPolicy** (*string*) –

The policy to use for the child workflow executions of the new execution if it is terminated by calling the `TerminateWorkflowExecution` action explicitly or due to an expired timeout.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.
- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON:** no action will be taken. The child executions will continue to run.
- **tagList** (*list*) –

The list of tags associated with the new workflow execution.

- (*string*) –
- **workflowType** (*dict*) –

Represents a workflow type.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique within a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique within a domain.

- **lambdaRole** (*string*) –

The IAM role attached to this workflow execution to use when invoking AWS Lambda functions.

- **continueAsNewWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `ContinueAsNewWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `ContinueAsNewWorkflowExecution` decision that started this execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **workflowExecutionTerminatedEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionTerminated` then this member is set and provides detailed information about the event. It is not set for other event types.

- **reason** (*string*) –

The reason provided for the termination (if any).

- **details** (*string*) –

The details provided for the termination (if any).

- **childPolicy** (*string*) –

The policy used for the child workflow executions of this workflow execution.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.
- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON:** no action will be taken. The child executions will continue to run.
- **cause** (*string*) –

If set, indicates that the workflow execution was automatically terminated, and specifies the cause. This happens if the parent workflow execution times out or is terminated and the child policy is set to terminate child executions.

- **workflowExecutionCancelRequestedEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionCancelRequested` then this member is set and provides detailed information about the event. It is not set for other event types.

- **externalWorkflowExecution** (*dict*) –

The external workflow execution for which the cancellation was requested.

- **workflowId** (*string*) –
The user defined identifier associated with the workflow execution.
- **runId** (*string*) –
A system-generated unique identifier for the workflow execution.
- **externalInitiatedEventId** (*integer*) –
The ID of the `RequestCancelExternalWorkflowExecutionInitiated` event corresponding to the `RequestCancelExternalWorkflowExecution` decision to cancel this workflow execution. The source event with this ID can be found in the history of the source workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.
- **cause** (*string*) –
If set, indicates that the request to cancel the workflow execution was automatically generated, and specifies the cause. This happens if the parent workflow execution times out or is terminated, and the child policy is set to cancel child executions.
- **decisionTaskScheduledEventAttributes** (*dict*) –
If the event is of type `DecisionTaskScheduled` then this member is set and provides detailed information about the event. It is not set for other event types.
- **taskList** (*dict*) –
The name of the task list in which the decision task was scheduled.
- **name** (*string*) –
The name of the task list.
- **taskPriority** (*string*) –
Optional. A task priority that, if set, specifies the priority for this decision task. Valid values are integers that range from Java's `Integer.MIN_VALUE` (-2147483648) to `Integer.MAX_VALUE` (2147483647). Higher numbers indicate higher priority.

For more information about setting task priority, see [Setting Task Priority](#) in the *Amazon Simple Workflow Developer Guide*.
- **startToCloseTimeout** (*string*) –
The maximum duration for this decision task. The task is considered timed out if it does not completed within this duration.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.
- **decisionTaskStartedEventAttributes** (*dict*) –
If the event is of type `DecisionTaskStarted` then this member is set and provides detailed information about the

event. It is not set for other event types.

- **identity** (*string*) –

Identity of the decider making the request. This enables diagnostic tracing when problems arise. The form of this identity is user defined.

- **scheduledEventId** (*integer*) –

The ID of the `DecisionTaskScheduled` event that was recorded when this decision task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **decisionTaskCompletedEventAttributes** (*dict*) –

If the event is of type `DecisionTaskCompleted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **executionContext** (*string*) –

User defined context for the workflow execution.

- **scheduledEventId** (*integer*) –

The ID of the `DecisionTaskScheduled` event that was recorded when this decision task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `DecisionTaskStarted` event recorded when this decision task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **decisionTaskTimedOutEventAttributes** (*dict*) –

If the event is of type `DecisionTaskTimedOut` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timeoutType** (*string*) –

The type of timeout that expired before the decision task could be completed.

- **scheduledEventId** (*integer*) –

The ID of the `DecisionTaskScheduled` event that was recorded when this decision task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `DecisionTaskStarted` event recorded when this decision task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **activityTaskScheduledEventAttributes** (*dict*) –

If the event is of type `ActivityTaskScheduled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **activityType** (*dict*) –

The type of the activity task.

- **name** (*string*) –

The name of this activity.

Note: The combination of activity type name and version must be unique within a domain.

- **version** (*string*) –

The version of this activity.

Note: The combination of activity type name and version must be unique with in a domain.

- **activityId** (*string*) –

The unique ID of the activity task.

- **input** (*string*) –

The input provided to the activity task.

- **control** (*string*) –

Optional. Data attached to the event that can be used by the decider in subsequent workflow tasks. This data is not sent to the activity.

- **scheduleToStartTimeout** (*string*) –

The maximum amount of time the activity task can wait to be assigned to a worker.

- **scheduleToCloseTimeout** (*string*) –

The maximum amount of time for this activity task.

- **startToCloseTimeout** (*string*) –

The maximum amount of time a worker may take to process the activity task.

- **taskList** (*dict*) –

The task list in which the activity task has been scheduled.

- **name** (*string*) –

The name of the task list.

- **taskPriority** (*string*) –

Optional. The priority to assign to the scheduled activity task. If set, this will override any default priority value that was assigned when the activity type was registered.

Valid values are integers that range from Java's `Integer.MIN_VALUE` (-2147483648) to `Integer.MAX_VALUE` (2147483647). Higher numbers indicate higher priority.

For more information about setting task priority, see [Setting Task Priority](#) in the *Amazon Simple Workflow Developer Guide*.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision that resulted in the scheduling of this activity task. This information can be useful for diagnosing

problems by tracing back the chain of events leading up to this event.

- **heartbeatTimeout** (*string*) –

The maximum time before which the worker processing this task must report progress by calling `RecordActivityTaskHeartbeat`. If the timeout is exceeded, the activity task is automatically timed out. If the worker subsequently attempts to record a heartbeat or return a result, it will be ignored.

- **activityTaskStartedEventAttributes** (*dict*) –

If the event is of type `ActivityTaskStarted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **identity** (*string*) –

Identity of the worker that was assigned this task. This aids diagnostics when problems arise. The form of this identity is user defined.

- **scheduledEventId** (*integer*) –

The ID of the `ActivityTaskScheduled` event that was recorded when this activity task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **activityTaskCompletedEventAttributes** (*dict*) –

If the event is of type `ActivityTaskCompleted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **result** (*string*) –

The results of the activity task (if any).

- **scheduledEventId** (*integer*) –

The ID of the `ActivityTaskScheduled` event that was recorded when this activity task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ActivityTaskStarted` event recorded when this activity task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **activityTaskFailedEventAttributes** (*dict*) –

If the event is of type `ActivityTaskFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **reason** (*string*) –

The reason provided for the failure (if any).

- **details** (*string*) –

The details of the failure (if any).

- **scheduledEventId** (*integer*) –

The ID of the `ActivityTaskScheduled` event that was recorded when this activity task was scheduled. This infor-

mation can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ActivityTaskStarted` event recorded when this activity task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **activityTaskTimedOutEventAttributes** (*dict*) –

If the event is of type `ActivityTaskTimedOut` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timeoutType** (*string*) –

The type of the timeout that caused this event.

- **scheduledEventId** (*integer*) –

The ID of the `ActivityTaskScheduled` event that was recorded when this activity task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ActivityTaskStarted` event recorded when this activity task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **details** (*string*) –

Contains the content of the `details` parameter for the last call made by the activity to `RecordActivityTaskHeartbeat`.

- **activityTaskCanceledEventAttributes** (*dict*) –

If the event is of type `ActivityTaskCanceled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **details** (*string*) –

Details of the cancellation (if any).

- **scheduledEventId** (*integer*) –

The ID of the `ActivityTaskScheduled` event that was recorded when this activity task was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ActivityTaskStarted` event recorded when this activity task was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **latestCancelRequestedEventId** (*integer*) –

If set, contains the ID of the last `ActivityTaskCancelRequested` event recorded for this activity task. This information can be useful for

diagnosing problems by tracing back the chain of events leading up to this event.

- **activityTaskCancelRequestedEventAttributes** (*dict*) –

If the event is of type `ActivityTaskCancelRequested` then this member is set and provides detailed information about the event. It is not set for other event types.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RequestCancelActivityTask` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **activityId** (*string*) –

The unique ID of the task.

- **workflowExecutionSignaledEventAttributes** (*dict*) –

If the event is of type `WorkflowExecutionSignaled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **signalName** (*string*) –

The name of the signal received. The decider can use the signal name and inputs to determine how to process the signal.

- **input** (*string*) –

Inputs provided with the signal (if any). The decider can use the signal name and inputs to determine how to process the signal.

- **externalWorkflowExecution** (*dict*) –

The workflow execution that sent the signal. This is set only if the signal was sent by another workflow execution.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **externalInitiatedEventId** (*integer*) –

The ID of the `SignalExternalWorkflowExecutionInitiated` event corresponding to the `SignalExternalWorkflow` decision to signal this workflow execution. The source event with this ID can be found in the history of the source workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event. This field is set only if the signal was initiated by another workflow execution.

- **markerRecordedEventAttributes** (*dict*) –

If the event is of type `MarkerRecorded` then this member

is set and provides detailed information about the event. It is not set for other event types.

- **markerName** (*string*) –

The name of the marker.

- **details** (*string*) –

Details of the marker (if any).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RecordMarker` decision that requested this marker. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **recordMarkerFailedEventAttributes** (*dict*) –

If the event is of type `DecisionTaskFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **markerName** (*string*) –

The marker's name.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RecordMarkerFailed` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **timerStartedEventAttributes** (*dict*) –

If the event is of type `TimerStarted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timerId** (*string*) –

The unique ID of the timer that was started.

- **control** (*string*) –

Optional. Data attached to the event that can be used by the decider in subsequent workflow tasks.

- **startToFireTimeout** (*string*) –

The duration of time after which the timer will fire.

The duration is specified in seconds; an integer greater than or equal to 0.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `StartTimer` decision for this activity task. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **timerFiredEventAttributes** (*dict*) –

If the event is of type `TimerFired` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timerId** (*string*) –

The unique ID of the timer that fired.

- **startedEventId** (*integer*) –

The ID of the `TimerStarted` event that was recorded when this timer was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **timerCanceledEventAttributes** (*dict*) –

If the event is of type `TimerCanceled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timerId** (*string*) –

The unique ID of the timer that was canceled.

- **startedEventId** (*integer*) –

The ID of the `TimerStarted` event that was recorded when this timer was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CancelTimer` decision to cancel this timer. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startChildWorkflowExecutionInitiatedEventAttributes** (*dict*) –

If the event is of type `StartChildWorkflowExecutionInitiated` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowId** (*string*) –

The `workflowId` of the child workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique within a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique within a domain.

- **control** (*string*) –

Optional. Data attached to the event that can be used by the decider in subsequent decision tasks. This data is not sent to the activity.

- **input** (*string*) –

The inputs provided to the child workflow execution (if any).

- **executionStartToCloseTimeout** (*string*) –

The maximum duration for the child workflow execution. If the workflow execution is not closed within this duration, it will be timed out and force terminated.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **taskList** (*dict*) –

The name of the task list used for the decision tasks of the child workflow execution.

- **name** (*string*) –

The name of the task list.

- **taskPriority** (*string*) –

Optional. The priority assigned for the decision tasks for this workflow execution. Valid values are integers that range from Java’s `Integer.MIN_VALUE` (-2147483648) to `Integer.MAX_VALUE` (2147483647). Higher numbers indicate higher priority.

For more information about setting task priority, see [Setting Task Priority](#) in the *Amazon Simple Workflow Developer Guide*.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `StartChildWorkflowExecution` decision to request this child workflow execution. This information can be useful for diagnosing problems by tracing back the cause of events.

- **childPolicy** (*string*) –

The policy to use for the child workflow executions if this execution gets terminated by explicitly calling the `TerminateWorkflowExecution` action or due to an expired timeout.

The supported child policies are:

- **TERMINATE:** the child executions will be terminated.
- **REQUEST_CANCEL:** a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in

its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.

- **ABANDON:** no action will be taken. The child executions will continue to run.
- **taskStartToCloseTimeout** (*string*) –

The maximum duration allowed for the decision tasks for this workflow execution.

The duration is specified in seconds; an integer greater than or equal to 0. The value “NONE” can be used to specify unlimited duration.

- **tagList** (*list*) –

The list of tags to associated with the child workflow execution.

- (*string*) –
- **lambdaRole** (*string*) –

The IAM role attached to this workflow execution to use when invoking AWS Lambda functions.

- **childWorkflowExecutionStartedEventAttributes** (*dict*) –

If the event is of type `ChildWorkflowExecutionStarted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that was started.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful

for diagnosing problems by tracing back the chain of events leading up to this event.

- **childWorkflowExecutionCompletedEventAttributes** (*dict*) –

If the event is of type `ChildWorkflowExecutionCompleted` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that was completed.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **result** (*string*) –

The result of the child workflow execution (if any).

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ChildWorkflowExecutionStarted` event recorded when this child workflow execution was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **childWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `ChildWorkflowExecutionFailed` then this member

is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that failed.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **reason** (*string*) –

The reason for the failure (if provided).

- **details** (*string*) –

The details of the failure (if provided).

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ChildWorkflowExecutionStarted` event recorded when this child workflow execution was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **childWorkflowExecutionTimedOutEventAttributes** (*dict*) –

If the event is of type `ChildWorkflowExecutionTimedOut` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that timed out.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **timeoutType** (*string*) –

The type of the timeout that caused the child workflow execution to time out.

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ChildWorkflowExecutionStarted` event recorded when this child workflow execution was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **childWorkflowExecutionCanceledEventAttributes** (*dict*) –

If the event is of type `ChildWorkflowExecutionCanceled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that was canceled.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **details** (*string*) –

Details of the cancellation (if provided).

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ChildWorkflowExecutionStarted` event recorded when this child workflow execution was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **childWorkflowExecutionTerminatedEventAttributes** (*dict*) –

If the event is of type `ChildWorkflowExecutionTerminated` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The child workflow execution that was terminated.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **workflowType** (*dict*) –

The type of the child workflow execution.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version

must be unique with in a domain.

-
- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

-
- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `ChildWorkflowExecutionStarted` event recorded when this child workflow execution was started. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **signalExternalWorkflowExecutionInitiatedEventAttributes** (*dict*) –

If the event is of type `SignalExternalWorkflowExecutionInitiated` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowId** (*string*) –

The `workflowId` of the external workflow execution.

- **runId** (*string*) –

The `runId` of the external workflow execution to send the signal to.

- **signalName** (*string*) –

The name of the signal.

- **input** (*string*) –

Input provided to the signal (if any).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `SignalExternalWorkflowExecution` decision for this signal. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **control** (*string*) –

Optional. data attached to the event that can be used by the decider in subsequent decision tasks.

- **externalWorkflowExecutionSignaledEventAttributes** (*dict*) –

If the event is of type

`ExternalWorkflowExecutionSignaled` then this member is set and provides detailed information about the event. It is not set for other event types.

- **`workflowExecution`** (*dict*) –

The external workflow execution that the signal was delivered to.

- **`workflowId`** (*string*) –

The user defined identifier associated with the workflow execution.

- **`runId`** (*string*) –

A system-generated unique identifier for the workflow execution.

- **`initiatedEventId`** (*integer*) –

The ID of the `SignalExternalWorkflowExecutionInitiated` event corresponding to the `SignalExternalWorkflowExecution` decision to request this signal. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **`signalExternalWorkflowExecutionFailedEventAttributes`** (*dict*) –

If the event is of type `SignalExternalWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **`workflowId`** (*string*) –

The `workflowId` of the external workflow execution that the signal was being delivered to.

- **`runId`** (*string*) –

The `runId` of the external workflow execution that the signal was being delivered to.

- **`cause`** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If `cause` is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **`initiatedEventId`** (*integer*) –

The ID of the `SignalExternalWorkflowExecutionInitiated` event corresponding to the `SignalExternalWorkflowExecution` decision to request this signal. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **`decisionTaskCompletedEventId`** (*integer*) –

The ID of the `DecisionTaskCompleted` event cor-

responding to the decision task that resulted in the `SignalExternalWorkflowExecution` decision for this signal. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **control** (*string*) –
- **externalWorkflowExecutionCancelRequestedEventAttributes** (*dict*) –

If the event is of type `ExternalWorkflowExecutionCancelRequested` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowExecution** (*dict*) –

The external workflow execution to which the cancellation request was delivered.

- **workflowId** (*string*) –

The user defined identifier associated with the workflow execution.

- **runId** (*string*) –

A system-generated unique identifier for the workflow execution.

- **initiatedEventId** (*integer*) –

The ID of the `RequestCancelExternalWorkflowExecutionInitiated` event corresponding to the `RequestCancelExternalWorkflowExecution` decision to cancel this external workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **requestCancelExternalWorkflowExecutionInitiatedEventAttributes** (*dict*) –

If the event is of type `RequestCancelExternalWorkflowExecutionInitiated` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowId** (*string*) –

The `workflowId` of the external workflow execution to be canceled.

- **runId** (*string*) –

The `runId` of the external workflow execution to be canceled.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RequestCancelExternalWorkflowExecution` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **control** (*string*) –

Optional. Data attached to the event that can be used by the decider in subsequent workflow tasks.

- **requestCancelExternalWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `RequestCancelExternalWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowId** (*string*) –

The `workflowId` of the external workflow to which the cancel request was to be delivered.

- **runId** (*string*) –

The `runId` of the external workflow execution.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **initiatedEventId** (*integer*) –

The ID of the `RequestCancelExternalWorkflowExecutionInitiated` event corresponding to the `RequestCancelExternalWorkflowExecution` decision to cancel this external workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RequestCancelExternalWorkflowExecution` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **control** (*string*) –

- **scheduleActivityTaskFailedEventAttributes** (*dict*) –

If the event is of type `ScheduleActivityTaskFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **activityType** (*dict*) –

The activity type provided in the `ScheduleActivityTask` decision that failed.

- **name** (*string*) –

The name of this activity.

Note: The combination of activity type name and version must be unique within a domain.

- **version** (*string*) –

The version of this activity.

Note: The combination of activity type name and version must be unique within a domain.

- **activityId** (*string*) –

The activityId provided in the `ScheduleActivityTask` decision that failed.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision that resulted in the scheduling of this activity task. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **requestCancelActivityTaskFailedEventAttributes** (*dict*) –

If the event is of type `RequestCancelActivityTaskFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **activityId** (*string*) –

The activityId provided in the `RequestCancelActivityTask` decision that failed.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `RequestCancelActivityTask` decision for this cancellation request. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startTimerFailedEventAttributes** (*dict*) –

If the event is of type `StartTimerFailed` then this mem-

ber is set and provides detailed information about the event. It is not set for other event types.

- **timerId** (*string*) –

The timerId provided in the `StartTimer` decision that failed.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `StartTimer` decision for this activity task. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **cancelTimerFailedEventAttributes** (*dict*) –

If the event is of type `CancelTimerFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **timerId** (*string*) –

The timerId provided in the `CancelTimer` decision that failed.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `CancelTimer` decision to cancel this timer. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startChildWorkflowExecutionFailedEventAttributes** (*dict*) –

If the event is of type `StartChildWorkflowExecutionFailed` then this member is set and provides detailed information about the event. It is not set for other event types.

- **workflowType** (*dict*) –

The workflow type provided in the

`StartChildWorkflowExecution` decision that failed.

- **name** (*string*) –

Required. The name of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **version** (*string*) –

Required. The version of the workflow type.

Note: The combination of workflow type name and version must be unique with in a domain.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **workflowId** (*string*) –

The `workflowId` of the child workflow execution.

- **initiatedEventId** (*integer*) –

The ID of the `StartChildWorkflowExecutionInitiated` event corresponding to the `StartChildWorkflowExecution` decision to start this child workflow execution. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision task that resulted in the `StartChildWorkflowExecution` decision to request this child workflow execution. This information can be useful for diagnosing problems by tracing back the cause of events.

- **control** (*string*) –

- **lambdaFunctionScheduledEventAttributes** (*dict*) –

Provides details for the `LambdaFunctionScheduled` event.

- **id** (*string*) –

The unique Amazon SWF ID for the AWS Lambda task.

- **name** (*string*) –

The name of the scheduled AWS Lambda function.

- **input** (*string*) –

Input provided to the AWS Lambda function.

- **startToCloseTimeout** (*string*) –

The maximum time, in seconds, that the AWS Lambda function can take to execute from start to close before it is marked as failed.

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event for the decision that resulted in the scheduling of this AWS Lambda function. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **lambdaFunctionStartedEventAttributes** (*dict*) –

Provides details for the `LambdaFunctionStarted` event.

- **scheduledEventId** (*integer*) –

The ID of the `LambdaFunctionScheduled` event that was recorded when this AWS Lambda function was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **lambdaFunctionCompletedEventAttributes** (*dict*) –

Provides details for the `LambdaFunctionCompleted` event.

- **scheduledEventId** (*integer*) –

The ID of the `LambdaFunctionScheduled` event that was recorded when this AWS Lambda function was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `LambdaFunctionStarted` event recorded in the history.

- **result** (*string*) –

The result of the function execution (if any).

- **lambdaFunctionFailedEventAttributes** (*dict*) –

Provides details for the `LambdaFunctionFailed` event.

- **scheduledEventId** (*integer*) –

The ID of the `LambdaFunctionScheduled` event that was recorded when this AWS Lambda function was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `LambdaFunctionStarted` event recorded in the history.

- **reason** (*string*) –

The reason provided for the failure (if any).

- **details** (*string*) –

The details of the failure (if any).

- **lambdaFunctionTimedOutEventAttributes** (*dict*) –

Provides details for the `LambdaFunctionTimedOut` event.

- **scheduledEventId** (*integer*) –

The ID of the `LambdaFunctionScheduled` event that was recorded when this AWS Lambda function was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startedEventId** (*integer*) –

The ID of the `LambdaFunctionStarted` event recorded in the history.

- **timeoutType** (*string*) –

The type of the timeout that caused this event.

- **scheduleLambdaFunctionFailedEventAttributes** (*dict*) –

Provides details for the `ScheduleLambdaFunctionFailed` event.

- **id** (*string*) –

The unique Amazon SWF ID of the AWS Lambda task.

- **name** (*string*) –

The name of the scheduled AWS Lambda function.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **decisionTaskCompletedEventId** (*integer*) –

The ID of the `DecisionTaskCompleted` event corresponding to the decision that resulted in the scheduling of this AWS Lambda function. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **startLambdaFunctionFailedEventAttributes** (*dict*) –

Provides details for the `StartLambdaFunctionFailed` event.

- **scheduledEventId** (*integer*) –

The ID of the `LambdaFunctionScheduled` event that was recorded when this AWS Lambda function was scheduled. This information can be useful for diagnosing problems by tracing back the chain of events leading up to this event.

- **cause** (*string*) –

The cause of the failure. This information is generated by the system and can be useful for diagnostic purposes.

Note: If **cause** is set to `OPERATION_NOT_PERMITTED`, the decision failed because it lacked sufficient permissions. For details and example IAM policies, see [Using IAM to Manage Access to Amazon SWF Workflows](#).

- **message** (*string*) –

The error message (if any).

– **previousStartedEventId** (*integer*) –

The ID of the DecisionTaskStarted event of the previous decision task of this workflow execution that was processed by the decider. This can be used to determine the events in the history new since the last decision task received by the decider.

– **NextToken** (*string*) –

A token to resume pagination.

WorkSpaces

Table of Contents

- WorkSpaces
 - Client

Client

class WorkSpaces.Client

A low-level client representing Amazon WorkSpaces:

```
import boto3
```

```
client = boto3.client('workspaces')
```

These are the available methods:

- `can_paginate()`
- `create_workspaces()`
- `describe_workspace_bundles()`
- `describe_workspace_directories()`
- `describe_workspaces()`
- `generate_presigned_url()`
- `get_paginator()`
- `get_waiter()`
- `reboot_workspaces()`
- `rebuild_workspaces()`
- `terminate_workspaces()`

can_paginate (*operation_name*)

Check if an operation can be paginated.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Returns True if the operation can be paginated, False otherwise.

create_workspaces (***kwargs*)

Creates one or more WorkSpaces.

Note: This operation is asynchronous and returns before the WorkSpaces are created.

Request Syntax

```
response = client.create_workspaces(
    Workspaces=[
        {
            'DirectoryId': 'string',
            'UserName': 'string',
            'BundleId': 'string'
        },
    ]
)
```

Parameters **Workspaces** (*list*) – [REQUIRED]

An array of structures that specify the WorkSpaces to create.

- (*dict*) –

Contains information about a WorkSpace creation request.

- **DirectoryId** (*string*) – [REQUIRED]

The identifier of the AWS Directory Service directory to create the WorkSpace in. You can use the DescribeWorkspaceDirectories operation to obtain a list of the directories that are available.

- **UserName** (*string*) – [REQUIRED]

The username that the WorkSpace is assigned to. This username must exist in the AWS Directory Service directory specified by the DirectoryId member.

- **BundleId** (*string*) – [REQUIRED]

The identifier of the bundle to create the WorkSpace from. You can use the DescribeWorkspaceBundles operation to obtain a list of the bundles that are available.

Return type dict

Returns

Response Syntax

```
{
    'FailedRequests': [
        {
            'WorkspaceRequest': {
                'DirectoryId': 'string',
                'UserName': 'string',
                'BundleId': 'string'
            },
            'ErrorCode': 'string',
            'ErrorMessage': 'string'
        },
    ],
    'PendingRequests': [
        {
            'WorkspaceId': 'string',
            'DirectoryId': 'string',
            'UserName': 'string',
            'IpAddress': 'string',
            'State': 'PENDING' | 'AVAILABLE' | 'IMPAIRED' | 'UNHEALTHY' | 'REBOOTING' | 'REBOOTING_FAILED' | 'STOPPED',
            'BundleId': 'string',
            'SubnetId': 'string',
        },
    ],
}
```

```

        'ErrorMessage': 'string',
        'ErrorCode': 'string'
    },
]
}

```

Response Structure

- *(dict)* –

Contains the result of the CreateWorkspaces operation.

- **FailedRequests** (*list*) –

An array of structures that represent the WorkSpaces that could not be created.

- * *(dict)* –

Contains information about a Workspace that could not be created.

- **WorkspaceRequest** (*dict*) –

A WorkspaceRequest object that contains the information about the Workspace that could not be created.

- **DirectoryId** (*string*) –

The identifier of the AWS Directory Service directory to create the Workspace in. You can use the DescribeWorkspaceDirectories operation to obtain a list of the directories that are available.

- **UserName** (*string*) –

The username that the Workspace is assigned to. This username must exist in the AWS Directory Service directory specified by the DirectoryId member.

- **BundleId** (*string*) –

The identifier of the bundle to create the Workspace from. You can use the DescribeWorkspaceBundles operation to obtain a list of the bundles that are available.

- **ErrorCode** (*string*) –

The error code.

- **ErrorMessage** (*string*) –

The textual error message.

- **PendingRequests** (*list*) –

An array of structures that represent the WorkSpaces that were created.

Because this operation is asynchronous, the identifier in WorkspaceId is not immediately available. If you immediately call DescribeWorkspaces with this identifier, no information will be returned.

- * *(dict)* –

Contains information about a Workspace.

- **WorkspaceId** (*string*) –

The identifier of the Workspace.

- **DirectoryId** (*string*) –

The identifier of the AWS Directory Service directory that the Workspace belongs to.

- **UserName** (*string*) –
The user that the Workspace is assigned to.
- **IpAddress** (*string*) –
The IP address of the Workspace.
- **State** (*string*) –
The operational state of the Workspace.
- **BundleId** (*string*) –
The identifier of the bundle that the Workspace was created from.
- **SubnetId** (*string*) –
The identifier of the subnet that the Workspace is in.
- **ErrorMessage** (*string*) –
If the Workspace could not be created, this contains a textual error message that describes the failure.
- **ErrorCode** (*string*) –
If the Workspace could not be created, this contains the error code.

describe_workspace_bundles (***kwargs*)

Obtains information about the Workspace bundles that are available to your account in the specified region.

You can filter the results with either the `BundleIds` parameter, or the `Owner` parameter, but not both.

This operation supports pagination with the use of the `NextToken` request and response parameters. If more results are available, the `NextToken` response member contains a token that you pass in the next call to this operation to retrieve the next set of items.

Request Syntax

```
response = client.describe_workspace_bundles(  
    BundleIds=[  
        'string',  
    ],  
    Owner='string',  
    NextToken='string'  
)
```

Parameters

- **BundleIds** (*list*) – An array of strings that contains the identifiers of the bundles to retrieve. This parameter cannot be combined with any other filter parameter.
 - (*string*) –
- **Owner** (*string*) – The owner of the bundles to retrieve. This parameter cannot be combined with any other filter parameter.

This contains one of the following values:

- `null` - Retrieves the bundles that belong to the account making the call.
- `AMAZON` - Retrieves the bundles that are provided by AWS.
- **NextToken** (*string*) – The `NextToken` value from a previous call to this operation. Pass `null` if this is the first call.

Return type dict

Returns

Response Syntax


```
{
  'Bundles': [
    {
      'BundleId': 'string',
      'Name': 'string',
      'Owner': 'string',
      'Description': 'string',
      'UserStorage': {
        'Capacity': 'string'
      },
      'ComputeType': {
        'Name': 'VALUE' | 'STANDARD' | 'PERFORMANCE'
      }
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the results of the DescribeWorkspaceBundles operation.

- **Bundles** (*list*) –

An array of structures that contain information about the bundles.

- * (*dict*) –

Contains information about a Workspace bundle.

- **BundleId** (*string*) –

The bundle identifier.

- **Name** (*string*) –

The name of the bundle.

- **Owner** (*string*) –

The owner of the bundle. This contains the owner's account identifier, or `AMAZON` if the bundle is provided by AWS.

- **Description** (*string*) –

The bundle description.

- **UserStorage** (*dict*) –

A UserStorage object that specifies the amount of user storage that the bundle contains.

- **Capacity** (*string*) –

The amount of user storage for the bundle.

- **ComputeType** (*dict*) –

A ComputeType object that specifies the compute type for the bundle.

- **Name** (*string*) –

The name of the compute type for the bundle.

- **NextToken** (*string*) –

If not null, more results are available. Pass this value for the `NextToken` parameter in a subsequent call to this operation to retrieve the next set of items. This token is valid for one day and must be used within that timeframe.

describe_workspace_directories (***kwargs*)

Retrieves information about the AWS Directory Service directories in the region that are registered with Amazon WorkSpaces and are available to your account.

This operation supports pagination with the use of the `NextToken` request and response parameters. If more results are available, the `NextToken` response member contains a token that you pass in the next call to this operation to retrieve the next set of items.

Request Syntax

```
response = client.describe_workspace_directories(
    DirectoryIds=[
        'string',
    ],
    NextToken='string'
)
```

Parameters

- **DirectoryIds** (*list*) – An array of strings that contains the directory identifiers to retrieve information for. If this member is null, all directories are retrieved.
 - (*string*) –
- **NextToken** (*string*) – The `NextToken` value from a previous call to this operation. Pass null if this is the first call.

Return type dict

Returns**Response Syntax**

```
{
    'Directories': [
        {
            'DirectoryId': 'string',
            'Alias': 'string',
            'DirectoryName': 'string',
            'RegistrationCode': 'string',
            'SubnetIds': [
                'string',
            ],
            'DnsIpAddresses': [
                'string',
            ],
            'CustomerUserName': 'string',
            'IamRoleId': 'string',
            'DirectoryType': 'SIMPLE_AD' | 'AD_CONNECTOR',
            'WorkspaceSecurityGroupId': 'string',
            'State': 'REGISTERING' | 'REGISTERED' | 'DEREGISTERING' | 'DEREGISTERED' | 'ERROR',
            'WorkspaceCreationProperties': {
                'EnableWorkDocs': True|False,
                'EnableInternetAccess': True|False,
                'DefaultOu': 'string',
                'CustomSecurityGroupId': 'string',
                'UserEnabledAsLocalAdministrator': True|False
            }
        },
    ],
    'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the results of the DescribeWorkspaceDirectories operation.

- **Directories** (*list*) –

An array of structures that contain information about the directories.

- * (*dict*) –

Contains information about an AWS Directory Service directory for use with Amazon WorkSpaces.

- **DirectoryId** (*string*) –

The directory identifier.

- **Alias** (*string*) –

The directory alias.

- **DirectoryName** (*string*) –

The name of the directory.

- **RegistrationCode** (*string*) –

The registration code for the directory. This is the code that users enter in their Amazon WorkSpaces client application to connect to the directory.

- **SubnetIds** (*list*) –

An array of strings that contains the identifiers of the subnets used with the directory.

- (*string*) –

- **DnsIpAddresses** (*list*) –

An array of strings that contains the IP addresses of the DNS servers for the directory.

- (*string*) –

- **CustomerUserName** (*string*) –

The user name for the service account.

- **IamRoleId** (*string*) –

The identifier of the IAM role. This is the role that allows Amazon WorkSpaces to make calls to other services, such as Amazon EC2, on your behalf.

- **DirectoryType** (*string*) –

The directory type.

- **WorkspaceSecurityGroupId** (*string*) –

The identifier of the security group that is assigned to new WorkSpaces.

- **State** (*string*) –

The state of the directory's registration with Amazon WorkSpaces

- **WorkspaceCreationProperties** (*dict*) –

A structure that specifies the default creation properties for all WorkSpaces in the directory.

- **EnableWorkDocs** (*boolean*) –

Specifies if the directory is enabled for Amazon WorkDocs.

- **EnableInternetAccess** (*boolean*) –

A public IP address will be attached to all WorkSpaces that are created or rebuilt.

- **DefaultOu** (*string*) –

The organizational unit (OU) in the directory that the WorkSpace machine accounts are placed in.

- **CustomSecurityGroupId** (*string*) –

The identifier of any custom security groups that are applied to the WorkSpaces when they are created.

- **UserEnabledAsLocalAdministrator** (*boolean*) –

The WorkSpace user is an administrator on the WorkSpace.

- **NextToken** (*string*) –

If not null, more results are available. Pass this value for the `NextToken` parameter in a subsequent call to this operation to retrieve the next set of items. This token is valid for one day and must be used within that timeframe.

describe_workspaces (***kwargs*)

Obtains information about the specified WorkSpaces.

Only one of the filter parameters, such as `BundleId`, `DirectoryId`, or `WorkspaceIds`, can be specified at a time.

This operation supports pagination with the use of the `NextToken` request and response parameters. If more results are available, the `NextToken` response member contains a token that you pass in the next call to this operation to retrieve the next set of items.

Request Syntax

```
response = client.describe_workspaces(
    WorkspaceIds=[
        'string',
    ],
    DirectoryId='string',
    UserName='string',
    BundleId='string',
    Limit=123,
    NextToken='string'
)
```

Parameters

- **WorkspaceIds** (*list*) – An array of strings that contain the identifiers of the WorkSpaces for which to retrieve information. This parameter cannot be combined with any other filter parameter.

Because the `CreateWorkspaces` operation is asynchronous, the identifier returned by `CreateWorkspaces` is not immediately available. If you immediately call `DescribeWorkspaces` with this identifier, no information will be returned.

– (*string*) –

- **DirectoryId** (*string*) – Specifies the directory identifier to which to limit the WorkSpaces. Optionally, you can specify a specific directory user with the `UserName` parameter. This parameter cannot be combined with any other filter parameter.
- **UserName** (*string*) – Used with the `DirectoryId` parameter to specify the directory user for which to obtain the WorkSpace.

- **BundleId** (*string*) – The identifier of a bundle to obtain the WorkSpaces for. All WorkSpaces that are created from this bundle will be retrieved. This parameter cannot be combined with any other filter parameter.
- **Limit** (*integer*) – The maximum number of items to return.
- **NextToken** (*string*) – The NextToken value from a previous call to this operation. Pass null if this is the first call.

Return type dict

Returns

Response Syntax

```
{
  'Workspaces': [
    {
      'WorkspaceId': 'string',
      'DirectoryId': 'string',
      'UserName': 'string',
      'IpAddress': 'string',
      'State': 'PENDING' | 'AVAILABLE' | 'IMPAIRED' | 'UNHEALTHY' | 'REBOOTING' | 'REBOOTING_FAILED',
      'BundleId': 'string',
      'SubnetId': 'string',
      'ErrorMessage': 'string',
      'ErrorCode': 'string'
    },
  ],
  'NextToken': 'string'
}
```

Response Structure

- (*dict*) –

Contains the results for the DescribeWorkspaces operation.

- **Workspaces** (*list*) –

An array of structures that contain the information about the WorkSpaces.

Because the CreateWorkspaces operation is asynchronous, some of this information may be incomplete for a newly-created Workspace.

- * (*dict*) –

Contains information about a Workspace.

- **WorkspaceId** (*string*) –
The identifier of the Workspace.
- **DirectoryId** (*string*) –
The identifier of the AWS Directory Service directory that the Workspace belongs to.
- **UserName** (*string*) –
The user that the Workspace is assigned to.
- **IpAddress** (*string*) –
The IP address of the Workspace.
- **State** (*string*) –
The operational state of the Workspace.
- **BundleId** (*string*) –
The identifier of the bundle that the Workspace was created from.

- **SubnetId** (*string*) –

The identifier of the subnet that the Workspace is in.

- **ErrorMessage** (*string*) –

If the Workspace could not be created, this contains a textual error message that describes the failure.

- **ErrorCode** (*string*) –

If the Workspace could not be created, this contains the error code.

- **NextToken** (*string*) –

If not null, more results are available. Pass this value for the `NextToken` parameter in a subsequent call to this operation to retrieve the next set of items. This token is valid for one day and must be used within that timeframe.

generate_presigned_url (*ClientMethod*, *Params=None*, *ExpiresIn=3600*, *HttpMethod=None*)

Generate a presigned url given a client, its method, and arguments

Parameters

- **ClientMethod** (*string*) – The client method to presign for
- **Params** (*dict*) – The parameters normally passed to `ClientMethod`.
- **ExpiresIn** (*int*) – The number of seconds the presigned url is valid for. By default it expires in an hour (3600 seconds)
- **HttpMethod** (*string*) – The http method to use on the generated url. By default, the http method is whatever is used in the method's model.

Returns The presigned url

get_paginator (*operation_name*)

Create a paginator for an operation.

Parameters *operation_name* (*string*) – The operation name. This is the same name as the method name on the client. For example, if the method name is `create_foo`, and you'd normally invoke the operation as `client.create_foo(**kwargs)`, if the `create_foo` operation can be paginated, you can use the call `client.get_paginator("create_foo")`.

Raises **OperationNotPageableError** Raised if the operation is not pageable. You can use the `client.can_paginate` method to check if an operation is pageable.

Return type `L{botocore.paginate.Paginator}`

Returns A paginator object.

get_waiter (*waiter_name*)

reboot_workspaces (***kwargs*)

Reboots the specified WorkSpaces.

To be able to reboot a Workspace, the Workspace must have a **State** of `AVAILABLE`, `IMPAIRED`, or `INOPERABLE`.

Note: This operation is asynchronous and will return before the WorkSpaces have rebooted.

Request Syntax

```
response = client.reboot_workspaces(
    RebootWorkspaceRequests=[
        {
            'WorkspaceId': 'string'
        },
    ],
```

```
]
)
```

Parameters `RebootWorkspaceRequests` (*list*) – [REQUIRED]

An array of structures that specify the WorkSpaces to reboot.

- (*dict*) –

Contains information used with the `RebootWorkspaces` operation to reboot a `Workspace`.

- **WorkspaceId** (*string*) – [REQUIRED]

The identifier of the `Workspace` to reboot.

Return type `dict`

Returns

Response Syntax

```
{
  'FailedRequests': [
    {
      'WorkspaceId': 'string',
      'ErrorCode': 'string',
      'ErrorMessage': 'string'
    },
  ]
}
```

Response Structure

- (*dict*) –

Contains the results of the `RebootWorkspaces` operation.

- **FailedRequests** (*list*) –

An array of structures that represent any `WorkSpaces` that could not be rebooted.

- * (*dict*) –

Contains information about a `Workspace` that could not be rebooted (`RebootWorkspaces`), rebuilt (`RebuildWorkspaces`), or terminated (`TerminateWorkspaces`).

- **WorkspaceId** (*string*) –

The identifier of the `Workspace`.

- **ErrorCode** (*string*) –

The error code.

- **ErrorMessage** (*string*) –

The textual error message.

rebuild_workspaces (***kwargs*)

Rebuilds the specified `WorkSpaces`.

Rebuilding a `Workspace` is a potentially destructive action that can result in the loss of data. Rebuilding a `Workspace` causes the following to occur:

- The system is restored to the image of the bundle that the `Workspace` is created from. Any applications that have been installed, or system settings that have been made since the `Workspace` was created will be lost.

- The data drive (D drive) is re-created from the last automatic snapshot taken of the data drive. The current contents of the data drive are overwritten. Automatic snapshots of the data drive are taken every 12 hours, so the snapshot can be as much as 12 hours old.

To be able to rebuild a WorkSpace, the WorkSpace must have a **State** of `AVAILABLE` or `ERROR`.

Note: This operation is asynchronous and will return before the WorkSpaces have been completely rebuilt.

Request Syntax

```
response = client.rebuild_workspaces(
    RebuildWorkspaceRequests=[
        {
            'WorkspaceId': 'string'
        },
    ]
)
```

Parameters **RebuildWorkspaceRequests** (*list*) – [REQUIRED]

An array of structures that specify the WorkSpaces to rebuild.

- (*dict*) –

Contains information used with the RebuildWorkspaces operation to rebuild a WorkSpace.

- **WorkspaceId** (*string*) – [REQUIRED]

The identifier of the WorkSpace to rebuild.

Return type dict

Returns

Response Syntax

```
{
    'FailedRequests': [
        {
            'WorkspaceId': 'string',
            'ErrorCode': 'string',
            'ErrorMessage': 'string'
        },
    ]
}
```

Response Structure

- (*dict*) –

Contains the results of the RebuildWorkspaces operation.

- **FailedRequests** (*list*) –

An array of structures that represent any WorkSpaces that could not be rebuilt.

- * (*dict*) –

Contains information about a WorkSpace that could not be rebooted (RebootWorkspaces), rebuilt (RebuildWorkspaces), or terminated (TerminateWorkspaces).

- **WorkspaceId** (*string*) –

The identifier of the WorkSpace.

- **ErrorCode** (*string*) –
The error code.
- **ErrorMessage** (*string*) –
The textual error message.

terminate_workspaces (***kwargs*)

Terminates the specified WorkSpaces.

Terminating a WorkSpace is a permanent action and cannot be undone. The user's data is not maintained and will be destroyed. If you need to archive any user data, contact Amazon Web Services before terminating the WorkSpace.

You can terminate a WorkSpace that is in any state except `SUSPENDED`.

Note: This operation is asynchronous and will return before the WorkSpaces have been completely terminated.

Request Syntax

```
response = client.terminate_workspaces(
    TerminateWorkspaceRequests=[
        {
            'WorkspaceId': 'string'
        },
    ]
)
```

Parameters **TerminateWorkspaceRequests** (*list*) – **[REQUIRED]**

An array of structures that specify the WorkSpaces to terminate.

- (*dict*) –
Contains information used with the TerminateWorkspaces operation to terminate a WorkSpace.
 - **WorkspaceId** (*string*) – **[REQUIRED]**
The identifier of the WorkSpace to terminate.

Return type dict

Returns

Response Syntax

```
{
    'FailedRequests': [
        {
            'WorkspaceId': 'string',
            'ErrorCode': 'string',
            'ErrorMessage': 'string'
        },
    ]
}
```

Response Structure

- (*dict*) –
Contains the results of the TerminateWorkspaces operation.

– **FailedRequests** (*list*) –

An array of structures that represent any WorkSpaces that could not be terminated.

* (*dict*) –

Contains information about a Workspace that could not be rebooted (`RebootWorkspaces`), rebuilt (`RebuildWorkspaces`), or terminated (`TerminateWorkspaces`).

· **WorkspaceId** (*string*) –

The identifier of the Workspace.

· **ErrorCode** (*string*) –

The error code.

· **ErrorMessage** (*string*) –

The textual error message.

3.2 Core

3.2.1 Core References

Boto3 Reference

`class boto3.NullHandler (level=0)`

Initializes the instance - basically setting the formatter to None and the filter list to empty.

`emit (record)`

`boto3.client (*args, **kwargs)`

Create a low-level service client by name using the default session.

See `boto3.session.Session.client()`.

`boto3.resource (*args, **kwargs)`

Create a resource service client by name using the default session.

See `boto3.session.Session.resource()`.

`boto3.set_stream_logger (name='boto3', level=10, format_string=None)`

Add a stream handler for the given name and level to the logging module. By default, this logs all boto3 messages to stdout.

```
>>> import boto3
>>> boto3.set_stream_logger('boto3.resources', logging.INFO)
```

Parameters

- **name** (*string*) – Log name
- **level** (*int*) – Logging level, e.g. `logging.INFO`
- **format_string** (*str*) – Log message format

`boto3.setup_default_session (**kwargs)`

Set up a default session, passing through any parameters to the session constructor. There is no need to call this unless you wish to pass custom parameters, because a default session will be created for you.

Collections Reference

class `boto3.resources.collection.CollectionFactory`

A factory to create new `CollectionManager` and `ResourceCollection` subclasses from a `Collection` model. These subclasses include methods to perform batch operations.

load_from_definition (*service_name, resource_name, collection_name, model, resource_defs*)

Loads a collection from a model, creating a new `CollectionManager` subclass with the correct properties and methods, named based on the service and resource name, e.g. `ec2.InstanceCollectionManager`. It also creates a new `ResourceCollection` subclass which is used by the new manager class.

Parameters

- **service_name** (*string*) – Name of the service to look up
- **resource_name** (*string*) – Name of the resource to look up. For services, this should match the `service_name`.
- **model** (*dict*) – The collection definition.
- **resource_defs** (*dict*) – The service's resource definitions, used to load collection resources (e.g. `sqs.Queue`).

Return type Subclass of `CollectionManager`

Returns The collection class.

class `boto3.resources.collection.CollectionManager` (*model, parent, factory, resource_defs, service_model*)

A collection manager provides access to resource collection instances, which can be iterated and filtered. The manager exposes some convenience functions that are also found on resource collections, such as `all()` and `filter()`.

Get all items:

```
>>> for bucket in s3.buckets.all():
...     print(bucket.name)
```

Get only some items via filtering:

```
>>> for queue in sqs.queues.filter(QueueNamePrefix='AWS'):
...     print(queue.url)
```

Get whole pages of items:

```
>>> for page in s3.Bucket('boto3').objects.pages():
...     for obj in page:
...         print(obj.key)
```

A collection manager is not iterable. You **must** call one of the methods that return a `ResourceCollection` before trying to iterate, slice, or convert to a list.

See the [Collections](#) guide for a high-level overview of collections, including when remote service requests are performed.

Parameters

- **model** (`Collection`) – Collection model
- **parent** (`ServiceResource`) – The collection's parent resource
- **factory** (`ResourceFactory`) – The resource factory to create new resources
- **resource_defs** (*dict*) – Service resource definitions.
- **service_model** (*botocore.model.ServiceModel*) – The Botocore service model

all()

Get all items from the collection, optionally with a custom page size and item count limit.

This method returns an iterable generator which yields individual resource instances. Example use:

```
# Iterate through items
>>> for queue in sqs.queues.all():
...     print(queue.url)
'https://url1'
'https://url2'

# Convert to list
>>> queues = list(sqs.queues.all())
>>> len(queues)
2
```

filter (***kwargs*)

Get items from the collection, passing keyword arguments along as parameters to the underlying service operation, which are typically used to filter the results.

This method returns an iterable generator which yields individual resource instances. Example use:

```
# Iterate through items
>>> for queue in sqs.queues.filter(Param='foo'):
...     print(queue.url)
'https://url1'
'https://url2'

# Convert to list
>>> queues = list(sqs.queues.filter(Param='foo'))
>>> len(queues)
2
```

Return type `ResourceCollection`

iterator (***kwargs*)

Get a resource collection iterator from this manager.

Return type `ResourceCollection`

Returns An iterable representing the collection of resources

limit (*count*)

Return at most this many resources.

```
>>> for bucket in s3.buckets.limit(5):
...     print(bucket.name)
'bucket1'
'bucket2'
'bucket3'
'bucket4'
'bucket5'
```

Parameters **count** (*int*) – Return no more than this many items

Return type `ResourceCollection`

page_size (*count*)

Fetch at most this many resources per service request.

```
>>> for obj in s3.Bucket('boto3').objects.page_size(100):
...     print(obj.key)
```

Parameters **count** (*int*) – Fetch this many items per request

Return type `ResourceCollection`

pages()

A generator which yields pages of resource instances after doing the appropriate service operation calls and handling any pagination on your behalf. Non-paginated calls will return a single page of items.

Page size, item limit, and filter parameters are applied if they have previously been set.

```
>>> bucket = s3.Bucket('boto3')
>>> for page in bucket.objects.pages():
...     for obj in page:
...         print(obj.key)
'key1'
'key2'
```

Return type `list(ServiceResource)`

Returns List of resource instances

class `boto3.resources.collection.ResourceCollection` (*model, parent, handler, **kwargs*)

Represents a collection of resources, which can be iterated through, optionally with filtering. Collections automatically handle pagination for you.

See [Collections](#) for a high-level overview of collections, including when remote service requests are performed.

Parameters

- **model** (`Collection`) – Collection model
- **parent** (`ServiceResource`) – The collection’s parent resource
- **handler** (`ResourceHandler`) – The resource response handler used to create resource instances

all()

Get all items from the collection, optionally with a custom page size and item count limit.

This method returns an iterable generator which yields individual resource instances. Example use:

```
# Iterate through items
>>> for queue in sqs.queues.all():
...     print(queue.url)
'https://url1'
'https://url2'

# Convert to list
>>> queues = list(sqs.queues.all())
>>> len(queues)
2
```

filter(kwargs)**

Get items from the collection, passing keyword arguments along as parameters to the underlying service operation, which are typically used to filter the results.

This method returns an iterable generator which yields individual resource instances. Example use:

```
# Iterate through items
>>> for queue in sqs.queues.filter(Param='foo'):
...     print(queue.url)
'https://url1'
'https://url2'

# Convert to list
>>> queues = list(sqs.queues.filter(Param='foo'))
>>> len(queues)
2
```

Return type `ResourceCollection`

limit (*count*)

Return at most this many resources.

```
>>> for bucket in s3.buckets.limit(5):
...     print(bucket.name)
'bucket1'
'bucket2'
'bucket3'
'bucket4'
'bucket5'
```

Parameters *count* (*int*) – Return no more than this many items

Return type `ResourceCollection`

page_size (*count*)

Fetch at most this many resources per service request.

```
>>> for obj in s3.Bucket('boto3').objects.page_size(100):
...     print(obj.key)
```

Parameters *count* (*int*) – Fetch this many items per request

Return type `ResourceCollection`

pages ()

A generator which yields pages of resource instances after doing the appropriate service operation calls and handling any pagination on your behalf. Non-paginated calls will return a single page of items.

Page size, item limit, and filter parameters are applied if they have previously been set.

```
>>> bucket = s3.Bucket('boto3')
>>> for page in bucket.objects.pages():
...     for obj in page:
...         print(obj.key)
'key1'
'key2'
```

Return type `list(ServiceResource)`

Returns List of resource instances

Resources Reference

Resource Model

The models defined in this file represent the resource JSON description format and provide a layer of abstraction from the raw JSON. The advantages of this are:

- Pythonic interface (e.g. `action.request.operation`)
- Consumers need not change for minor JSON changes (e.g. renamed field)

These models are used both by the resource factory to generate resource classes as well as by the documentation generator.

class `boto3.resources.model.Action` (*name*, *definition*, *resource_defs*)

A service operation action.

Parameters

- **name** (*string*) – The name of the action
- **definition** (*dict*) – The JSON definition
- **resource_defs** (*dict*) – All resources defined in the service

name = None

(*string*) The name of the action

path = None

(*string*) The JMESPath search path or None

request = None

(*Request*) This action's request or None

resource = None

(*ResponseResource*) This action's resource or None

class boto3.resources.model.**Collection** (*name, definition, resource_defs*)

A group of resources. See [Action](#).

Parameters

- **name** (*string*) – The name of the collection
- **definition** (*dict*) – The JSON definition
- **resource_defs** (*dict*) – All resources defined in the service

batch_actions

Get a list of batch actions supported by the resource type contained in this action. This is a shortcut for accessing the same information through the resource model.

Return type list(*Action*)

class boto3.resources.model.**DefinitionWithParams** (*definition*)

An item which has parameters exposed via the `params` property. A request has an operation and parameters, while a waiter has a name, a low-level waiter name and parameters.

Parameters **definition** (*dict*) – The JSON definition

params

Get a list of auto-filled parameters for this request.

Type list(*Parameter*)

class boto3.resources.model.**Identifier** (*name*)

A resource identifier, given by its name.

Parameters **name** (*string*) – The name of the identifier

name = None

(*string*) The name of the identifier

class boto3.resources.model.**Parameter** (*target, source, name=None, path=None, value=None, **kwargs*)

An auto-filled parameter which has a source and target. For example, the `QueueUrl` may be auto-filled from a resource's `url` identifier when making calls to `queue.receive_messages`.

Parameters

- **target** (*string*) – The destination parameter name, e.g. `QueueUrl`
- **source_type** (*string*) – Where the source is defined.
- **source** (*string*) – The source name, e.g. `Url`

name = None

(*string*) The name of the source, if given

path = None

(*string*) The JMESPath query of the source

source = None

(*string*) Where the source is defined

target = None

(*string*) The destination parameter name

value = None
(string|int|float|bool) The source constant value

class boto3.resources.model.**Request** (*definition*)

A service operation action request.

Parameters *definition* (*dict*) – The JSON definition

operation = None
(string) The name of the low-level service operation

params
Get a list of auto-filled parameters for this request.
Type list(*Parameter*)

class boto3.resources.model.**ResourceModel** (*name*, *definition*, *resource_defs*)

A model representing a resource, defined via a JSON description format. A resource has identifiers, attributes, actions, sub-resources, references and collections. For more information on resources, see [Resources](#).

Parameters

- **name** (*string*) – The name of this resource, e.g. `sqs` or `Queue`
- **definition** (*dict*) – The JSON definition
- **resource_defs** (*dict*) – All resources defined in the service

actions
Get a list of actions for this resource.
Type list(*Action*)

batch_actions
Get a list of batch actions for this resource.
Type list(*Action*)

collections
Get a list of collections for this resource.
Type list(*Collection*)

get_attributes (*shape*)
Get a dictionary of attribute names to original name and shape models that represent the attributes of this resource. Looks like the following:

```
{ 'some_name': ('SomeName', <Shape...>)  
}
```

Parameters *shape* (*botocore.model.Shape*) – The underlying shape for this resource.

Return type dict

Returns Mapping of resource attributes.

identifiers
Get a list of resource identifiers.
Type list(*Identifier*)

load
Get the load action for this resource, if it is defined.
Type *Action* or None

load_rename_map (*shape=None*)
Load a name translation map given a shape. This will set up renamed values for any collisions, e.g. if the shape, an action, and a subresource all are all named `foo` then the resource will have an action `foo`, a subresource named `Foo` and a property named `foo_attribute`. This is the order of precedence, from most important to least important:

- Load action (`resource.load`)
- Identifiers
- Actions

- Subresources
- References
- Collections
- Waiters
- Attributes (shape members)

Batch actions are only exposed on collections, so do not get modified here. Subresources use upper camel casing, so are unlikely to collide with anything but other subresources.

Creates a structure like this:

```
renames = {
    ('action', 'id'): 'id_action',
    ('collection', 'id'): 'id_collection',
    ('attribute', 'id'): 'id_attribute'
}

# Get the final name for an action named 'id'
name = renames.get(('action', 'id'), 'id')
```

Parameters `shape` (*botocore.model.Shape*) – The underlying shape for this resource.

name = None

(string) The name of this resource

references

Get a list of reference resources.

Type list(*ResponseResource*)

shape = None

(string) The service shape name for this resource or None

subresources

Get a list of sub-resources.

Type list(:py:class'ResponseResource')

waiters

Get a list of waiters for this resource.

Type list(*Waiter*)

class boto3.resources.model.**ResponseResource** (*definition, resource_defs*)

A resource response to create after performing an action.

Parameters

- **definition** (*dict*) – The JSON definition
- **resource_defs** (*dict*) – All resources defined in the service

identifiers

A list of resource identifiers.

Type list(*Identifier*)

model

Get the resource model for the response resource.

Type *ResourceModel*

path = None

(string) The JMESPath search query or None

type = None

(string) The name of the response resource type

class boto3.resources.model.**Waiter** (*name, definition*)

An event waiter specification.

Parameters

- **name** (*string*) – Name of the waiter
- **definition** (*dict*) – The JSON definition

PREFIX = 'WaitUntil'**name** = None

(string) The name of this waiter

params

Get a list of auto-filled parameters for this request.

Type list(*Parameter*)**waiter_name** = None

(string) The name of the underlying event waiter

Request Parameters`boto3.resources.params.build_param_structure(params, target, value)`

This method provides a basic reverse JMESPath implementation that lets you go from a JMESPath-like string to a possibly deeply nested object. The `params` are mutated in-place, so subsequent calls can modify the same element by its index.

```
>>> build_param_structure(params, 'test[0]', 1)
>>> print(params)
{'test': [1]}

>>> build_param_structure(params, 'foo.bar[0].baz', 'hello world')
>>> print(params)
{'test': [1], 'foo': {'bar': [{'baz': 'hello, world'}]}}
```

`boto3.resources.params.create_request_parameters(parent, request_model,
 params=None)`

Handle request parameters that can be filled in from identifiers, resource data members or constants.

By passing `params`, you can invoke this method multiple times and build up a parameter dict over time, which is particularly useful for reverse JMESPath expressions that append to lists.

Parameters

- **parent** (*ServiceResource*) – The resource instance to which this action is attached.
- **request_model** (*Request*) – The action request model.
- **params** (*dict*) – If set, then add to this existing dict. It is both edited in-place and returned.

Return type dict**Returns** Pre-filled parameters to be sent to the request operation.`boto3.resources.params.get_data_member(parent, path)`

Get a data member from a parent using a JMESPath search query, loading the parent if required. If the parent cannot be loaded and no data is present then an exception is raised.

Parameters

- **parent** (*ServiceResource*) – The resource instance to which contains data we are interested in.
- **path** (*string*) – The JMESPath expression to query

Raises ResourceLoadException When no data is present and the resource cannot be loaded.**Returns** The queried data or None.

Response Handlers

class `boto3.resources.response.RawHandler` (*search_path*)

A raw action response handler. This passed through the response dictionary, optionally after performing a JMESPath search if one has been defined for the action.

Parameters *search_path* (*string*) – JMESPath expression to search in the response

Return type dict

Returns Service response

class `boto3.resources.response.ResourceHandler` (*search_path*, *factory*, *resource_defs*, *service_model*, *resource_model*, *operation_name=None*)

Creates a new resource or list of new resources from the low-level response based on the given response resource definition.

Parameters

- **search_path** (*string*) – JMESPath expression to search in the response
- **factory** (*ResourceFactory*) – The factory that created the resource class to which this action is attached.
- **resource_defs** (*dict*) – Service resource definitions.
- **service_model** (*botocore.model.ServiceModel*) – The Botocore service model
- **resource_model** (*ResponseResource*) – Response resource model.
- **operation_name** (*string*) – Name of the underlying service operation, if it exists.

Return type ServiceResource or list

Returns New resource instance(s).

handle_response_item (*resource_cls*, *parent*, *identifiers*, *resource_data*)

Handles the creation of a single response item by setting parameters and creating the appropriate resource instance.

Parameters

- **resource_cls** (*ServiceResource subclass*) – The resource class to instantiate.
- **parent** (*ServiceResource*) – The resource instance to which this action is attached.
- **identifiers** (*dict*) – Map of identifier names to value or values.
- **resource_data** (*dict or None*) – Data for resource attributes.

Return type ServiceResource

Returns New resource instance.

`boto3.resources.response.all_not_none` (*iterable*)

Return True if all elements of the iterable are not None (or if the iterable is empty). This is like the built-in `all`, except checks against None, so 0 and False are allowable values.

build_empty_response (*search_path*, *operation_name*, *service_model*)

Creates an appropriate empty response for the type that is expected, based on the service model's shape type. For example, a value that is normally a list would then return an empty list. A structure would return an empty dict, and a number would return None.

Parameters

- **search_path** (*string*) – JMESPath expression to search in the response
- **operation_name** (*string*) – Name of the underlying service operation.
- **service_model** (*botocore.model.ServiceModel*) – The Botocore service model

Return type dict, list, or None

Returns An appropriate empty value

build_identifiers (*identifiers*, *parent*, *params=None*, *raw_response=None*)

Builds a mapping of identifier names to values based on the identifier source location, type, and target. Identifier values may be scalars or lists depending on the source type and location.

Parameters

- **identifiers** (*list*) – List of `Parameter` definitions
- **parent** (*ServiceResource*) – The resource instance to which this action is attached.
- **params** (*dict*) – Request parameters sent to the service.
- **raw_response** (*dict*) – Low-level operation response.

Return type `list`**Returns** An ordered list of (name, value) identifier tuples.**Resource Actions**

```
class boto3.resources.action.BatchAction(action_model, factory=None, resource_defs=None,
                                         service_model=None)
```

An action which operates on a batch of items in a collection, typically a single page of results from the collection's underlying service operation call. For example, this allows you to delete up to 999 S3 objects in a single operation rather than calling `.delete()` on each one individually.

Parameters

- **action_model** (*Action*) – The action model.
- **factory** (*ResourceFactory*) – The factory that created the resource class to which this action is attached.
- **resource_defs** (*dict*) – Service resource definitions.
- **service_model** (*botocore.model.ServiceModel*) – The Botocore service model

```
class boto3.resources.action.ServiceAction(action_model, factory=None, re-
                                           source_defs=None, service_model=None)
```

A class representing a callable action on a resource, for example `sqs.get_queue_by_name(...)` or `s3.Bucket('foo').delete()`. The action may construct parameters from existing resource identifiers and may return either a raw response or a new resource instance.

Parameters

- **action_model** (*:py:class '~boto3.resources.model.Action'*) – The action model.
- **factory** (*ResourceFactory*) – The factory that created the resource class to which this action is attached.
- **resource_defs** (*dict*) – Service resource definitions.
- **service_model** (*botocore.model.ServiceModel*) – The Botocore service model

```
class boto3.resources.action.WaiterAction(waiter_model, waiter_resource_name)
```

A class representing a callable waiter action on a resource, for example `s3.Bucket('foo').wait_until_bucket_exists()`. The waiter action may construct parameters from existing resource identifiers.

Parameters

- **waiter_model** (*:py:class '~boto3.resources.model.Waiter'*) – The action waiter.
- **waiter_resource_name** (*string*) – The name of the waiter action for the resource. It usually begins with a `wait_until_`

Resource Base

```
class boto3.resources.base.ResourceMeta(service_name, identifiers=None, client=None,
                                         data=None, resource_model=None)
```

An object containing metadata about a resource.

client = `None`

(`BaseClient`) Low-level Botocore client

copy ()

Create a copy of this metadata object.

data = None
(dict) Loaded resource data attributes

identifiers = None
(list) List of identifier names

service_name = None
(string) The service name, e.g. 's3'

class boto3.resources.base.**ServiceResource** (*args, **kwargs)

A base class for resources.

Parameters **client** (*botocore.client*) – A low-level Botocore client instance

meta = None

Stores metadata about this resource instance, such as the `service_name`, the low-level `client` and any cached data from when the instance was hydrated. For example:

```
# Get a low-level client from a resource instance
client = resource.meta.client
response = client.operation(Param='foo')
```

```
# Print the resource instance's service short name
print(resource.meta.service_name)
```

See `ResourceMeta` for more information.

Resource Factory

class boto3.resources.factory.**ResourceFactory** (*emitter*)

A factory to create new `ServiceResource` classes from a `ResourceModel`. There are two types of lookups that can be done: one on the service itself (e.g. an SQS resource) and another on models contained within the service (e.g. an SQS Queue resource).

load_from_definition (*service_name, resource_name, model, resource_defs, service_model*)

Loads a resource from a model, creating a new `ServiceResource` subclass with the correct properties and methods, named based on the service and resource name, e.g. `EC2.Instance`.

Parameters

- **service_name** (*string*) – Name of the service to look up
- **resource_name** (*string*) – Name of the resource to look up. For services, this should match the `service_name`.
- **model** (*dict*) – The service or resource definition.
- **resource_defs** (*dict*) – The service's resource definitions, used to load sub-resources (e.g. `sqs.Queue`).
- **service_model** (*botocore.model.ServiceModel*) – The Botocore service model, required only if the resource shape contains members. This is used to expose lazy-loaded attributes on the resource.

Return type Subclass of `ServiceResource`

Returns The service or resource class.

Session Reference

class boto3.session.**Session** (*aws_access_key_id=None, aws_secret_access_key=None, aws_session_token=None, region_name=None, boto_core_session=None, profile_name=None*)

A session stores configuration state and allows you to create service clients and resources.

Parameters

- **aws_access_key_id** (*string*) – AWS access key ID
- **aws_secret_access_key** (*string*) – AWS secret access key
- **aws_session_token** (*string*) – AWS temporary session token
- **region_name** (*string*) – Default region when creating new connections
- **botocore_session** (*botocore.session.Session*) – Use this Botocore session instead of creating a new default one.
- **profile_name** (*string*) – The name of a profile to use. If not given, then the default profile is used.

client (*service_name*, *region_name=None*, *api_version=None*, *use_ssl=True*, *verify=None*, *endpoint_url=None*, *aws_access_key_id=None*, *aws_secret_access_key=None*, *aws_session_token=None*, *config=None*)

Create a low-level service client by name.

Parameters

- **service_name** (*string*) – The name of a service, e.g. 's3' or 'ec2'. You can get a list of available services via `get_available_services()`.
- **region_name** (*string*) – The name of the region associated with the client. A client is associated with a single region.
- **api_version** (*string*) – The API version to use. By default, botocore will use the latest API version when creating a client. You only need to specify this parameter if you want to use a previous API version of the client.
- **use_ssl** (*boolean*) – Whether or not to use SSL. By default, SSL is used. Note that not all services support non-ssl connections.
- **verify** (*boolean/string*) – Whether or not to verify SSL certificates. By default SSL certificates are verified. You can provide the following values:
 - `False` - do not validate SSL certificates. SSL will still be used (unless `use_ssl` is `False`), but SSL certificates will not be verified.
 - `path/to/cert/bundle.pem` - A filename of the CA cert bundle to uses. You can specify this argument if you want to use a different CA cert bundle than the one used by botocore.
- **endpoint_url** (*string*) – The complete URL to use for the constructed client. Normally, botocore will automatically construct the appropriate URL to use when communicating with a service. You can specify a complete URL (including the "http/https" scheme) to override this behavior. If this value is provided, then `use_ssl` is ignored.
- **aws_access_key_id** (*string*) – The access key to use when creating the client. This is entirely optional, and if not provided, the credentials configured for the session will automatically be used. You only need to provide this argument if you want to override the credentials used for this specific client.
- **aws_secret_access_key** (*string*) – The secret key to use when creating the client. Same semantics as `aws_access_key_id` above.
- **aws_session_token** (*string*) – The session token to use when creating the client. Same semantics as `aws_access_key_id` above.
- **config** (*botocore.client.Config*) – Advanced client configuration options. If `region_name` is specified in the client config, its value will take precedence over environment variables and configuration values, but not over a `region_name` value passed explicitly to the method.

Returns Service client instance

events

The event emitter for a session

get_available_resources ()

Get a list of available services that can be loaded as resource clients via `Session.resource()`.

Return type list

Returns List of service names

get_available_services()

Get a list of available services that can be loaded as low-level clients via `Session.client()`.

Return type list

Returns List of service names

profile_name

The read-only profile name.

resource (*service_name*, *region_name=None*, *api_version=None*, *use_ssl=True*, *verify=None*, *endpoint_url=None*, *aws_access_key_id=None*, *aws_secret_access_key=None*, *aws_session_token=None*)

Create a resource service client by name.

Parameters

- **service_name** (*string*) – The name of a service, e.g. ‘s3’ or ‘ec2’. You can get a list of available services via `get_available_resources()`.
- **region_name** (*string*) – The name of the region associated with the client. A client is associated with a single region.
- **api_version** (*string*) – The API version to use. By default, botocore will use the latest API version when creating a client. You only need to specify this parameter if you want to use a previous API version of the client.
- **use_ssl** (*boolean*) – Whether or not to use SSL. By default, SSL is used. Note that not all services support non-ssl connections.
- **verify** (*boolean/string*) – Whether or not to verify SSL certificates. By default SSL certificates are verified. You can provide the following values:
 - False - do not validate SSL certificates. SSL will still be used (unless `use_ssl` is False), but SSL certificates will not be verified.
 - path/to/cert/bundle.pem - A filename of the CA cert bundle to uses. You can specify this argument if you want to use a different CA cert bundle than the one used by botocore.
- **endpoint_url** (*string*) – The complete URL to use for the constructed client. Normally, botocore will automatically construct the appropriate URL to use when communicating with a service. You can specify a complete URL (including the “http/https” scheme) to override this behavior. If this value is provided, then `use_ssl` is ignored.
- **aws_access_key_id** (*string*) – The access key to use when creating the client. This is entirely optional, and if not provided, the credentials configured for the session will automatically be used. You only need to provide this argument if you want to override the credentials used for this specific client.
- **aws_secret_access_key** (*string*) – The secret key to use when creating the client. Same semantics as `aws_access_key_id` above.
- **aws_session_token** (*string*) – The session token to use when creating the client. Same semantics as `aws_access_key_id` above.

Returns Subclass of `ServiceResource`

3.3 Customizations

3.3.1 Customization References

DynamoDB Customization Reference

Valid DynamoDB Types

These are the valid item types to use with Boto3 Table Resource (`dynamodb.Table`) and DynamoDB:

Python Type	DynamoDB Type
string	String (S)
integer	Number (N)
decimal.Decimal	Number (N)
<code>boto3.dynamodb.types.Binary</code>	Binary (B)
boolean	Boolean (BOOL)
None	Null (NULL)
string set	String Set (SS)
integer set	Number Set (NS)
decimal.Decimal set	Number Set (NS)
<code>boto3.dynamodb.types.Binary</code> set	Binary Set (BS)
list	List (L)
dict	Map (M)

Custom Boto3 Types

class `boto3.dynamodb.types.Binary` (*value*)

A class for representing Binary in dynamodb

Especially for Python 2, use this class to explicitly specify binary data for item in DynamoDB. It is essentially a wrapper around binary. Unicode and Python 3 string types are not allowed.

DynamoDB Conditions

class `boto3.dynamodb.conditions.Key` (*name*)

begins_with (*value*)

Creates a condition where the attribute begins with the value.

Parameters *value* – The value that the attribute begins with.

between (*low_value*, *high_value*)

Creates a condition where the attribute is between the low value and the high value.

Parameters

- **low_value** – The value that the attribute is greater than.
- **high_value** – The value that the attribute is less than.

eq (*value*)

Creates a condition where the attribute is equal to the value.

Parameters *value* – The value that the attribute is equal to.

gt (*value*)

Creates a condition where the attribute is greater than the value.

Parameters *value* – The value that the attribute is greater than.

gte (*value*)

Creates a condition where the attribute is greater than or equal to the value.

Parameters *value* – The value that the attribute is greater than or equal to.

lt (*value*)

Creates a condition where the attribute is less than the value.

Parameters *value* – The value that the attribute is less than.

lte (*value*)

Creates a condition where the attribute is less than or equal to the value.

Parameters *value* – The value that the attribute is less than or equal to.

class `boto3.dynamodb.conditions.Attr` (*name*)

Represents an DynamoDB item's attribute.

attribute_type (*value*)

Creates a condition for the attribute type.

Parameters *value* – The type of the attribute.

begins_with (*value*)

Creates a condition where the attribute begins with the value.

Parameters *value* – The value that the attribute begins with.

between (*low_value*, *high_value*)

Creates a condition where the attribute is between the low value and the high value.

Parameters

- **low_value** – The value that the attribute is greater than.
- **high_value** – The value that the attribute is less than.

contains (*value*)

Creates a condition where the attribute contains the value.

Parameters *value* – The value the attribute contains.

eq (*value*)

Creates a condition where the attribute is equal to the value.

Parameters *value* – The value that the attribute is equal to.

exists ()

Creates a condition where the attribute exists.

gt (*value*)

Creates a condition where the attribute is greater than the value.

Parameters *value* – The value that the attribute is greater than.

gte (*value*)

Creates a condition where the attribute is greater than or equal to the value.

Parameters *value* – The value that the attribute is greater than or equal to.

is_in (*value*)

Creates a condition where the attribute is in the value,

Parameters *value* (*list*) – The value that the attribute is in.

lt (*value*)

Creates a condition where the attribute is less than the value.

Parameters *value* – The value that the attribute is less than.

lte (*value*)

Creates a condition where the attribute is less than or equal to the value.

Parameters *value* – The value that the attribute is less than or equal to.

ne (*value*)

Creates a condition where the attribute is not equal to the value

Parameters *value* – The value that the attribute is not equal to.

not_exists ()

Creates a condition where the attribute does not exist.

size ()

Creates a condition for the attribute size.

Note another AttributeBase method must be called on the returned size condition to be a valid DynamoDB condition.

S3 Customization Reference

S3 Command Injection

`boto3.s3.inject.bucket_download_file(self, Key, Filename, ExtraArgs=None, Callback=None, Config=None)`

Download an S3 object to a file.

Usage:

```
import boto3
s3 = boto3.resource('s3')
s3.Bucket('mybucket').download_file('hello.txt', '/tmp/hello.txt')
```

Similar behavior as `S3Transfer`'s `download_file()` method, except that parameters are capitalized. Detailed examples can be found at [S3Transfer's Usage](#).

`boto3.s3.inject.bucket_load(self, *args, **kwargs)`
Calls `s3.Client.list_buckets()` to update the attributes of the Bucket resource.

`boto3.s3.inject.bucket_upload_file(self, Filename, Key, ExtraArgs=None, Callback=None, Config=None)`

Upload a file to an S3 object.

Usage:

```
import boto3
s3 = boto3.resource('s3')
s3.Bucket('mybucket').upload_file('/tmp/hello.txt', 'hello.txt')
```

Similar behavior as `S3Transfer`'s `upload_file()` method, except that parameters are capitalized. Detailed examples can be found at [S3Transfer's Usage](#).

`boto3.s3.inject.download_file(self, Bucket, Key, Filename, ExtraArgs=None, Callback=None, Config=None)`

Download an S3 object to a file.

Usage:

```
import boto3
s3 = boto3.resource('s3')
s3.meta.client.download_file('mybucket', 'hello.txt', '/tmp/hello.txt')
```

Similar behavior as `S3Transfer`'s `download_file()` method, except that parameters are capitalized. Detailed examples can be found at [S3Transfer's Usage](#).

`boto3.s3.inject.inject_bucket_methods(class_attributes, **kwargs)`

`boto3.s3.inject.inject_object_methods(class_attributes, **kwargs)`

`boto3.s3.inject.inject_s3_transfer_methods(class_attributes, **kwargs)`

`boto3.s3.inject.object_download_file(self, Filename, ExtraArgs=None, Callback=None, Config=None)`

Download an S3 object to a file.

Usage:

```
import boto3
s3 = boto3.resource('s3')
s3.Object('mybucket', 'hello.txt').download_file('/tmp/hello.txt')
```

Similar behavior as S3Transfer's `download_file()` method, except that parameters are capitalized. Detailed examples can be found at *S3Transfer's Usage*.

```
boto3.s3.inject.object_upload_file(self, Filename, ExtraArgs=None, Callback=None, Config=None)
```

Upload a file to an S3 object.

Usage:

```
import boto3
s3 = boto3.resource('s3')
s3.Object('mybucket', 'hello.txt').upload_file('/tmp/hello.txt')
```

Similar behavior as S3Transfer's `upload_file()` method, except that parameters are capitalized. Detailed examples can be found at *S3Transfer's Usage*.

```
boto3.s3.inject.upload_file(self, Filename, Bucket, Key, ExtraArgs=None, Callback=None, Config=None)
```

Upload a file to an S3 object.

Usage:

```
import boto3
s3 = boto3.resource('s3')
s3.meta.client.upload_file('/tmp/hello.txt', 'mybucket', 'hello.txt')
```

Similar behavior as S3Transfer's `upload_file()` method, except that parameters are capitalized. Detailed examples can be found at *S3Transfer's Usage*.

S3 Transfers

Abstractions over S3's upload/download operations.

This module provides high level abstractions for efficient uploads/downloads. It handles several things for the user:

- Automatically switching to multipart transfers when a file is over a specific size threshold
- Uploading/downloading a file in parallel
- Throttling based on max bandwidth
- Progress callbacks to monitor transfers
- Retries. While botocore handles retries for streaming uploads, it is not possible for it to handle retries for streaming downloads. This module handles retries for both cases so you don't need to implement any retry logic yourself.

This module has a reasonable set of defaults. It also allows you to configure many aspects of the transfer process including:

- Multipart threshold size
- Max parallel downloads
- Max bandwidth
- Socket timeouts

- Retry amounts

There is no support for s3->s3 multipart copies at this time.

Usage The simplest way to use this module is:

```
client = boto3.client('s3', 'us-west-2')
transfer = S3Transfer(client)
# Upload /tmp/myfile to s3://bucket/key
transfer.upload_file('/tmp/myfile', 'bucket', 'key')

# Download s3://bucket/key to /tmp/myfile
transfer.download_file('bucket', 'key', '/tmp/myfile')
```

The `upload_file` and `download_file` methods also accept `**kwargs`, which will be forwarded through to the corresponding client operation. Here are a few examples using `upload_file`:

```
# Making the object public
transfer.upload_file('/tmp/myfile', 'bucket', 'key',
                    extra_args={'ACL': 'public-read'})

# Setting metadata
transfer.upload_file('/tmp/myfile', 'bucket', 'key',
                    extra_args={'Metadata': {'a': 'b', 'c': 'd'}})

# Setting content type
transfer.upload_file('/tmp/myfile.json', 'bucket', 'key',
                    extra_args={'ContentType': "application/json"})
```

The `S3Transfer` class also supports progress callbacks so you can provide transfer progress to users. Both the `upload_file` and `download_file` methods take an optional callback parameter. Here's an example of how to print a simple progress percentage to the user:

```
class ProgressPercentage(object):
    def __init__(self, filename):
        self._filename = filename
        self._size = float(os.path.getsize(filename))
        self._seen_so_far = 0
        self._lock = threading.Lock()

    def __call__(self, bytes_amount):
        # To simplify we'll assume this is hooked up
        # to a single filename.
        with self._lock:
            self._seen_so_far += bytes_amount
            percentage = (self._seen_so_far / self._size) * 100
            sys.stdout.write(
                "\r%s / %s (%.2f%%)" % (self._filename, self._seen_so_far,
                                         self._size, percentage))
            sys.stdout.flush()

transfer = S3Transfer(boto3.client('s3', 'us-west-2')) # Upload /tmp/myfile to s3://bucket/key and print upload
progress. transfer.upload_file('/tmp/myfile', 'bucket', 'key',
                               callback=ProgressPercentage('/tmp/myfile'))
```

You can also provide a `TransferConfig` object to the `S3Transfer` object that gives you more fine grained control over the transfer. For example:

```
client = boto3.client('s3', 'us-west-2')
config = TransferConfig(
    multipart_threshold=8 * 1024 * 1024,
    max_concurrency=10,
    num_download_attempts=10,
)
transfer = S3Transfer(client, config)
transfer.upload_file('/tmp/foo', 'bucket', 'key')
```

```
class boto3.s3.transfer.MultipartDownloader (client, config, osutil, executor_cls=<class 'concurrent.futures.thread.ThreadPoolExecutor'>)
```

```
    download_file (bucket, key, filename, object_size, extra_args, callback=None)
```

```
class boto3.s3.transfer.MultipartUploader (client, config, osutil, executor_cls=<class 'concurrent.futures.thread.ThreadPoolExecutor'>)
```

```
    UPLOAD_PART_ARGS = ['SSECustomerKey', 'SSECustomerAlgorithm', 'SSECustomerKeyMD5', 'RequestPayer']
```

```
    upload_file (filename, bucket, key, callback, extra_args)
```

```
class boto3.s3.transfer.OSUtils
```

```
    get_file_size (filename)
```

```
    open (filename, mode)
```

```
    open_file_chunk_reader (filename, start_byte, size, callback)
```

```
    remove_file (filename)
```

Remove a file, noop if file does not exist.

```
    rename_file (current_filename, new_filename)
```

```
exception boto3.s3.transfer.QueueShutdownError
```

```
class boto3.s3.transfer.ReadFileChunk (fileobj, start_byte, chunk_size, full_file_size, callback=None, enable_callback=True)
```

Given a file object shown below:

```
|-----| 0 || full_file_size
|---chunk_size---
```

start_byte

Parameters

- **fileobj** (*file*) – File like object
- **start_byte** (*int*) – The first byte from which to start reading.
- **chunk_size** (*int*) – The max chunk size to read. Trying to read past the end of the chunk size will behave like you've reached the end of the file.
- **full_file_size** (*int*) – The entire content length associated with `fileobj`.
- **callback** (*function(amount_read)*) – Called whenever data is read from this object.

```
close ()
```

```
disable_callback ()
```

```
enable_callback ()
```

classmethod `from_filename(filename, start_byte, chunk_size, callback=None, enable_callback=True)`

Convenience factory function to create from a filename.

Parameters

- **start_byte** (*int*) – The first byte from which to start reading.
- **chunk_size** (*int*) – The max chunk size to read. Trying to read past the end of the chunk size will behave like you’ve reached the end of the file.
- **full_file_size** (*int*) – The entire content length associated with `fileobj`.
- **callback** (*function(amount_read)*) – Called whenever data is read from this object.
- **enable_callback** (*bool*) – Indicate whether to invoke callback during `read()` calls.

Return type `ReadFileChunk`

Returns A new instance of `ReadFileChunk`

read (*amount=None*)

seek (*where*)

tell ()

class `boto3.s3.transfer.S3Transfer(client, config=None, osutil=None)`

ALLOWED_DOWNLOAD_ARGS = ['VersionId', 'SSECustomerAlgorithm', 'SSECustomerKey', 'SSECustomerKeyMD5', 'R

ALLOWED_UPLOAD_ARGS = ['ACL', 'CacheControl', 'ContentDisposition', 'ContentEncoding', 'ContentLanguage', 'Co

download_file (*bucket, key, filename, extra_args=None, callback=None*)

Download an S3 object to a file.

Variants have also been injected into S3 client, Bucket and Object. You don’t have to use `S3Transfer.download_file()` directly.

upload_file (*filename, bucket, key, callback=None, extra_args=None*)

Upload a file to an S3 object.

Variants have also been injected into S3 client, Bucket and Object. You don’t have to use `S3Transfer.upload_file()` directly.

class `boto3.s3.transfer.ShutdownQueue(maxsize=0)`

A queue implementation that can be shutdown.

Shutting down a queue means that this class adds a `trigger_shutdown` method that will trigger all subsequent calls to `put()` to fail with a `QueueShutdownError`.

It purposefully deviates from `queue.Queue`, and is *not* meant to be a drop in replacement for `queue.Queue`.

put (*item*)

trigger_shutdown ()

class `boto3.s3.transfer.StreamReaderProgress(stream, callback=None)`

Wrapper for a read only stream that adds progress callbacks.

read (**args, **kwargs*)

class `boto3.s3.transfer.TransferConfig(multipart_threshold=8388608, max_concurrency=10, multipart_chunksize=8388608, num_download_attempts=5, max_io_queue=100)`

`boto3.s3.transfer.disable_upload_callbacks(request, operation_name, **kwargs)`

`boto3.s3.transfer.enable_upload_callbacks(request, operation_name, **kwargs)`

`boto3.s3.transfer.random_file_extension(num_digits=8)`

Indices and tables

- *genindex*
- *modindex*
- *search*

b

- `boto3`, [4634](#)
- `boto3.resources.action`, [4644](#)
- `boto3.resources.base`, [4644](#)
- `boto3.resources.collection`, [4635](#)
- `boto3.resources.factory`, [4645](#)
- `boto3.resources.model`, [4638](#)
- `boto3.resources.params`, [4642](#)
- `boto3.resources.response`, [4643](#)
- `boto3.s3.inject`, [4650](#)
- `boto3.s3.transfer`, [4651](#)
- `boto3.session`, [4645](#)

A

- abort() (Glacier.MultipartUpload method), 2690
- abort() (S3.MultipartUpload method), 4080
- abort_environment_update() (ElasticBeanstalk.Client method), 2041
- abort_multipart_upload() (Glacier.Client method), 2635
- abort_multipart_upload() (S3.Client method), 3955
- abort_vault_lock() (Glacier.Client method), 2635
- accept() (EC2.VpcPeeringConnection method), 1788
- accept_ranges (S3.Object attribute), 4089
- accept_vpc_peering_connection() (EC2.Client method), 1104
- accepted_vpc_peering_connections (EC2.Vpc attribute), 1768
- accepter_vpc (EC2.VpcPeeringConnection attribute), 1787
- accepter_vpc_info (EC2.VpcPeeringConnection attribute), 1787
- access_key_id (IAM.AccessKey attribute), 2895
- access_key_id (IAM.AccessKeyPair attribute), 2897
- access_keys (IAM.CurrentUser attribute), 2905
- access_keys (IAM.User attribute), 2960
- AccessKey() (IAM.ServiceResource method), 2880
- AccessKey() (IAM.User method), 2959
- AccessKeyPair() (IAM.ServiceResource method), 2880
- Account() (Glacier.ServiceResource method), 2676
- Account() (Glacier.Vault method), 2704
- account_id (Glacier.Archive attribute), 2681
- account_id (Glacier.Job attribute), 2685
- account_id (Glacier.MultipartUpload attribute), 2690
- account_id (Glacier.Notification attribute), 2696
- account_id (Glacier.Vault attribute), 2699
- AccountPasswordPolicy() (IAM.ServiceResource method), 2881
- AccountSummary() (IAM.ServiceResource method), 2881
- acknowledge_job() (CodePipeline.Client method), 508
- acknowledge_third_party_job() (CodePipeline.Client method), 508
- Acl() (S3.Bucket method), 4046
- Acl() (S3.Object method), 4101
- Acl() (S3.ObjectSummary method), 4118
- Action (class in boto3.resources.model), 4638
- action (Glacier.Job attribute), 2685
- actions (boto3.resources.model.ResourceModel attribute), 4640
- activate() (IAM.AccessKey method), 2895
- activate() (IAM.AccessKeyPair method), 2897
- activate() (IAM.SigningCertificate method), 2952
- activate_gateway() (StorageGateway.Client method), 4273
- activate_pipeline() (DataPipeline.Client method), 610
- add_attachments_to_set() (Support.Client method), 4347
- add_cache() (StorageGateway.Client method), 4274
- add_client_id_to_open_id_connect_provider() (IAM.Client method), 2715
- add_communication_to_case() (Support.Client method), 4349
- add_group() (IAM.User method), 2955
- add_instance_groups() (EMR.Client method), 2581
- add_job_flow_steps() (EMR.Client method), 2583
- add_permission() (Lambda.Client method), 3025
- add_permission() (SNS.Client method), 4167
- add_permission() (SNS.Topic method), 4208
- add_permission() (SQS.Client method), 4215
- add_permission() (SQS.Queue method), 4245
- add_role() (IAM.InstanceProfile method), 2920
- add_role_to_instance_profile() (IAM.Client method), 2715
- add_source_identifier_to_subscription() (RDS.Client method), 3279
- add_tags() (DataPipeline.Client method), 611
- add_tags() (ElasticLoadBalancing.Client method), 2538
- add_tags() (EMR.Client method), 2585
- add_tags_to_on_premises_instances() (CodeDeploy.Client method), 463
- add_tags_to_resource() (ElastiCache.Client method), 1896
- add_tags_to_resource() (RDS.Client method), 3281
- add_tags_to_resource() (StorageGateway.Client method), 4275

- `add_tags_to_stream()` (Kinesis.Client method), 2980
- `add_tags_to_vault()` (Glacier.Client method), 2636
- `add_upload_buffer()` (StorageGateway.Client method), 4276
- `add_user()` (IAM.Group method), 2910
- `add_user_to_group()` (IAM.Client method), 2716
- `add_working_storage()` (StorageGateway.Client method), 4277
- `agent_version` (OpsWorks.Stack attribute), 3269
- `all()` (boto3.resources.collection.CollectionManager method), 4635
- `all()` (boto3.resources.collection.ResourceCollection method), 4637
- `all()` (CloudFormation.ServiceResource method), 155
- `all()` (CloudFormation.Stack method), 163, 164
- `all()` (DynamoDB.ServiceResource method), 997
- `all()` (EC2.Instance method), 1640
- `all()` (EC2.PlacementGroup method), 1679
- `all()` (EC2.RouteTable method), 1692
- `all()` (EC2.ServiceResource method), 1556, 1557, 1560, 1573, 1575, 1576, 1578, 1581, 1583, 1585, 1587, 1590, 1592, 1594, 1596
- `all()` (EC2.Subnet method), 1730, 1739
- `all()` (EC2.Volume method), 1755
- `all()` (EC2.Vpc method), 1768, 1769, 1778–1780, 1782–1785
- `all()` (Glacier.Account method), 2680
- `all()` (Glacier.ServiceResource method), 2678
- `all()` (Glacier.Vault method), 2705–2707, 2709–2711
- `all()` (IAM.CurrentUser method), 2905–2907
- `all()` (IAM.Group method), 2913–2915
- `all()` (IAM.Policy method), 2931, 2933–2935
- `all()` (IAM.Role method), 2941, 2943, 2944
- `all()` (IAM.ServiceResource method), 2884, 2885, 2887–2890, 2892, 2893
- `all()` (IAM.User method), 2960, 2961, 2963–2966
- `all()` (OpsWorks.ServiceResource method), 3263
- `all()` (OpsWorks.Stack method), 3275
- `all()` (S3.Bucket method), 4048, 4049, 4051
- `all()` (S3.MultipartUpload method), 4082
- `all()` (S3.ServiceResource method), 4039
- `all()` (SNS.PlatformApplication method), 4200
- `all()` (SNS.ServiceResource method), 4194–4196
- `all()` (SNS.Topic method), 4213
- `all()` (SQS.Queue method), 4259
- `all()` (SQS.ServiceResource method), 4240
- `all_not_none()` (in module `boto3.resources.response`), 4643
- `allocate_address()` (EC2.Client method), 1105
- `allocate_connection_on_interconnect()` (DirectConnect.Client method), 700
- `allocate_private_virtual_interface()` (DirectConnect.Client method), 702
- `allocate_public_virtual_interface()` (DirectConnect.Client method), 705
- `allow_users_to_change_password` (IAM.AccountPasswordPolicy attribute), 2899
- `ALLOWED_DOWNLOAD_ARGS` (boto3.s3.transfer.S3Transfer attribute), 4654
- `ALLOWED_UPLOAD_ARGS` (boto3.s3.transfer.S3Transfer attribute), 4654
- `ami_launch_index` (EC2.Instance attribute), 1612
- `apply_pending_maintenance_action()` (RDS.Client method), 3282
- `apply_security_groups_to_load_balancer()` (ElasticLoadBalancing.Client method), 2539
- `apps_count` (OpsWorks.StackSummary attribute), 3277
- `architecture` (EC2.Image attribute), 1602
- `architecture` (EC2.Instance attribute), 1612
- `Archive()` (Glacier.ServiceResource method), 2676
- `Archive()` (Glacier.Vault method), 2704
- `archive_description` (Glacier.MultipartUpload attribute), 2690
- `archive_id` (Glacier.Job attribute), 2685
- `archive_sha256_tree_hash` (Glacier.Job attribute), 2685
- `archive_size_in_bytes` (Glacier.Job attribute), 2685
- `arn` (IAM.CurrentUser attribute), 2904
- `arn` (IAM.Group attribute), 2909
- `arn` (IAM.InstanceProfile attribute), 2919
- `arn` (IAM.Policy attribute), 2927
- `arn` (IAM.PolicyVersion attribute), 2937
- `arn` (IAM.Role attribute), 2939
- `arn` (IAM.SamlProvider attribute), 2948
- `arn` (IAM.User attribute), 2954
- `arn` (OpsWorks.Stack attribute), 3269
- `arn` (OpsWorks.StackSummary attribute), 3277
- `arn` (SNS.PlatformApplication attribute), 4198
- `arn` (SNS.PlatformEndpoint attribute), 4202
- `arn` (SNS.Subscription attribute), 4206
- `arn` (SNS.Topic attribute), 4208
- `assign_instance()` (OpsWorks.Client method), 3159
- `assign_private_ip_addresses()` (EC2.Client method), 1106
- `assign_private_ip_addresses()` (EC2.NetworkInterface method), 1671
- `assign_volume()` (OpsWorks.Client method), 3159
- `associate()` (IAM.MfaDevice method), 2925
- `associate_address()` (EC2.Client method), 1107
- `associate_dhcp_options()` (EC2.Client method), 1108
- `associate_dhcp_options()` (EC2.Vpc method), 1758
- `associate_elastic_ip()` (OpsWorks.Client method), 3160
- `associate_route_table()` (EC2.Client method), 1109
- `associate_vpc_with_hosted_zone()` (Route53.Client method), 3812
- `associate_with_subnet()` (EC2.RouteTable method), 1690
- `associate_with_vpc()` (EC2.DhcpOptions method), 1599
- `association` (EC2.NetworkInterface attribute), 1669

- associations (EC2.NetworkAcl attribute), 1663
 - associations (EC2.RouteTable attribute), 1692
 - associations_attribute (EC2.RouteTable attribute), 1689
 - assume_role() (STS.Client method), 4331
 - assume_role_policy_document (IAM.Role attribute), 2939
 - assume_role_with_saml() (STS.Client method), 4334
 - assume_role_with_web_identity() (STS.Client method), 4338
 - AssumeRolePolicy() (IAM.Role method), 2941
 - AssumeRolePolicy() (IAM.ServiceResource method), 2881
 - attach() (EC2.NetworkInterface method), 1672
 - attach_classic_link_instance() (EC2.Vpc method), 1759
 - attach_classic_link_vpc() (EC2.Client method), 1109
 - attach_classic_link_vpc() (EC2.Instance method), 1616
 - attach_elastic_load_balancer() (OpsWorks.Client method), 3160
 - attach_group() (IAM.Policy method), 2928
 - attach_group_policy() (IAM.Client method), 2716
 - attach_instances() (AutoScaling.Client method), 41
 - attach_internet_gateway() (EC2.Client method), 1110
 - attach_internet_gateway() (EC2.Vpc method), 1759
 - attach_load_balancer_to_subnets() (ElasticLoadBalancing.Client method), 2539
 - attach_load_balancers() (AutoScaling.Client method), 41
 - attach_network_interface() (EC2.Client method), 1111
 - attach_policy() (IAM.Group method), 2910
 - attach_policy() (IAM.Role method), 2939
 - attach_policy() (IAM.User method), 2955
 - attach_role() (IAM.Policy method), 2928
 - attach_role_policy() (IAM.Client method), 2716
 - attach_to_instance() (EC2.Volume method), 1744
 - attach_to_vpc() (EC2.InternetGateway method), 1658
 - attach_user() (IAM.Policy method), 2929
 - attach_user_policy() (IAM.Client method), 2717
 - attach_volume() (EC2.Client method), 1111
 - attach_volume() (EC2.Instance method), 1616
 - attach_vpn_gateway() (EC2.Client method), 1113
 - attached_groups (IAM.Policy attribute), 2931
 - attached_policies (IAM.Group attribute), 2913
 - attached_policies (IAM.Role attribute), 2941
 - attached_policies (IAM.User attribute), 2961
 - attached_roles (IAM.Policy attribute), 2932
 - attached_users (IAM.Policy attribute), 2934
 - attachment (EC2.NetworkInterface attribute), 1669
 - attachment_count (IAM.Policy attribute), 2927
 - attachments (EC2.InternetGateway attribute), 1658
 - attachments (EC2.Volume attribute), 1743
 - Attr (class in boto3.dynamodb.conditions), 4649
 - attribute_definitions (DynamoDB.Table attribute), 999
 - attribute_type() (boto3.dynamodb.conditions.Attr method), 4649
 - attributes (OpsWorks.Layer attribute), 3265
 - attributes (OpsWorks.Stack attribute), 3269
 - attributes (SNS.PlatformApplication attribute), 4198
 - attributes (SNS.PlatformEndpoint attribute), 4202
 - attributes (SNS.Subscription attribute), 4206
 - attributes (SNS.Topic attribute), 4208
 - attributes (SQS.Message attribute), 4242
 - attributes (SQS.Queue attribute), 4245
 - authorize_cache_security_group_ingress() (ElasticCache.Client method), 1897
 - authorize_cluster_security_group_ingress() (Redshift.Client method), 3591
 - authorize_db_security_group_ingress() (RDS.Client method), 3283
 - authorize_egress() (EC2.SecurityGroup method), 1698
 - authorize_ingress() (EC2.SecurityGroup method), 1700
 - authorize_security_group_egress() (EC2.Client method), 1114
 - authorize_security_group_ingress() (EC2.Client method), 1116
 - authorize_snapshot_access() (Redshift.Client method), 3593
 - auto_assign_elastic_ips (OpsWorks.Layer attribute), 3265
 - auto_assign_public_ips (OpsWorks.Layer attribute), 3265
 - AutoScaling.Client (built-in class), 39
 - AutoScaling.Paginator.DescribeAutoScalingGroups (built-in class), 98
 - AutoScaling.Paginator.DescribeAutoScalingInstances (built-in class), 102
 - AutoScaling.Paginator.DescribeLaunchConfigurations (built-in class), 104
 - AutoScaling.Paginator.DescribeNotificationConfigurations (built-in class), 108
 - AutoScaling.Paginator.DescribePolicies (built-in class), 109
 - AutoScaling.Paginator.DescribeScalingActivities (built-in class), 113
 - AutoScaling.Paginator.DescribeScheduledActions (built-in class), 115
 - AutoScaling.Paginator.DescribeTags (built-in class), 117
 - availability_zone (EC2.NetworkInterface attribute), 1669
 - availability_zone (EC2.Subnet attribute), 1720
 - availability_zone (EC2.Volume attribute), 1743
 - available_ip_address_count (EC2.Subnet attribute), 1721
- ## B
- base_32_string_seed (IAM.VirtualMfaDevice attribute), 2970
 - batch_actions (boto3.resources.model.Collection attribute), 4639
 - batch_actions (boto3.resources.model.ResourceModel attribute), 4640
 - batch_delete_attributes() (SimpleDB.Client method), 4129

`batch_get_applications()` (CodeDeploy.Client method), 463
`batch_get_deployments()` (CodeDeploy.Client method), 464
`batch_get_item()` (DynamoDB.Client method), 755
`batch_get_item()` (DynamoDB.ServiceResource method), 976
`batch_get_on_premises_instances()` (CodeDeploy.Client method), 468
`batch_get_repositories()` (CodeCommit.Client method), 452
`batch_put_attributes()` (SimpleDB.Client method), 4130
`batch_write_item()` (DynamoDB.Client method), 767
`batch_write_item()` (DynamoDB.ServiceResource method), 984
`batch_writer()` (DynamoDB.Table method), 1002
`BatchAction` (class in boto3.resources.action), 4644
`begins_with()` (boto3.dynamodb.conditions.Attr method), 4649
`begins_with()` (boto3.dynamodb.conditions.Key method), 4648
`between()` (boto3.dynamodb.conditions.Attr method), 4649
`between()` (boto3.dynamodb.conditions.Key method), 4648
`Binary` (class in boto3.dynamodb.types), 4648
`block_device_mappings` (EC2.Image attribute), 1602
`block_device_mappings` (EC2.Instance attribute), 1612
`body` (SQS.Message attribute), 4242
`boto3` (module), 4634
`boto3.resources.action` (module), 4644
`boto3.resources.base` (module), 4644
`boto3.resources.collection` (module), 4635
`boto3.resources.factory` (module), 4645
`boto3.resources.model` (module), 4638
`boto3.resources.params` (module), 4642
`boto3.resources.response` (module), 4643
`boto3.s3.inject` (module), 4650
`boto3.s3.transfer` (module), 4651
`boto3.session` (module), 4645
`Bucket()` (S3.BucketAcl method), 4056
`Bucket()` (S3.BucketCors method), 4058
`Bucket()` (S3.BucketLifecycle method), 4061
`Bucket()` (S3.BucketLogging method), 4064
`Bucket()` (S3.BucketNotification method), 4068
`Bucket()` (S3.BucketPolicy method), 4069
`Bucket()` (S3.BucketRequestPayment method), 4071
`Bucket()` (S3.BucketTagging method), 4073
`Bucket()` (S3.BucketVersioning method), 4075
`Bucket()` (S3.BucketWebsite method), 4079
`Bucket()` (S3.Object method), 4101
`Bucket()` (S3.ObjectSummary method), 4118
`Bucket()` (S3.ServiceResource method), 4036
`bucket_download_file()` (in module boto3.s3.inject), 4650
`bucket_load()` (in module boto3.s3.inject), 4650
`bucket_name` (S3.BucketAcl attribute), 4054
`bucket_name` (S3.BucketCors attribute), 4057
`bucket_name` (S3.BucketLifecycle attribute), 4059
`bucket_name` (S3.BucketLogging attribute), 4062
`bucket_name` (S3.BucketNotification attribute), 4064
`bucket_name` (S3.BucketPolicy attribute), 4068
`bucket_name` (S3.BucketRequestPayment attribute), 4070
`bucket_name` (S3.BucketTagging attribute), 4072
`bucket_name` (S3.BucketVersioning attribute), 4074
`bucket_name` (S3.BucketWebsite attribute), 4076
`bucket_name` (S3.MultipartUpload attribute), 4080
`bucket_name` (S3.MultipartUploadPart attribute), 4083
`bucket_name` (S3.Object attribute), 4089
`bucket_name` (S3.ObjectAcl attribute), 4104
`bucket_name` (S3.ObjectSummary attribute), 4107
`bucket_name` (S3.ObjectVersion attribute), 4122
`bucket_upload_file()` (in module boto3.s3.inject), 4650
`BucketAcl()` (S3.ServiceResource method), 4036
`BucketCors()` (S3.ServiceResource method), 4036
`BucketLifecycle()` (S3.ServiceResource method), 4036
`BucketLogging()` (S3.ServiceResource method), 4037
`BucketNotification()` (S3.ServiceResource method), 4037
`BucketPolicy()` (S3.ServiceResource method), 4037
`BucketRequestPayment()` (S3.ServiceResource method), 4037
`buckets` (S3.ServiceResource attribute), 4039
`BucketTagging()` (S3.ServiceResource method), 4037
`BucketVersioning()` (S3.ServiceResource method), 4037
`BucketWebsite()` (S3.ServiceResource method), 4038
`build_empty_response()` (in module boto3.resources.response), 4643
`build_identifiers()` (in module boto3.resources.response), 4643
`build_param_structure()` (in module boto3.resources.params), 4642
`build_suggesters()` (CloudSearch.Client method), 332
`bulk_publish()` (CognitoSync.Client method), 577
`bundle_instance()` (EC2.Client method), 1119

C

`cache_control` (S3.Object attribute), 4089
`can_paginate()` (AutoScaling.Client method), 42
`can_paginate()` (CloudFormation.Client method), 119
`can_paginate()` (CloudFront.Client method), 169
`can_paginate()` (CloudHSM.Client method), 318
`can_paginate()` (CloudSearch.Client method), 333
`can_paginate()` (CloudSearchDomain.Client method), 405
`can_paginate()` (CloudTrail.Client method), 416
`can_paginate()` (CloudWatch.Client method), 426
`can_paginate()` (CloudWatchLogs.Client method), 3053
`can_paginate()` (CodeCommit.Client method), 454
`can_paginate()` (CodeDeploy.Client method), 469

- `can_paginate()` (CodePipeline.Client method), 509
- `can_paginate()` (CognitoIdentity.Client method), 560
- `can_paginate()` (CognitoSync.Client method), 577
- `can_paginate()` (ConfigService.Client method), 595
- `can_paginate()` (DataPipeline.Client method), 611
- `can_paginate()` (DeviceFarm.Client method), 642
- `can_paginate()` (DirectConnect.Client method), 708
- `can_paginate()` (DirectoryService.Client method), 736
- `can_paginate()` (DynamoDB.Client method), 780
- `can_paginate()` (DynamoDBStreams.Client method), 1087
- `can_paginate()` (EC2.Client method), 1121
- `can_paginate()` (ECS.Client method), 1792
- `can_paginate()` (EFS.Client method), 1882
- `can_paginate()` (ElastiCache.Client method), 1899
- `can_paginate()` (ElasticBeanstalk.Client method), 2041
- `can_paginate()` (ElasticLoadBalancing.Client method), 2540
- `can_paginate()` (ElasticTranscoder.Client method), 2101
- `can_paginate()` (EMR.Client method), 2585
- `can_paginate()` (Glacier.Client method), 2637
- `can_paginate()` (IAM.Client method), 2717
- `can_paginate()` (ImportExport.Client method), 2972
- `can_paginate()` (Kinesis.Client method), 2981
- `can_paginate()` (KMS.Client method), 3002
- `can_paginate()` (Lambda.Client method), 3026
- `can_paginate()` (MachineLearning.Client method), 3087
- `can_paginate()` (OpsWorks.Client method), 3160
- `can_paginate()` (RDS.Client method), 3286
- `can_paginate()` (Redshift.Client method), 3597
- `can_paginate()` (Route53.Client method), 3813
- `can_paginate()` (Route53Domains.Client method), 3877
- `can_paginate()` (S3.Client method), 3955
- `can_paginate()` (SES.Client method), 4142
- `can_paginate()` (SimpleDB.Client method), 4131
- `can_paginate()` (SNS.Client method), 4167
- `can_paginate()` (SQS.Client method), 4216
- `can_paginate()` (SSM.Client method), 4260
- `can_paginate()` (StorageGateway.Client method), 4278
- `can_paginate()` (STS.Client method), 4341
- `can_paginate()` (Support.Client method), 4349
- `can_paginate()` (SWF.Client method), 4376
- `can_paginate()` (WorkSpaces.Client method), 4621
- `cancel_archival()` (StorageGateway.Client method), 4278
- `cancel_bundle_task()` (EC2.Client method), 1121
- `cancel_conversion_task()` (EC2.Client method), 1123
- `cancel_export_task()` (CloudWatchLogs.Client method), 3053
- `cancel_export_task()` (EC2.Client method), 1124
- `cancel_import_task()` (EC2.Client method), 1124
- `cancel_job()` (ElasticTranscoder.Client method), 2101
- `cancel_job()` (ImportExport.Client method), 2972
- `cancel_reserved_instances_listing()` (EC2.Client method), 1125
- `cancel_retrieval()` (StorageGateway.Client method), 4278
- `cancel_spot_fleet_requests()` (EC2.Client method), 1127
- `cancel_spot_instance_requests()` (EC2.Client method), 1129
- `cancel_update()` (CloudFormation.Stack method), 160
- `cancel_update_stack()` (CloudFormation.Client method), 119
- `capabilities` (CloudFormation.Stack attribute), 159
- `certificate_body` (IAM.ServerCertificate attribute), 2950
- `certificate_body` (IAM.SigningCertificate attribute), 2952
- `certificate_chain` (IAM.ServerCertificate attribute), 2950
- `certificate_id` (IAM.SigningCertificate attribute), 2952
- `change_message_visibility()` (SQS.Client method), 4216
- `change_message_visibility_batch()` (SQS.Client method), 4217
- `change_message_visibility_batch()` (SQS.Queue method), 4246
- `change_password()` (IAM.Client method), 2718
- `change_password()` (IAM.ServiceResource method), 2873
- `change_resource_record_sets()` (Route53.Client method), 3813
- `change_tags_for_resource()` (Route53.Client method), 3818
- `change_visibility()` (SQS.Message method), 4242
- `check_dns_availability()` (ElasticBeanstalk.Client method), 2041
- `check_domain_availability()` (Route53Domains.Client method), 3877
- `chef_configuration` (OpsWorks.Stack attribute), 3269
- `cidr_block` (EC2.Subnet attribute), 1721
- `cidr_block` (EC2.Vpc attribute), 1757
- `client` (boto3.resources.base.ResourceMeta attribute), 4644
- `client()` (boto3.session.Session method), 4646
- `client()` (in module boto3), 4634
- `client_token` (EC2.Instance attribute), 1612
- `clone_stack()` (OpsWorks.Client method), 3161
- `close()` (boto3.s3.transfer.ReadFileChunk method), 4653
- `CloudFormation.Client` (built-in class), 119
- `CloudFormation.Event` (built-in class), 156
- `CloudFormation.Paginator.DescribeStackEvents` (built-in class), 143
- `CloudFormation.Paginator.DescribeStacks` (built-in class), 145
- `CloudFormation.Paginator.ListStackResources` (built-in class), 148
- `CloudFormation.Paginator.ListStacks` (built-in class), 150
- `CloudFormation.ServiceResource` (built-in class), 152
- `CloudFormation.Stack` (built-in class), 158
- `CloudFormation.StackResource` (built-in class), 165
- `CloudFormation.StackResourceSummary` (built-in class), 167
- `CloudFront.Client` (built-in class), 169

CloudFront.Paginator.ListCloudFrontOriginAccessIdentities (built-in class), 296

CloudFront.Paginator.ListDistributions (built-in class), 298

CloudFront.Paginator.ListInvalidations (built-in class), 312

CloudFront.Paginator.ListStreamingDistributions (built-in class), 313

CloudFront.Waiter.DistributionDeployed (built-in class), 316

CloudFront.Waiter.InvalidationCompleted (built-in class), 316

CloudFront.Waiter.StreamingDistributionDeployed (built-in class), 317

CloudHSM.Client (built-in class), 317

CloudSearch.Client (built-in class), 331

CloudSearchDomain.Client (built-in class), 404

CloudTrail.Client (built-in class), 416

CloudWatch.Client (built-in class), 426

CloudWatch.Paginator.DescribeAlarmHistory (built-in class), 444

CloudWatch.Paginator.DescribeAlarms (built-in class), 446

CloudWatch.Paginator.ListMetrics (built-in class), 450

CloudWatchLogs.Client (built-in class), 3052

CloudWatchLogs.Paginator.DescribeDestinations (built-in class), 3076

CloudWatchLogs.Paginator.DescribeLogGroups (built-in class), 3077

CloudWatchLogs.Paginator.DescribeLogStreams (built-in class), 3079

CloudWatchLogs.Paginator.DescribeMetricFilters (built-in class), 3081

CloudWatchLogs.Paginator.DescribeSubscriptionFilters (built-in class), 3082

CloudWatchLogs.Paginator.FilterLogEvents (built-in class), 3084

CodeCommit.Client (built-in class), 452

CodeDeploy.Client (built-in class), 462

CodePipeline.Client (built-in class), 507

CognitoIdentity.Client (built-in class), 559

CognitoSync.Client (built-in class), 576

Collection (class in boto3.resources.model), 4639

CollectionFactory (class in boto3.resources.collection), 4635

CollectionManager (class in boto3.resources.collection), 4635

collections (boto3.resources.model.ResourceModel attribute), 4640

complete() (Glacier.MultipartUpload method), 2691

complete() (S3.MultipartUpload method), 4081

complete_lifecycle_action() (AutoScaling.Client method), 42

complete_multipart_upload() (Glacier.Client method), 2637

complete_multipart_upload() (S3.Client method), 3956

complete_vault_lock() (Glacier.Client method), 2638

completed (Glacier.Job attribute), 2685

completed_jobs (Glacier.Vault attribute), 2705

completion_date (Glacier.Job attribute), 2685

ConfigService.Client (built-in class), 595

configuration_manager (OpsWorks.Stack attribute), 3269

configure_health_check() (ElasticLoadBalancing.Client method), 2540

confirm_connection() (DirectConnect.Client method), 709

confirm_private_virtual_interface() (DirectConnect.Client method), 709

confirm_product_instance() (EC2.Client method), 1130

confirm_public_virtual_interface() (DirectConnect.Client method), 711

confirm_subscription() (SNS.Client method), 4167

confirm_subscription() (SNS.Topic method), 4209

connect_directory() (DirectoryService.Client method), 736

console_output() (EC2.Instance method), 1618

contains() (boto3.dynamodb.conditions.Attr method), 4649

content_disposition (S3.Object attribute), 4089

content_encoding (S3.Object attribute), 4089

content_language (S3.Object attribute), 4089

content_length (S3.Object attribute), 4089

content_type (S3.Object attribute), 4089

copy() (boto3.resources.base.ResourceMeta method), 4644

copy() (EC2.Snapshot method), 1712

copy_cluster_snapshot() (Redshift.Client method), 3597

copy_db_cluster_snapshot() (RDS.Client method), 3286

copy_db_parameter_group() (RDS.Client method), 3289

copy_db_snapshot() (RDS.Client method), 3290

copy_from() (S3.MultipartUploadPart method), 4084

copy_from() (S3.Object method), 4090

copy_from() (S3.ObjectSummary method), 4108

copy_image() (EC2.Client method), 1130

copy_object() (S3.Client method), 3957

copy_option_group() (RDS.Client method), 3293

copy_snapshot() (EC2.Client method), 1131

copy_snapshot() (ElastiCache.Client method), 1899

Cors() (S3.Bucket method), 4046

cors_rules (S3.BucketCors attribute), 4057

count_closed_workflow_executions() (SWF.Client method), 4376

count_open_workflow_executions() (SWF.Client method), 4378

count_pending_activity_tasks() (SWF.Client method), 4380

count_pending_decision_tasks() (SWF.Client method), 4381

create() (Glacier.Vault method), 2699
 create() (IAM.Group method), 2910
 create() (IAM.LoginProfile method), 2922
 create() (IAM.User method), 2956
 create() (S3.Bucket method), 4041
 create_access_key() (IAM.Client method), 2718
 create_access_key_pair() (IAM.User method), 2956
 create_account_alias() (IAM.Client method), 2719
 create_account_alias() (IAM.ServiceResource method), 2874
 create_account_password_policy() (IAM.ServiceResource method), 2874
 create_alias() (DirectoryService.Client method), 737
 create_alias() (KMS.Client method), 3002
 create_app() (OpsWorks.Client method), 3165
 create_app_cookie_stickiness_policy() (ElasticLoadBalancing.Client method), 2542
 create_application() (CodeDeploy.Client method), 470
 create_application() (ElasticBeanstalk.Client method), 2042
 create_application_version() (ElasticBeanstalk.Client method), 2043
 create_association() (SSM.Client method), 4260
 create_association_batch() (SSM.Client method), 4261
 create_auto_scaling_group() (AutoScaling.Client method), 43
 create_batch_prediction() (MachineLearning.Client method), 3087
 create_branch() (CodeCommit.Client method), 454
 create_bucket() (S3.Client method), 3960
 create_bucket() (S3.ServiceResource method), 4035
 create_cache_cluster() (ElastiCache.Client method), 1902
 create_cache_parameter_group() (ElastiCache.Client method), 1912
 create_cache_security_group() (ElastiCache.Client method), 1913
 create_cache_subnet_group() (ElastiCache.Client method), 1914
 create_cached_iscsi_volume() (StorageGateway.Client method), 4279
 create_case() (Support.Client method), 4349
 create_cloud_front_origin_access_identity() (CloudFront.Client method), 169
 create_cluster() (ECS.Client method), 1792
 create_cluster() (Redshift.Client method), 3601
 create_cluster_parameter_group() (Redshift.Client method), 3612
 create_cluster_security_group() (Redshift.Client method), 3613
 create_cluster_snapshot() (Redshift.Client method), 3616
 create_cluster_subnet_group() (Redshift.Client method), 3620
 create_computer() (DirectoryService.Client method), 738
 create_configuration_template() (ElasticBeanstalk.Client method), 2045
 create_connection() (DirectConnect.Client method), 711
 create_custom_action_type() (CodePipeline.Client method), 509
 create_customer_gateway() (EC2.Client method), 1133
 create_data_source_from_rds() (MachineLearning.Client method), 3088
 create_data_source_from_redshift() (MachineLearning.Client method), 3092
 create_data_source_from_s3() (MachineLearning.Client method), 3095
 create_date (IAM.AccessKey attribute), 2895
 create_date (IAM.AccessKeyPair attribute), 2897
 create_date (IAM.CurrentUser attribute), 2904
 create_date (IAM.Group attribute), 2909
 create_date (IAM.InstanceProfile attribute), 2920
 create_date (IAM.LoginProfile attribute), 2922
 create_date (IAM.Policy attribute), 2927
 create_date (IAM.PolicyVersion attribute), 2937
 create_date (IAM.Role attribute), 2939
 create_date (IAM.SamlProvider attribute), 2948
 create_date (IAM.User attribute), 2954
 create_db_cluster() (RDS.Client method), 3297
 create_db_cluster_parameter_group() (RDS.Client method), 3303
 create_db_cluster_snapshot() (RDS.Client method), 3305
 create_db_instance() (RDS.Client method), 3308
 create_db_instance_read_replica() (RDS.Client method), 3325
 create_db_parameter_group() (RDS.Client method), 3335
 create_db_security_group() (RDS.Client method), 3337
 create_db_snapshot() (RDS.Client method), 3340
 create_db_subnet_group() (RDS.Client method), 3343
 create_deployment() (CodeDeploy.Client method), 470
 create_deployment() (OpsWorks.Client method), 3169
 create_deployment_config() (CodeDeploy.Client method), 472
 create_deployment_group() (CodeDeploy.Client method), 473
 create_device_pool() (DeviceFarm.Client method), 642
 create_dhcp_options() (EC2.Client method), 1135
 create_dhcp_options() (EC2.ServiceResource method), 1532
 create_directory() (DirectoryService.Client method), 739
 create_distribution() (CloudFront.Client method), 171
 create_document() (SSM.Client method), 4263
 create_domain() (CloudSearch.Client method), 333
 create_domain() (SimpleDB.Client method), 4131
 create_entry() (EC2.NetworkAcl method), 1663
 create_environment() (ElasticBeanstalk.Client method), 2048
 create_evaluation() (MachineLearning.Client method), 3097

`create_event_source_mapping()` (Lambda.Client method), 3026

`create_event_subscription()` (RDS.Client method), 3345

`create_event_subscription()` (Redshift.Client method), 3622

`create_export_task()` (CloudWatchLogs.Client method), 3053

`create_file_system()` (EFS.Client method), 1882

`create_flow_logs()` (EC2.Client method), 1137

`create_function()` (Lambda.Client method), 3028

`create_grant()` (KMS.Client method), 3003

`create_group()` (IAM.Client method), 2719

`create_group()` (IAM.ServiceResource method), 2875

`create_hapg()` (CloudHSM.Client method), 318

`create_health_check()` (Route53.Client method), 3819

`create_hosted_zone()` (Route53.Client method), 3823

`create_hsm()` (CloudHSM.Client method), 318

`create_hsm_client_certificate()` (Redshift.Client method), 3625

`create_hsm_configuration()` (Redshift.Client method), 3627

`create_identity_pool()` (CognitoIdentity.Client method), 560

`create_image()` (EC2.Client method), 1138

`create_image()` (EC2.Instance method), 1619

`create_instance()` (OpsWorks.Client method), 3171

`create_instance_export_task()` (EC2.Client method), 1141

`create_instance_profile()` (IAM.Client method), 2720

`create_instance_profile()` (IAM.ServiceResource method), 2875

`create_instances()` (EC2.ServiceResource method), 1533

`create_instances()` (EC2.Subnet method), 1721

`create_interconnect()` (DirectConnect.Client method), 713

`create_internet_gateway()` (EC2.Client method), 1143

`create_internet_gateway()` (EC2.ServiceResource method), 1540

`create_invalidation()` (CloudFront.Client method), 198

`create_job()` (ElasticTranscoder.Client method), 2102

`create_job()` (ImportExport.Client method), 2972

`create_key()` (KMS.Client method), 3004

`create_key_pair()` (EC2.Client method), 1144

`create_key_pair()` (EC2.ServiceResource method), 1540

`create_launch_configuration()` (AutoScaling.Client method), 45

`create_layer()` (OpsWorks.Client method), 3174

`create_layer()` (OpsWorks.Stack method), 3271

`create_lb_cookie_stickiness_policy()` (ElasticLoadBalancing.Client method), 2543

`create_load_balancer()` (ElasticLoadBalancing.Client method), 2544

`create_load_balancer_listeners()` (ElasticLoadBalancing.Client method), 2546

`create_load_balancer_policy()` (ElasticLoadBalancing.Client method), 2547

`create_log_group()` (CloudWatchLogs.Client method), 3054

`create_log_stream()` (CloudWatchLogs.Client method), 3055

`create_login_profile()` (IAM.Client method), 2722

`create_login_profile()` (IAM.User method), 2956

`create_luna_client()` (CloudHSM.Client method), 319

`create_ml_model()` (MachineLearning.Client method), 3098

`create_mount_target()` (EFS.Client method), 1884

`create_multipart_upload()` (S3.Client method), 3961

`create_network_acl()` (EC2.Client method), 1145

`create_network_acl()` (EC2.ServiceResource method), 1540

`create_network_acl()` (EC2.Vpc method), 1760

`create_network_acl_entry()` (EC2.Client method), 1148

`create_network_interface()` (EC2.Client method), 1149

`create_network_interface()` (EC2.ServiceResource method), 1541

`create_network_interface()` (EC2.Subnet method), 1728

`create_open_id_connect_provider()` (IAM.Client method), 2723

`create_option_group()` (RDS.Client method), 3348

`create_or_update_tags()` (AutoScaling.Client method), 49

`create_pipeline()` (CodePipeline.Client method), 514

`create_pipeline()` (DataPipeline.Client method), 612

`create_pipeline()` (ElasticTranscoder.Client method), 2179

`create_placement_group()` (EC2.Client method), 1154

`create_placement_group()` (EC2.ServiceResource method), 1542

`create_platform_application()` (SNS.Client method), 4168

`create_platform_application()` (SNS.ServiceResource method), 4193

`create_platform_endpoint()` (SNS.Client method), 4169

`create_platform_endpoint()` (SNS.PlatformApplication method), 4198

`create_policy()` (IAM.Client method), 2725

`create_policy()` (IAM.Group method), 2911

`create_policy()` (IAM.ServiceResource method), 2876

`create_policy()` (IAM.User method), 2957

`create_policy_version()` (IAM.Client method), 2727

`create_preset()` (ElasticTranscoder.Client method), 2192

`create_private_virtual_interface()` (DirectConnect.Client method), 715

`create_project()` (DeviceFarm.Client method), 645

`create_public_virtual_interface()` (DirectConnect.Client method), 718

`create_queue()` (SQS.Client method), 4219

`create_queue()` (SQS.ServiceResource method), 4238

`create_realtime_endpoint()` (MachineLearning.Client method), 3100

[create_replication_group\(\)](#) (ElastiCache.Client method), 1916
[create_repository\(\)](#) (CodeCommit.Client method), 454
[create_request_parameters\(\)](#) (in module `boto3.resources.params`), 4642
[create_reserved_instances_listing\(\)](#) (EC2.Client method), 1154
[create_reusable_delegation_set\(\)](#) (Route53.Client method), 3826
[create_role\(\)](#) (IAM.Client method), 2728
[create_role\(\)](#) (IAM.ServiceResource method), 2877
[create_route\(\)](#) (EC2.Client method), 1157
[create_route\(\)](#) (EC2.RouteTable method), 1690
[create_route_table\(\)](#) (EC2.Client method), 1159
[create_route_table\(\)](#) (EC2.ServiceResource method), 1542
[create_route_table\(\)](#) (EC2.Vpc method), 1760
[create_saml_provider\(\)](#) (IAM.Client method), 2729
[create_saml_provider\(\)](#) (IAM.ServiceResource method), 2877
[create_security_group\(\)](#) (EC2.Client method), 1161
[create_security_group\(\)](#) (EC2.ServiceResource method), 1543
[create_security_group\(\)](#) (EC2.Vpc method), 1760
[create_server_certificate\(\)](#) (IAM.ServiceResource method), 2878
[create_service\(\)](#) (ECS.Client method), 1793
[create_signing_certificate\(\)](#) (IAM.ServiceResource method), 2879
[create_snapshot\(\)](#) (DirectoryService.Client method), 740
[create_snapshot\(\)](#) (EC2.Client method), 1162
[create_snapshot\(\)](#) (EC2.ServiceResource method), 1544
[create_snapshot\(\)](#) (EC2.Volume method), 1745
[create_snapshot\(\)](#) (ElastiCache.Client method), 1922
[create_snapshot\(\)](#) (StorageGateway.Client method), 4280
[create_snapshot_copy_grant\(\)](#) (Redshift.Client method), 3629
[create_snapshot_from_volume_recovery_point\(\)](#) (StorageGateway.Client method), 4281
[create_spot_datafeed_subscription\(\)](#) (EC2.Client method), 1165
[create_stack\(\)](#) (CloudFormation.Client method), 120
[create_stack\(\)](#) (CloudFormation.ServiceResource method), 152
[create_stack\(\)](#) (OpsWorks.Client method), 3177
[create_stack\(\)](#) (OpsWorks.ServiceResource method), 3258
[create_storage_location\(\)](#) (ElasticBeanstalk.Client method), 2053
[create_stored_iscsi_volume\(\)](#) (StorageGateway.Client method), 4282
[create_stream\(\)](#) (Kinesis.Client method), 2981
[create_streaming_distribution\(\)](#) (CloudFront.Client method), 200
[create_subnet\(\)](#) (EC2.Client method), 1166
[create_subnet\(\)](#) (EC2.ServiceResource method), 1545
[create_subnet\(\)](#) (EC2.Vpc method), 1761
[create_table\(\)](#) (DynamoDB.Client method), 780
[create_table\(\)](#) (DynamoDB.ServiceResource method), 991
[create_tags\(\)](#) (EC2.Client method), 1168
[create_tags\(\)](#) (EC2.DhcpOptions method), 1599
[create_tags\(\)](#) (EC2.Image method), 1604
[create_tags\(\)](#) (EC2.Instance method), 1621
[create_tags\(\)](#) (EC2.InternetGateway method), 1659
[create_tags\(\)](#) (EC2.NetworkAcl method), 1665
[create_tags\(\)](#) (EC2.NetworkInterface method), 1672
[create_tags\(\)](#) (EC2.PlacementGroup method), 1679
[create_tags\(\)](#) (EC2.RouteTable method), 1691
[create_tags\(\)](#) (EC2.SecurityGroup method), 1703
[create_tags\(\)](#) (EC2.ServiceResource method), 1545, 1560
[create_tags\(\)](#) (EC2.Snapshot method), 1713
[create_tags\(\)](#) (EC2.Subnet method), 1729, 1730
[create_tags\(\)](#) (EC2.Volume method), 1746
[create_tags\(\)](#) (EC2.Vpc method), 1762, 1769
[create_tags\(\)](#) (EFS.Client method), 1886
[create_tags\(\)](#) (Redshift.Client method), 3630
[create_tapes\(\)](#) (StorageGateway.Client method), 4283
[create_time](#) (EC2.Volume attribute), 1743
[create_topic\(\)](#) (SNS.Client method), 4170
[create_topic\(\)](#) (SNS.ServiceResource method), 4193
[create_trail\(\)](#) (CloudTrail.Client method), 416
[create_upload\(\)](#) (DeviceFarm.Client method), 645
[create_user\(\)](#) (IAM.Client method), 2730
[create_user\(\)](#) (IAM.ServiceResource method), 2879
[create_user_profile\(\)](#) (OpsWorks.Client method), 3182
[create_vault\(\)](#) (Glacier.Account method), 2679
[create_vault\(\)](#) (Glacier.Client method), 2639
[create_vault\(\)](#) (Glacier.ServiceResource method), 2676
[create_version\(\)](#) (IAM.Policy method), 2929
[create_virtual_mfa_device\(\)](#) (IAM.Client method), 2732
[create_virtual_mfa_device\(\)](#) (IAM.ServiceResource method), 2879
[create_volume\(\)](#) (EC2.Client method), 1169
[create_volume\(\)](#) (EC2.ServiceResource method), 1546
[create_vpc\(\)](#) (EC2.Client method), 1172
[create_vpc\(\)](#) (EC2.ServiceResource method), 1548
[create_vpc_endpoint\(\)](#) (EC2.Client method), 1174
[create_vpc_peering_connection\(\)](#) (EC2.Client method), 1176
[create_vpc_peering_connection\(\)](#) (EC2.ServiceResource method), 1548
[create_vpn_connection\(\)](#) (EC2.Client method), 1178
[create_vpn_connection_route\(\)](#) (EC2.Client method), 1181
[create_vpn_gateway\(\)](#) (EC2.Client method), 1181
[create_workspaces\(\)](#) (WorkSpaces.Client method), 4621
[created_at](#) (OpsWorks.Layer attribute), 3266

created_at (OpsWorks.Stack attribute), 3269
creation_date (EC2.Image attribute), 1602
creation_date (Glacier.Job attribute), 2686
creation_date (Glacier.MultipartUpload attribute), 2690
creation_date (Glacier.Vault attribute), 2699
creation_date (S3.Bucket attribute), 4041
creation_date_time (DynamoDB.Table attribute), 999
creation_time (CloudFormation.Stack attribute), 159
CurrentUser() (IAM.ServiceResource method), 2881
custom_cookbooks_source (OpsWorks.Stack attribute), 3269
custom_instance_profile_arn (OpsWorks.Layer attribute), 3266
custom_json (OpsWorks.Layer attribute), 3266
custom_json (OpsWorks.Stack attribute), 3270
custom_recipes (OpsWorks.Layer attribute), 3266
custom_security_group_ids (OpsWorks.Layer attribute), 3266

D

data (boto3.resources.base.ResourceMeta attribute), 4644
data_encryption_key_id (EC2.Snapshot attribute), 1710
DataPipeline.Client (built-in class), 609
DataPipeline.Paginator.DescribeObjects (built-in class), 636
DataPipeline.Paginator.ListPipelines (built-in class), 638
DataPipeline.Paginator.QueryObjects (built-in class), 639
deactivate() (IAM.AccessKey method), 2895
deactivate() (IAM.AccessKeyPair method), 2897
deactivate() (IAM.SigningCertificate method), 2953
deactivate_mfa_device() (IAM.Client method), 2734
deactivate_pipeline() (DataPipeline.Client method), 613
dead_letter_source_queues (SQS.Queue attribute), 4259
decode_authorization_message() (STS.Client method), 4341
decrypt() (KMS.Client method), 3005
default_availability_zone (OpsWorks.Stack attribute), 3270
default_for_az (EC2.Subnet attribute), 1721
default_instance_profile_arn (OpsWorks.Stack attribute), 3270
default_os (OpsWorks.Stack attribute), 3270
default_recipes (OpsWorks.Layer attribute), 3266
default_root_device_type (OpsWorks.Stack attribute), 3270
default_security_group_names (OpsWorks.Layer attribute), 3266
default_ssh_key_name (OpsWorks.Stack attribute), 3270
default_subnet_id (OpsWorks.Stack attribute), 3270
default_version (IAM.Policy attribute), 2928
default_version_id (IAM.Policy attribute), 2927
define_analysis_scheme() (CloudSearch.Client method), 335
define_expression() (CloudSearch.Client method), 339

define_index_field() (CloudSearch.Client method), 340
define_suggester() (CloudSearch.Client method), 356
DefinitionWithParams (class in boto3.resources.model), 4639
delete() (CloudFormation.Stack method), 160
delete() (DynamoDB.Table method), 1002
delete() (EC2.DhcpOptions method), 1600
delete() (EC2.InternetGateway method), 1660
delete() (EC2.KeyPair method), 1661
delete() (EC2.NetworkAcl method), 1666
delete() (EC2.NetworkInterface method), 1673
delete() (EC2.PlacementGroup method), 1678
delete() (EC2.RouteTable method), 1692
delete() (EC2.RouteTableAssociation method), 1696
delete() (EC2.SecurityGroup method), 1704
delete() (EC2.Snapshot method), 1714
delete() (EC2.Subnet method), 1729
delete() (EC2.Tag method), 1741
delete() (EC2.Volume method), 1747
delete() (EC2.Vpc method), 1763
delete() (EC2.VpcPeeringConnection method), 1789
delete() (Glacier.Archive method), 2682
delete() (Glacier.Notification method), 2696
delete() (Glacier.Vault method), 2700
delete() (IAM.AccessKey method), 2896
delete() (IAM.AccessKeyPair method), 2898
delete() (IAM.AccountPasswordPolicy method), 2899
delete() (IAM.Group method), 2911
delete() (IAM.GroupPolicy method), 2918
delete() (IAM.InstanceProfile method), 2920
delete() (IAM.LoginProfile method), 2923
delete() (IAM.Policy method), 2930
delete() (IAM.PolicyVersion method), 2937
delete() (IAM.Role method), 2940
delete() (IAM.RolePolicy method), 2946
delete() (IAM.SamlProvider method), 2948
delete() (IAM.ServerCertificate method), 2950
delete() (IAM.SigningCertificate method), 2953
delete() (IAM.User method), 2957
delete() (IAM.UserPolicy method), 2968
delete() (IAM.VirtualMfaDevice method), 2971
delete() (OpsWorks.Layer method), 3267
delete() (OpsWorks.Stack method), 3274
delete() (S3.Bucket method), 4042, 4049, 4051
delete() (S3.BucketCors method), 4057
delete() (S3.BucketLifecycle method), 4059
delete() (S3.BucketPolicy method), 4069
delete() (S3.BucketTagging method), 4072
delete() (S3.BucketWebsite method), 4076
delete() (S3.Object method), 4093
delete() (S3.ObjectSummary method), 4111
delete() (S3.ObjectVersion method), 4122
delete() (SNS.PlatformApplication method), 4199
delete() (SNS.PlatformEndpoint method), 4202

- `delete()` (SNS.Subscription method), 4206
- `delete()` (SNS.Topic method), 4209
- `delete()` (SQS.Message method), 4243
- `delete()` (SQS.Queue method), 4248
- `delete_access_key()` (IAM.Client method), 2734
- `delete_account_alias()` (IAM.Client method), 2734
- `delete_account_password_policy()` (IAM.Client method), 2735
- `delete_alarms()` (CloudWatch.Client method), 427
- `delete_alias()` (KMS.Client method), 3006
- `delete_analysis_scheme()` (CloudSearch.Client method), 358
- `delete_app()` (OpsWorks.Client method), 3183
- `delete_application()` (CodeDeploy.Client method), 475
- `delete_application()` (ElasticBeanstalk.Client method), 2054
- `delete_application_version()` (ElasticBeanstalk.Client method), 2054
- `delete_archive()` (Glacier.Client method), 2640
- `delete_association()` (SSM.Client method), 4264
- `delete_attributes()` (SimpleDB.Client method), 4132
- `delete_auto_scaling_group()` (AutoScaling.Client method), 50
- `delete_bandwidth_rate_limit()` (StorageGateway.Client method), 4284
- `delete_batch_prediction()` (MachineLearning.Client method), 3102
- `delete_bucket()` (S3.Client method), 3963
- `delete_bucket_cors()` (S3.Client method), 3963
- `delete_bucket_lifecycle()` (S3.Client method), 3963
- `delete_bucket_policy()` (S3.Client method), 3963
- `delete_bucket_replication()` (S3.Client method), 3964
- `delete_bucket_tagging()` (S3.Client method), 3964
- `delete_bucket_website()` (S3.Client method), 3964
- `delete_cache_cluster()` (ElastiCache.Client method), 1926
- `delete_cache_parameter_group()` (ElastiCache.Client method), 1932
- `delete_cache_security_group()` (ElastiCache.Client method), 1932
- `delete_cache_subnet_group()` (ElastiCache.Client method), 1932
- `delete_chap_credentials()` (StorageGateway.Client method), 4285
- `delete_cloud_front_origin_access_identity()` (CloudFront.Client method), 205
- `delete_cluster()` (ECS.Client method), 1797
- `delete_cluster()` (Redshift.Client method), 3631
- `delete_cluster_parameter_group()` (Redshift.Client method), 3639
- `delete_cluster_security_group()` (Redshift.Client method), 3639
- `delete_cluster_snapshot()` (Redshift.Client method), 3639
- `delete_cluster_subnet_group()` (Redshift.Client method), 3643
- `delete_configuration_template()` (ElasticBeanstalk.Client method), 2054
- `delete_connection()` (DirectConnect.Client method), 722
- `delete_custom_action_type()` (CodePipeline.Client method), 520
- `delete_customer_gateway()` (EC2.Client method), 1183
- `delete_data_source()` (MachineLearning.Client method), 3102
- `delete_dataset()` (CognitoSync.Client method), 577
- `delete_db_cluster()` (RDS.Client method), 3352
- `delete_db_cluster_parameter_group()` (RDS.Client method), 3356
- `delete_db_cluster_snapshot()` (RDS.Client method), 3356
- `delete_db_instance()` (RDS.Client method), 3358
- `delete_db_parameter_group()` (RDS.Client method), 3367
- `delete_db_security_group()` (RDS.Client method), 3367
- `delete_db_snapshot()` (RDS.Client method), 3368
- `delete_db_subnet_group()` (RDS.Client method), 3370
- `delete_delivery_channel()` (ConfigService.Client method), 595
- `delete_deployment_config()` (CodeDeploy.Client method), 476
- `delete_deployment_group()` (CodeDeploy.Client method), 476
- `delete_destination()` (CloudWatchLogs.Client method), 3055
- `delete_dhcp_options()` (EC2.Client method), 1183
- `delete_directory()` (DirectoryService.Client method), 741
- `delete_distribution()` (CloudFront.Client method), 206
- `delete_document()` (SSM.Client method), 4265
- `delete_domain()` (CloudSearch.Client method), 361
- `delete_domain()` (SimpleDB.Client method), 4133
- `delete_endpoint()` (SNS.Client method), 4171
- `delete_entry()` (EC2.NetworkAcl method), 1666
- `delete_environment_configuration()` (ElasticBeanstalk.Client method), 2055
- `delete_evaluation()` (MachineLearning.Client method), 3103
- `delete_event_source_mapping()` (Lambda.Client method), 3030
- `delete_event_subscription()` (RDS.Client method), 3371
- `delete_event_subscription()` (Redshift.Client method), 3643
- `delete_expression()` (CloudSearch.Client method), 363
- `delete_file_system()` (EFS.Client method), 1887
- `delete_flow_logs()` (EC2.Client method), 1183
- `delete_function()` (Lambda.Client method), 3032
- `delete_gateway()` (StorageGateway.Client method), 4286
- `delete_group()` (IAM.Client method), 2735
- `delete_group_policy()` (IAM.Client method), 2735
- `delete_hapg()` (CloudHSM.Client method), 320
- `delete_health_check()` (Route53.Client method), 3827
- `delete_hosted_zone()` (Route53.Client method), 3828

`delete_hsm()` (CloudHSM.Client method), 320
`delete_hsm_client_certificate()` (Redshift.Client method), 3643
`delete_hsm_configuration()` (Redshift.Client method), 3644
`delete_identities()` (CognitoIdentity.Client method), 562
`delete_identity()` (SES.Client method), 4142
`delete_identity_policy()` (SES.Client method), 4142
`delete_identity_pool()` (CognitoIdentity.Client method), 562
`delete_index_field()` (CloudSearch.Client method), 365
`delete_instance()` (OpsWorks.Client method), 3183
`delete_instance_profile()` (IAM.Client method), 2735
`delete_interconnect()` (DirectConnect.Client method), 724
`delete_internet_gateway()` (EC2.Client method), 1184
`delete_item()` (DynamoDB.Client method), 796
`delete_item()` (DynamoDB.Table method), 1012
`delete_key_pair()` (EC2.Client method), 1185
`delete_launch_configuration()` (AutoScaling.Client method), 51
`delete_layer()` (OpsWorks.Client method), 3183
`delete_lifecycle_hook()` (AutoScaling.Client method), 51
`delete_load_balancer()` (ElasticLoadBalancing.Client method), 2548
`delete_load_balancer_listeners()` (ElasticLoadBalancing.Client method), 2549
`delete_load_balancer_policy()` (ElasticLoadBalancing.Client method), 2549
`delete_log_group()` (CloudWatchLogs.Client method), 3055
`delete_log_stream()` (CloudWatchLogs.Client method), 3056
`delete_login_profile()` (IAM.Client method), 2736
`delete_luna_client()` (CloudHSM.Client method), 321
`delete_marker` (S3.Object attribute), 4089
`delete_message()` (SQS.Client method), 4220
`delete_message_batch()` (SQS.Client method), 4221
`delete_messages()` (SQS.Queue method), 4248
`delete_metric_filter()` (CloudWatchLogs.Client method), 3056
`delete_ml_model()` (MachineLearning.Client method), 3103
`delete_mount_target()` (EFS.Client method), 1887
`delete_network_acl()` (EC2.Client method), 1185
`delete_network_acl_entry()` (EC2.Client method), 1185
`delete_network_interface()` (EC2.Client method), 1186
`delete_notification_configuration()` (AutoScaling.Client method), 52
`delete_object()` (S3.Client method), 3964
`delete_objects()` (S3.Bucket method), 4042
`delete_objects()` (S3.Client method), 3965
`delete_open_id_connect_provider()` (IAM.Client method), 2736
`delete_option_group()` (RDS.Client method), 3372
`delete_pipeline()` (CodePipeline.Client method), 521
`delete_pipeline()` (DataPipeline.Client method), 613
`delete_pipeline()` (ElasticTranscoder.Client method), 2225
`delete_placement_group()` (EC2.Client method), 1186
`delete_platform_application()` (SNS.Client method), 4171
`delete_policy()` (AutoScaling.Client method), 52
`delete_policy()` (IAM.Client method), 2736
`delete_policy_version()` (IAM.Client method), 2737
`delete_preset()` (ElasticTranscoder.Client method), 2225
`delete_queue()` (SQS.Client method), 4223
`delete_realtime_endpoint()` (MachineLearning.Client method), 3104
`delete_replication_group()` (ElastiCache.Client method), 1933
`delete_repository()` (CodeCommit.Client method), 456
`delete_retention_policy()` (CloudWatchLogs.Client method), 3056
`delete_reusable_delegation_set()` (Route53.Client method), 3829
`delete_role()` (IAM.Client method), 2738
`delete_role_policy()` (IAM.Client method), 2738
`delete_route()` (EC2.Client method), 1187
`delete_route_table()` (EC2.Client method), 1187
`delete_saml_provider()` (IAM.Client method), 2738
`delete_scheduled_action()` (AutoScaling.Client method), 52
`delete_security_group()` (EC2.Client method), 1187
`delete_server_certificate()` (IAM.Client method), 2739
`delete_service()` (ECS.Client method), 1798
`delete_signing_certificate()` (IAM.Client method), 2739
`delete_snapshot()` (DirectoryService.Client method), 741
`delete_snapshot()` (EC2.Client method), 1188
`delete_snapshot()` (ElastiCache.Client method), 1936
`delete_snapshot_copy_grant()` (Redshift.Client method), 3644
`delete_snapshot_schedule()` (StorageGateway.Client method), 4286
`delete_spot_datafeed_subscription()` (EC2.Client method), 1188
`delete_ssh_public_key()` (IAM.Client method), 2739
`delete_stack()` (CloudFormation.Client method), 122
`delete_stack()` (OpsWorks.Client method), 3184
`delete_stream()` (Kinesis.Client method), 2982
`delete_streaming_distribution()` (CloudFront.Client method), 206
`delete_subnet()` (EC2.Client method), 1189
`delete_subscription_filter()` (CloudWatchLogs.Client method), 3056
`delete_suggester()` (CloudSearch.Client method), 373
`delete_table()` (DynamoDB.Client method), 815
`delete_tags()` (AutoScaling.Client method), 53
`delete_tags()` (EC2.Client method), 1189

- `delete_tags()` (EFS.Client method), 1888
- `delete_tags()` (Redshift.Client method), 3644
- `delete_tags_for_domain()` (Route53Domains.Client method), 3878
- `delete_tape()` (StorageGateway.Client method), 4287
- `delete_tape_archive()` (StorageGateway.Client method), 4288
- `delete_topic()` (SNS.Client method), 4171
- `delete_trail()` (CloudTrail.Client method), 418
- `delete_user()` (IAM.Client method), 2740
- `delete_user_policy()` (IAM.Client method), 2740
- `delete_user_profile()` (OpsWorks.Client method), 3184
- `delete_vault()` (Glacier.Client method), 2641
- `delete_vault_access_policy()` (Glacier.Client method), 2641
- `delete_vault_notifications()` (Glacier.Client method), 2642
- `delete_verified_email_address()` (SES.Client method), 4143
- `delete_virtual_interface()` (DirectConnect.Client method), 724
- `delete_virtual_mfa_device()` (IAM.Client method), 2740
- `delete_volume()` (EC2.Client method), 1190
- `delete_volume()` (StorageGateway.Client method), 4288
- `delete_vpc()` (EC2.Client method), 1190
- `delete_vpc_endpoints()` (EC2.Client method), 1191
- `delete_vpc_peering_connection()` (EC2.Client method), 1192
- `delete_vpn_connection()` (EC2.Client method), 1192
- `delete_vpn_connection_route()` (EC2.Client method), 1193
- `delete_vpn_gateway()` (EC2.Client method), 1193
- `deliver_config_snapshot()` (ConfigService.Client method), 595
- `deprecate_activity_type()` (SWF.Client method), 4382
- `deprecate_domain()` (SWF.Client method), 4383
- `deprecate_workflow_type()` (SWF.Client method), 4383
- `deregister()` (EC2.Image method), 1604
- `deregister_container_instance()` (ECS.Client method), 1801
- `deregister_ecs_cluster()` (OpsWorks.Client method), 3184
- `deregister_elastic_ip()` (OpsWorks.Client method), 3185
- `deregister_image()` (EC2.Client method), 1194
- `deregister_instance()` (OpsWorks.Client method), 3185
- `deregister_instances_from_load_balancer()` (ElasticLoadBalancing.Client method), 2550
- `deregister_on_premises_instance()` (CodeDeploy.Client method), 477
- `deregister_rds_db_instance()` (OpsWorks.Client method), 3185
- `deregister_task_definition()` (ECS.Client method), 1805
- `deregister_volume()` (OpsWorks.Client method), 3186
- `describe_account_attributes()` (EC2.Client method), 1194
- `describe_account_attributes()` (RDS.Client method), 3372
- `describe_account_limits()` (AutoScaling.Client method), 53
- `describe_activity_type()` (SWF.Client method), 4384
- `describe_addresses()` (EC2.Client method), 1195
- `describe_adjustment_types()` (AutoScaling.Client method), 54
- `describe_agent_versions()` (OpsWorks.Client method), 3186
- `describe_alarm_history()` (CloudWatch.Client method), 427
- `describe_alarms()` (CloudWatch.Client method), 428
- `describe_alarms_for_metric()` (CloudWatch.Client method), 432
- `describe_analysis_schemes()` (CloudSearch.Client method), 375
- `describe_application_versions()` (ElasticBeanstalk.Client method), 2055
- `describe_applications()` (ElasticBeanstalk.Client method), 2056
- `describe_apps()` (OpsWorks.Client method), 3187
- `describe_association()` (SSM.Client method), 4265
- `describe_attachment()` (Support.Client method), 4351
- `describe_attribute()` (EC2.Image method), 1605
- `describe_attribute()` (EC2.Instance method), 1622
- `describe_attribute()` (EC2.NetworkInterface method), 1673
- `describe_attribute()` (EC2.Snapshot method), 1714
- `describe_attribute()` (EC2.Volume method), 1747
- `describe_attribute()` (EC2.Vpc method), 1763
- `describe_auto_scaling_groups()` (AutoScaling.Client method), 55
- `describe_auto_scaling_instances()` (AutoScaling.Client method), 59
- `describe_auto_scaling_notification_types()` (AutoScaling.Client method), 60
- `describe_availability_options()` (CloudSearch.Client method), 378
- `describe_availability_zones()` (EC2.Client method), 1197
- `describe_bandwidth_rate_limit()` (StorageGateway.Client method), 4289
- `describe_batch_predictions()` (MachineLearning.Client method), 3105
- `describe_bundle_tasks()` (EC2.Client method), 1199
- `describe_cache()` (StorageGateway.Client method), 4290
- `describe_cache_clusters()` (ElastiCache.Client method), 1939
- `describe_cache_engine_versions()` (ElastiCache.Client method), 1946
- `describe_cache_parameter_groups()` (ElastiCache.Client method), 1947
- `describe_cache_parameters()` (ElastiCache.Client method), 1948
- `describe_cache_security_groups()` (ElastiCache.Client method), 1951

`describe_cache_subnet_groups()` (ElastiCache.Client method), 1953

`describe_cached_iscsi_volumes()` (StorageGateway.Client method), 4290

`describe_cases()` (Support.Client method), 4352

`describe_certificates()` (RDS.Client method), 3373

`describe_chap_credentials()` (StorageGateway.Client method), 4292

`describe_classic_link_instances()` (EC2.Client method), 1201

`describe_cluster()` (EMR.Client method), 2586

`describe_cluster_parameter_groups()` (Redshift.Client method), 3644

`describe_cluster_parameters()` (Redshift.Client method), 3647

`describe_cluster_security_groups()` (Redshift.Client method), 3649

`describe_cluster_snapshots()` (Redshift.Client method), 3652

`describe_cluster_subnet_groups()` (Redshift.Client method), 3657

`describe_cluster_versions()` (Redshift.Client method), 3659

`describe_clusters()` (ECS.Client method), 1811

`describe_clusters()` (Redshift.Client method), 3661

`describe_commands()` (OpsWorks.Client method), 3191

`describe_communications()` (Support.Client method), 4356

`describe_configuration_options()` (ElasticBeanstalk.Client method), 2058

`describe_configuration_recorder_status()` (ConfigService.Client method), 596

`describe_configuration_recorders()` (ConfigService.Client method), 597

`describe_configuration_settings()` (ElasticBeanstalk.Client method), 2061

`describe_connections()` (DirectConnect.Client method), 725

`describe_connections_on_interconnect()` (DirectConnect.Client method), 727

`describe_container_instances()` (ECS.Client method), 1812

`describe_conversion_tasks()` (EC2.Client method), 1204

`describe_customer_gateways()` (EC2.Client method), 1208

`describe_data_sources()` (MachineLearning.Client method), 3108

`describe_dataset()` (CognitoSync.Client method), 578

`describe_db_cluster_parameter_groups()` (RDS.Client method), 3375

`describe_db_cluster_parameters()` (RDS.Client method), 3376

`describe_db_cluster_snapshots()` (RDS.Client method), 3379

`describe_db_clusters()` (RDS.Client method), 3381

`describe_db_engine_versions()` (RDS.Client method), 3386

`describe_db_instances()` (RDS.Client method), 3388

`describe_db_log_files()` (RDS.Client method), 3397

`describe_db_parameter_groups()` (RDS.Client method), 3399

`describe_db_parameters()` (RDS.Client method), 3400

`describe_db_security_groups()` (RDS.Client method), 3403

`describe_db_snapshots()` (RDS.Client method), 3405

`describe_db_subnet_groups()` (RDS.Client method), 3408

`describe_default_cluster_parameters()` (Redshift.Client method), 3669

`describe_delivery_channel_status()` (ConfigService.Client method), 599

`describe_delivery_channels()` (ConfigService.Client method), 601

`describe_deployments()` (OpsWorks.Client method), 3193

`describe_destinations()` (CloudWatchLogs.Client method), 3057

`describe_dhcp_options()` (EC2.Client method), 1210

`describe_directories()` (DirectoryService.Client method), 742

`describe_document()` (SSM.Client method), 4266

`describe_domain()` (SWF.Client method), 4388

`describe_domains()` (CloudSearch.Client method), 379

`describe_ecs_clusters()` (OpsWorks.Client method), 3196

`describe_elastic_ips()` (OpsWorks.Client method), 3198

`describe_elastic_load_balancers()` (OpsWorks.Client method), 3199

`describe_engine_default_cluster_parameters()` (RDS.Client method), 3411

`describe_engine_default_parameters()` (ElastiCache.Client method), 1954

`describe_engine_default_parameters()` (RDS.Client method), 3413

`describe_environment_health()` (ElasticBeanstalk.Client method), 2063

`describe_environment_resources()` (ElasticBeanstalk.Client method), 2067

`describe_environments()` (ElasticBeanstalk.Client method), 2069

`describe_evaluations()` (MachineLearning.Client method), 3112

`describe_event_categories()` (RDS.Client method), 3415

`describe_event_categories()` (Redshift.Client method), 3671

`describe_event_subscriptions()` (RDS.Client method), 3416

`describe_event_subscriptions()` (Redshift.Client method), 3672

`describe_events()` (ElastiCache.Client method), 1957

[describe_events\(\) \(ElasticBeanstalk.Client method\), 2073](#)
[describe_events\(\) \(RDS.Client method\), 3418](#)
[describe_events\(\) \(Redshift.Client method\), 3674](#)
[describe_export_tasks\(\) \(CloudWatchLogs.Client method\), 3058](#)
[describe_export_tasks\(\) \(EC2.Client method\), 1213](#)
[describe_expressions\(\) \(CloudSearch.Client method\), 382](#)
[describe_file_systems\(\) \(EFS.Client method\), 1888](#)
[describe_flow_logs\(\) \(EC2.Client method\), 1214](#)
[describe_gateway_information\(\) \(StorageGateway.Client method\), 4293](#)
[describe_hapg\(\) \(CloudHSM.Client method\), 321](#)
[describe_hsm\(\) \(CloudHSM.Client method\), 323](#)
[describe_hsm_client_certificates\(\) \(Redshift.Client method\), 3677](#)
[describe_hsm_configurations\(\) \(Redshift.Client method\), 3679](#)
[describe_identity\(\) \(CognitoIdentity.Client method\), 563](#)
[describe_identity_pool\(\) \(CognitoIdentity.Client method\), 563](#)
[describe_identity_pool_usage\(\) \(CognitoSync.Client method\), 580](#)
[describe_identity_usage\(\) \(CognitoSync.Client method\), 580](#)
[describe_image_attribute\(\) \(EC2.Client method\), 1216](#)
[describe_images\(\) \(EC2.Client method\), 1219](#)
[describe_import_image_tasks\(\) \(EC2.Client method\), 1226](#)
[describe_import_snapshot_tasks\(\) \(EC2.Client method\), 1229](#)
[describe_index_fields\(\) \(CloudSearch.Client method\), 383](#)
[describe_instance_attribute\(\) \(EC2.Client method\), 1231](#)
[describe_instance_health\(\) \(ElasticLoadBalancing.Client method\), 2551](#)
[describe_instance_status\(\) \(EC2.Client method\), 1235](#)
[describe_instances\(\) \(EC2.Client method\), 1239](#)
[describe_instances\(\) \(OpsWorks.Client method\), 3200](#)
[describe_instances_health\(\) \(ElasticBeanstalk.Client method\), 2074](#)
[describe_interconnects\(\) \(DirectConnect.Client method\), 729](#)
[describe_internet_gateways\(\) \(EC2.Client method\), 1252](#)
[describe_job\(\) \(Glacier.Client method\), 2642](#)
[describe_job_flows\(\) \(EMR.Client method\), 2591](#)
[describe_key\(\) \(KMS.Client method\), 3007](#)
[describe_key_pairs\(\) \(EC2.Client method\), 1254](#)
[describe_launch_configurations\(\) \(AutoScaling.Client method\), 61](#)
[describe_layers\(\) \(OpsWorks.Client method\), 3206](#)
[describe_lifecycle_hook_types\(\) \(AutoScaling.Client method\), 64](#)
[describe_lifecycle_hooks\(\) \(AutoScaling.Client method\), 65](#)
[describe_load_balancer_attributes\(\) \(ElasticLoadBalancing.Client method\), 2552](#)
[describe_load_balancer_policies\(\) \(ElasticLoadBalancing.Client method\), 2554](#)
[describe_load_balancer_policy_types\(\) \(ElasticLoadBalancing.Client method\), 2556](#)
[describe_load_balancers\(\) \(AutoScaling.Client method\), 67](#)
[describe_load_balancers\(\) \(ElasticLoadBalancing.Client method\), 2557](#)
[describe_load_based_auto_scaling\(\) \(OpsWorks.Client method\), 3211](#)
[describe_locations\(\) \(DirectConnect.Client method\), 730](#)
[describe_log_groups\(\) \(CloudWatchLogs.Client method\), 3060](#)
[describe_log_streams\(\) \(CloudWatchLogs.Client method\), 3061](#)
[describe_logging_status\(\) \(Redshift.Client method\), 3681](#)
[describe_luna_client\(\) \(CloudHSM.Client method\), 324](#)
[describe_maintenance_start_time\(\) \(StorageGateway.Client method\), 4295](#)
[describe_metric_collection_types\(\) \(AutoScaling.Client method\), 68](#)
[describe_metric_filters\(\) \(CloudWatchLogs.Client method\), 3063](#)
[describe_ml_models\(\) \(MachineLearning.Client method\), 3116](#)
[describe_mount_target_security_groups\(\) \(EFS.Client method\), 1890](#)
[describe_mount_targets\(\) \(EFS.Client method\), 1891](#)
[describe_moving_addresses\(\) \(EC2.Client method\), 1256](#)
[describe_my_user_profile\(\) \(OpsWorks.Client method\), 3214](#)
[describe_network_acls\(\) \(EC2.Client method\), 1257](#)
[describe_network_interface_attribute\(\) \(EC2.Client method\), 1261](#)
[describe_network_interfaces\(\) \(EC2.Client method\), 1263](#)
[describe_notification_configurations\(\) \(AutoScaling.Client method\), 69](#)
[describe_objects\(\) \(DataPipeline.Client method\), 614](#)
[describe_option_group_options\(\) \(RDS.Client method\), 3421](#)
[describe_option_groups\(\) \(RDS.Client method\), 3423](#)
[describe_orderable_cluster_options\(\) \(Redshift.Client method\), 3682](#)
[describe_orderable_db_instance_options\(\) \(RDS.Client method\), 3427](#)
[describe_pending_maintenance_actions\(\) \(RDS.Client method\), 3430](#)
[describe_permissions\(\) \(OpsWorks.Client method\), 3215](#)
[describe_pipelines\(\) \(DataPipeline.Client method\), 615](#)
[describe_placement_groups\(\) \(EC2.Client method\), 1268](#)

`describe_policies()` (AutoScaling.Client method), 70
`describe_prefix_lists()` (EC2.Client method), 1270
`describe_raid_arrays()` (OpsWorks.Client method), 3216
`describe_rds_db_instances()` (OpsWorks.Client method), 3218
`describe_regions()` (EC2.Client method), 1271
`describe_replication_groups()` (ElastiCache.Client method), 1959
`describe_reserved_cache_nodes()` (ElastiCache.Client method), 1962
`describe_reserved_cache_nodes_offerings()` (ElastiCache.Client method), 1965
`describe_reserved_db_instances()` (RDS.Client method), 3432
`describe_reserved_db_instances_offerings()` (RDS.Client method), 3435
`describe_reserved_instances()` (EC2.Client method), 1273
`describe_reserved_instances_listings()` (EC2.Client method), 1276
`describe_reserved_instances_modifications()` (EC2.Client method), 1279
`describe_reserved_instances_offerings()` (EC2.Client method), 1282
`describe_reserved_node_offerings()` (Redshift.Client method), 3683
`describe_reserved_nodes()` (Redshift.Client method), 3685
`describe_resize()` (Redshift.Client method), 3688
`describe_route_tables()` (EC2.Client method), 1286
`describe_scaling_activities()` (AutoScaling.Client method), 73
`describe_scaling_parameters()` (CloudSearch.Client method), 392
`describe_scaling_process_types()` (AutoScaling.Client method), 75
`describe_scheduled_actions()` (AutoScaling.Client method), 75
`describe_security_groups()` (EC2.Client method), 1290
`describe_service_access_policies()` (CloudSearch.Client method), 394
`describe_service_errors()` (OpsWorks.Client method), 3219
`describe_services()` (ECS.Client method), 1816
`describe_services()` (Support.Client method), 4358
`describe_severity_levels()` (Support.Client method), 4359
`describe_snapshot_attribute()` (EC2.Client method), 1296
`describe_snapshot_copy_grants()` (Redshift.Client method), 3690
`describe_snapshot_schedule()` (StorageGateway.Client method), 4295
`describe_snapshots()` (DirectoryService.Client method), 746
`describe_snapshots()` (EC2.Client method), 1297
`describe_snapshots()` (ElastiCache.Client method), 1968
`describe_spot_datafeed_subscription()` (EC2.Client method), 1301
`describe_spot_fleet_instances()` (EC2.Client method), 1302
`describe_spot_fleet_request_history()` (EC2.Client method), 1304
`describe_spot_fleet_requests()` (EC2.Client method), 1306
`describe_spot_instance_requests()` (EC2.Client method), 1313
`describe_spot_price_history()` (EC2.Client method), 1323
`describe_stack_events()` (CloudFormation.Client method), 123
`describe_stack_provisioning_parameters()` (OpsWorks.Client method), 3221
`describe_stack_resource()` (CloudFormation.Client method), 124
`describe_stack_resources()` (CloudFormation.Client method), 126
`describe_stack_summary()` (OpsWorks.Client method), 3221
`describe_stacks()` (CloudFormation.Client method), 128
`describe_stacks()` (OpsWorks.Client method), 3223
`describe_status()` (EC2.Volume method), 1748
`describe_step()` (EMR.Client method), 2598
`describe_stored_iscsi_volumes()` (StorageGateway.Client method), 4296
`describe_stream()` (DynamoDBStreams.Client method), 1087
`describe_stream()` (Kinesis.Client method), 2982
`describe_subnets()` (EC2.Client method), 1325
`describe_subscription_filters()` (CloudWatchLogs.Client method), 3065
`describe_suggesters()` (CloudSearch.Client method), 395
`describe_table()` (DynamoDB.Client method), 824
`describe_tags()` (AutoScaling.Client method), 77
`describe_tags()` (EC2.Client method), 1328
`describe_tags()` (EFS.Client method), 1893
`describe_tags()` (ElasticLoadBalancing.Client method), 2563
`describe_tags()` (Redshift.Client method), 3692
`describe_tape_archives()` (StorageGateway.Client method), 4297
`describe_tape_recovery_points()` (StorageGateway.Client method), 4299
`describe_tapes()` (StorageGateway.Client method), 4300
`describe_task_definition()` (ECS.Client method), 1819
`describe_tasks()` (ECS.Client method), 1825
`describe_termination_policy_types()` (AutoScaling.Client method), 79
`describe_time_based_auto_scaling()` (OpsWorks.Client method), 3227
`describe_trails()` (CloudTrail.Client method), 418

- `describe_trusted_advisor_check_refresh_statuses()` (Support.Client method), 4360
- `describe_trusted_advisor_check_result()` (Support.Client method), 4361
- `describe_trusted_advisor_check_summaries()` (Support.Client method), 4363
- `describe_trusted_advisor_checks()` (Support.Client method), 4365
- `describe_upload_buffer()` (StorageGateway.Client method), 4302
- `describe_user_profiles()` (OpsWorks.Client method), 3229
- `describe_vault()` (Glacier.Client method), 2645
- `describe_virtual_gateways()` (DirectConnect.Client method), 731
- `describe_virtual_interfaces()` (DirectConnect.Client method), 732
- `describe_volume_attribute()` (EC2.Client method), 1329
- `describe_volume_status()` (EC2.Client method), 1330
- `describe_volumes()` (EC2.Client method), 1334
- `describe_volumes()` (OpsWorks.Client method), 3230
- `describe_vpc_attribute()` (EC2.Client method), 1338
- `describe_vpc_classic_link()` (EC2.Client method), 1339
- `describe_vpc_endpoint_services()` (EC2.Client method), 1341
- `describe_vpc_endpoints()` (EC2.Client method), 1342
- `describe_vpc_peering_connections()` (EC2.Client method), 1344
- `describe_vpcs()` (EC2.Client method), 1347
- `describe_vpn_connections()` (EC2.Client method), 1349
- `describe_vpn_gateways()` (EC2.Client method), 1353
- `describe_vtl_devices()` (StorageGateway.Client method), 4302
- `describe_workflow_execution()` (SWF.Client method), 4389
- `describe_workflow_type()` (SWF.Client method), 4393
- `describe_working_storage()` (StorageGateway.Client method), 4304
- `describe_workspace_bundles()` (WorkSpaces.Client method), 4624
- `describe_workspace_directories()` (WorkSpaces.Client method), 4625
- `describe_workspaces()` (WorkSpaces.Client method), 4628
- `description` (CloudFormation.Stack attribute), 159
- `description` (CloudFormation.StackResource attribute), 166
- `description` (EC2.Image attribute), 1602
- `description` (EC2.NetworkInterface attribute), 1670
- `description` (EC2.SecurityGroup attribute), 1697
- `description` (EC2.Snapshot attribute), 1710
- `description` (IAM.Policy attribute), 2927
- `detach()` (EC2.NetworkInterface method), 1675
- `detach_classic_link_instance()` (EC2.Vpc method), 1764
- `detach_classic_link_vpc()` (EC2.Client method), 1355
- `detach_classic_link_vpc()` (EC2.Instance method), 1625
- `detach_elastic_load_balancer()` (OpsWorks.Client method), 3232
- `detach_from_instance()` (EC2.Volume method), 1752
- `detach_from_vpc()` (EC2.InternetGateway method), 1660
- `detach_group()` (IAM.Policy method), 2930
- `detach_group_policy()` (IAM.Client method), 2741
- `detach_instances()` (AutoScaling.Client method), 79
- `detach_internet_gateway()` (EC2.Client method), 1356
- `detach_internet_gateway()` (EC2.Vpc method), 1765
- `detach_load_balancer_from_subnets()` (ElasticLoadBalancing.Client method), 2563
- `detach_load_balancers()` (AutoScaling.Client method), 81
- `detach_network_interface()` (EC2.Client method), 1356
- `detach_policy()` (IAM.Group method), 2911
- `detach_policy()` (IAM.Role method), 2940
- `detach_policy()` (IAM.User method), 2957
- `detach_role()` (IAM.Policy method), 2930
- `detach_role_policy()` (IAM.Client method), 2741
- `detach_user()` (IAM.Policy method), 2931
- `detach_user_policy()` (IAM.Client method), 2742
- `detach_volume()` (EC2.Client method), 1356
- `detach_volume()` (EC2.Instance method), 1626
- `detach_vpn_gateway()` (EC2.Client method), 1358
- `DeviceFarm.Client` (built-in class), 642
- `dhcp_configurations` (EC2.DhcpOptions attribute), 1598
- `dhcp_options` (EC2.Vpc attribute), 1758
- `dhcp_options_id` (EC2.DhcpOptions attribute), 1599
- `dhcp_options_id` (EC2.Vpc attribute), 1757
- `dhcp_options_sets` (EC2.ServiceResource attribute), 1556
- `DhcpOptions()` (EC2.ServiceResource method), 1553
- `DirectConnect.Client` (built-in class), 699
- `DirectoryService.Client` (built-in class), 735
- `disable_alarm_actions()` (CloudWatch.Client method), 435
- `disable_availability_zones_for_load_balancer()` (ElasticLoadBalancing.Client method), 2564
- `disable_callback()` (boto3.s3.transfer.ReadFileChunk method), 4653
- `disable_classic_link()` (EC2.Vpc method), 1765
- `disable_domain_auto_renew()` (Route53Domains.Client method), 3879
- `disable_domain_transfer_lock()` (Route53Domains.Client method), 3880
- `disable_gateway()` (StorageGateway.Client method), 4305
- `disable_key()` (KMS.Client method), 3008
- `disable_key_rotation()` (KMS.Client method), 3008
- `disable_logging()` (Redshift.Client method), 3694
- `disable_metrics_collection()` (AutoScaling.Client method), 81
- `disable_radius()` (DirectoryService.Client method), 748
- `disable_rollback` (CloudFormation.Stack attribute), 159

- `disable_snapshot_copy()` (Redshift.Client method), 3695
- `disable_sso()` (DirectoryService.Client method), 748
- `disable_stage_transition()` (CodePipeline.Client method), 521
- `disable_upload_callbacks()` (in module boto3.s3.transfer), 4654
- `disable_vgw_route_propagation()` (EC2.Client method), 1358
- `disable_vpc_classic_link()` (EC2.Client method), 1359
- `disassociate()` (IAM.MfaDevice method), 2925
- `disassociate_address()` (EC2.Client method), 1359
- `disassociate_elastic_ip()` (OpsWorks.Client method), 3232
- `disassociate_route_table()` (EC2.Client method), 1360
- `disassociate_route_table()` (EC2.ServiceResource method), 1549
- `disassociate_vpc_from_hosted_zone()` (Route53.Client method), 3830
- `discover_poll_endpoint()` (ECS.Client method), 1828
- `document` (IAM.PolicyVersion attribute), 2937
- `domain_metadata()` (SimpleDB.Client method), 4133
- `download_db_log_file_portion()` (RDS.Client method), 3438
- `download_file()` (boto3.s3.transfer.MultipartDownloader method), 4653
- `download_file()` (boto3.s3.transfer.S3Transfer method), 4654
- `download_file()` (in module boto3.s3.inject), 4650
- `download_file()` (S3.Bucket method), 4041
- `download_file()` (S3.Client method), 3966
- `download_file()` (S3.Object method), 4090
- DynamoDB.Client (built-in class), 755
- DynamoDB.Paginator.ListTables (built-in class), 940
- DynamoDB.Paginator.Query (built-in class), 941
- DynamoDB.Paginator.Scan (built-in class), 961
- DynamoDB.ServiceResource (built-in class), 976
- DynamoDB.Table (built-in class), 998
- DynamoDB.Waiter.TableExists (built-in class), 975
- DynamoDB.Waiter.TableNotExists (built-in class), 976
- DynamoDBStreams.Client (built-in class), 1087
- E**
- `e_tag` (S3.MultipartUploadPart attribute), 4084
- `e_tag` (S3.Object attribute), 4089
- `e_tag` (S3.ObjectSummary attribute), 4108
- `e_tag` (S3.ObjectVersion attribute), 4122
- `ebs_optimized` (EC2.Instance attribute), 1613
- EC2.Client (built-in class), 1100
- EC2.DhcpOptions (built-in class), 1598
- EC2.Image (built-in class), 1601
- EC2.Instance (built-in class), 1610
- EC2.InternetGateway (built-in class), 1658
- EC2.KeyPair (built-in class), 1661
- EC2.NetworkAcl (built-in class), 1662
- EC2.NetworkInterface (built-in class), 1668
- EC2.Paginator.DescribeInstances (built-in class), 1456
- EC2.Paginator.DescribeInstanceStatus (built-in class), 1452
- EC2.Paginator.DescribeReservedInstancesModifications (built-in class), 1469
- EC2.Paginator.DescribeReservedInstancesOfferings (built-in class), 1473
- EC2.Paginator.DescribeSnapshots (built-in class), 1477
- EC2.Paginator.DescribeSpotPriceHistory (built-in class), 1480
- EC2.Paginator.DescribeTags (built-in class), 1483
- EC2.Paginator.DescribeVolumeStatus (built-in class), 1485
- EC2.PlacementGroup (built-in class), 1677
- EC2.RouteTable (built-in class), 1688
- EC2.RouteTableAssociation (built-in class), 1695
- EC2.SecurityGroup (built-in class), 1697
- EC2.ServiceResource (built-in class), 1531
- EC2.Snapshot (built-in class), 1709
- EC2.Subnet (built-in class), 1720
- EC2.Tag (built-in class), 1741
- EC2.Volume (built-in class), 1742
- EC2.Vpc (built-in class), 1756
- EC2.VpcPeeringConnection (built-in class), 1786
- EC2.Waiter.BundleTaskComplete (built-in class), 1489
- EC2.Waiter.ConversionTaskCancelled (built-in class), 1490
- EC2.Waiter.ConversionTaskCompleted (built-in class), 1491
- EC2.Waiter.ConversionTaskDeleted (built-in class), 1492
- EC2.Waiter.CustomerGatewayAvailable (built-in class), 1492
- EC2.Waiter.ExportTaskCancelled (built-in class), 1494
- EC2.Waiter.ExportTaskCompleted (built-in class), 1494
- EC2.Waiter.ImageAvailable (built-in class), 1494
- EC2.Waiter.InstanceExists (built-in class), 1496
- EC2.Waiter.InstanceRunning (built-in class), 1500
- EC2.Waiter.InstanceStatusOk (built-in class), 1505
- EC2.Waiter.InstanceStopped (built-in class), 1506
- EC2.Waiter.InstanceTerminated (built-in class), 1510
- EC2.Waiter.PasswordDataAvailable (built-in class), 1514
- EC2.Waiter.SnapshotCompleted (built-in class), 1515
- EC2.Waiter.SpotInstanceRequestFulfilled (built-in class), 1517
- EC2.Waiter.SubnetAvailable (built-in class), 1519
- EC2.Waiter.SystemStatusOk (built-in class), 1520
- EC2.Waiter.VolumeAvailable (built-in class), 1522
- EC2.Waiter.VolumeDeleted (built-in class), 1524
- EC2.Waiter.VolumeInUse (built-in class), 1525
- EC2.Waiter.VpcAvailable (built-in class), 1527
- EC2.Waiter.VpnConnectionAvailable (built-in class), 1528
- EC2.Waiter.VpnConnectionDeleted (built-in class), 1530

- ECS.Client (built-in class), 1791
- ECS.Paginator.ListClusters (built-in class), 1872
- ECS.Paginator.ListContainerInstances (built-in class), 1873
- ECS.Paginator.ListServices (built-in class), 1874
- ECS.Paginator.ListTaskDefinitionFamilies (built-in class), 1875
- ECS.Paginator.ListTaskDefinitions (built-in class), 1876
- ECS.Paginator.ListTasks (built-in class), 1877
- ECS.Waiter.ServicesInactive (built-in class), 1879
- ECS.Waiter.ServicesStable (built-in class), 1880
- ECS.Waiter.TasksRunning (built-in class), 1880
- ECS.Waiter.TasksStopped (built-in class), 1881
- EFS.Client (built-in class), 1881
- ElastiCache.Client (built-in class), 1895
- ElastiCache.Paginator.DescribeCacheClusters (built-in class), 2002
- ElastiCache.Paginator.DescribeCacheEngineVersions (built-in class), 2008
- ElastiCache.Paginator.DescribeCacheParameterGroups (built-in class), 2010
- ElastiCache.Paginator.DescribeCacheParameters (built-in class), 2012
- ElastiCache.Paginator.DescribeCacheSecurityGroups (built-in class), 2014
- ElastiCache.Paginator.DescribeCacheSubnetGroups (built-in class), 2016
- ElastiCache.Paginator.DescribeEngineDefaultParameters (built-in class), 2018
- ElastiCache.Paginator.DescribeEvents (built-in class), 2021
- ElastiCache.Paginator.DescribeReplicationGroups (built-in class), 2023
- ElastiCache.Paginator.DescribeReservedCacheNodes (built-in class), 2026
- ElastiCache.Paginator.DescribeReservedCacheNodesOfferings (built-in class), 2029
- ElastiCache.Paginator.DescribeSnapshots (built-in class), 2032
- ElastiCache.Waiter.CacheClusterAvailable (built-in class), 2037
- ElastiCache.Waiter.CacheClusterDeleted (built-in class), 2037
- ElastiCache.Waiter.ReplicationGroupAvailable (built-in class), 2038
- ElastiCache.Waiter.ReplicationGroupDeleted (built-in class), 2039
- ElasticBeanstalk.Client (built-in class), 2040
- ElasticBeanstalk.Paginator.DescribeEvents (built-in class), 2098
- ElasticLoadBalancing.Client (built-in class), 2537
- ElasticLoadBalancing.Paginator.DescribeLoadBalancers (built-in class), 2574
- ElasticTranscoder.Client (built-in class), 2101
- ElasticTranscoder.Paginator.ListJobsByPipeline (built-in class), 2428
- ElasticTranscoder.Paginator.ListJobsByStatus (built-in class), 2469
- ElasticTranscoder.Paginator.ListPipelines (built-in class), 2511
- ElasticTranscoder.Paginator.ListPresets (built-in class), 2519
- ElasticTranscoder.Waiter.JobComplete (built-in class), 2536
- emit() (boto3.NullHandler method), 4634
- EMR.Client (built-in class), 2580
- EMR.Paginator.ListBootstrapActions (built-in class), 2622
- EMR.Paginator.ListClusters (built-in class), 2623
- EMR.Paginator.ListInstanceGroups (built-in class), 2625
- EMR.Paginator.ListInstances (built-in class), 2628
- EMR.Paginator.ListSteps (built-in class), 2630
- EMR.Waiter.ClusterRunning (built-in class), 2633
- enable() (S3.BucketVersioning method), 4074
- enable_alarm_actions() (CloudWatch.Client method), 435
- enable_auto_healing (OpsWorks.Layer attribute), 3266
- enable_availability_zones_for_load_balancer() (ElasticLoadBalancing.Client method), 2565
- enable_callback() (boto3.s3.transfer.ReadFileChunk method), 4653
- enable_classic_link() (EC2.Vpc method), 1766
- enable_date (IAM.MfaDevice attribute), 2925
- enable_date (IAM.VirtualMfaDevice attribute), 2970
- enable_domain_auto_renew() (Route53Domains.Client method), 3880
- enable_domain_transfer_lock() (Route53Domains.Client method), 3881
- enable_io() (EC2.Volume method), 1753
- enable_key() (KMS.Client method), 3008
- enable_key_rotation() (KMS.Client method), 3008
- enable_logging() (Redshift.Client method), 3702
- enable_metrics_collection() (AutoScaling.Client method), 82
- enable_mfa() (IAM.User method), 2958
- enable_mfa_device() (IAM.Client method), 2742
- enable_radius() (DirectoryService.Client method), 749
- enable_snapshot_copy() (Redshift.Client method), 3703
- enable_sso() (DirectoryService.Client method), 750
- enable_stage_transition() (CodePipeline.Client method), 522
- enable_upload_callbacks() (in module boto3.s3.transfer), 4654
- enable_vgw_route_propagation() (EC2.Client method), 1360
- enable_volume_io() (EC2.Client method), 1360
- enable_vpc_classic_link() (EC2.Client method), 1361
- encrypt() (KMS.Client method), 3009

encrypted (EC2.Snapshot attribute), 1710
encrypted (EC2.Volume attribute), 1743
endpoints (SNS.PlatformApplication attribute), 4200
enter_standby() (AutoScaling.Client method), 82
entries (EC2.NetworkAcl attribute), 1663
eq() (boto3.dynamodb.conditions.Attr method), 4649
eq() (boto3.dynamodb.conditions.Key method), 4648
error_document (S3.BucketWebsite attribute), 4076
estimate_template_cost() (CloudFormation.Client method), 130
evaluate_expression() (DataPipeline.Client method), 617
Event() (CloudFormation.ServiceResource method), 154
event_id (CloudFormation.Event attribute), 157
events (boto3.session.Session attribute), 4646
events (CloudFormation.Stack attribute), 163
events (Glacier.Notification attribute), 2696
execute_policy() (AutoScaling.Client method), 84
exists() (boto3.dynamodb.conditions.Attr method), 4649
exit_standby() (AutoScaling.Client method), 85
expiration (S3.Object attribute), 4089
expiration_time (EC2.VpcPeeringConnection attribute), 1787
expire_passwords (IAM.AccountPasswordPolicy attribute), 2899
expires (S3.Object attribute), 4089

F

failed_jobs (Glacier.Vault attribute), 2706
failover_db_cluster() (RDS.Client method), 3439
filter() (boto3.resources.collection.CollectionManager method), 4636
filter() (boto3.resources.collection.ResourceCollection method), 4637
filter() (CloudFormation.ServiceResource method), 155
filter() (CloudFormation.Stack method), 163, 164
filter() (DynamoDB.ServiceResource method), 997
filter() (EC2.Instance method), 1641
filter() (EC2.PlacementGroup method), 1680
filter() (EC2.RouteTable method), 1693
filter() (EC2.ServiceResource method), 1556, 1558, 1561, 1573, 1575, 1576, 1579, 1581, 1583, 1585, 1588, 1590, 1592, 1594, 1596
filter() (EC2.Subnet method), 1731, 1740
filter() (EC2.Volume method), 1755
filter() (EC2.Vpc method), 1768, 1770, 1778, 1780–1785
filter() (Glacier.Account method), 2680
filter() (Glacier.ServiceResource method), 2678
filter() (Glacier.Vault method), 2705, 2706, 2708–2711
filter() (IAM.CurrentUser method), 2905–2907
filter() (IAM.Group method), 2913, 2915, 2916
filter() (IAM.Policy method), 2931, 2933–2935
filter() (IAM.Role method), 2941, 2943, 2944
filter() (IAM.ServiceResource method), 2884, 2885, 2887–2890, 2892, 2893

filter() (IAM.User method), 2960, 2962–2966
filter() (OpsWorks.ServiceResource method), 3263
filter() (OpsWorks.Stack method), 3275
filter() (S3.Bucket method), 4048, 4050, 4053
filter() (S3.MultipartUpload method), 4082
filter() (S3.ServiceResource method), 4039
filter() (SNS.PlatformApplication method), 4200
filter() (SNS.ServiceResource method), 4195, 4196
filter() (SNS.Topic method), 4213
filter() (SQS.Queue method), 4259
filter() (SQS.ServiceResource method), 4240
filter_log_events() (CloudWatchLogs.Client method), 3066
from_filename() (boto3.s3.transfer.ReadFileChunk class method), 4653

G

generate_credential_report() (IAM.Client method), 2742
generate_data_key() (KMS.Client method), 3010
generate_data_key_without_plaintext() (KMS.Client method), 3012
generate_presigned_post() (S3.Client method), 3967
generate_presigned_url() (AutoScaling.Client method), 86
generate_presigned_url() (CloudFormation.Client method), 132
generate_presigned_url() (CloudFront.Client method), 206
generate_presigned_url() (CloudHSM.Client method), 325
generate_presigned_url() (CloudSearch.Client method), 397
generate_presigned_url() (CloudSearchDomain.Client method), 405
generate_presigned_url() (CloudTrail.Client method), 419
generate_presigned_url() (CloudWatch.Client method), 436
generate_presigned_url() (CloudWatchLogs.Client method), 3068
generate_presigned_url() (CodeCommit.Client method), 456
generate_presigned_url() (CodeDeploy.Client method), 477
generate_presigned_url() (CodePipeline.Client method), 522
generate_presigned_url() (CognitoIdentity.Client method), 564
generate_presigned_url() (CognitoSync.Client method), 581
generate_presigned_url() (ConfigService.Client method), 602
generate_presigned_url() (DataPipeline.Client method), 618

[generate_presigned_url\(\)](#) (DeviceFarm.Client method), [647](#)
[generate_presigned_url\(\)](#) (DirectConnect.Client method), [735](#)
[generate_presigned_url\(\)](#) (DirectoryService.Client method), [751](#)
[generate_presigned_url\(\)](#) (DynamoDB.Client method), [833](#)
[generate_presigned_url\(\)](#) (DynamoDBStreams.Client method), [1090](#)
[generate_presigned_url\(\)](#) (EC2.Client method), [1361](#)
[generate_presigned_url\(\)](#) (ECS.Client method), [1829](#)
[generate_presigned_url\(\)](#) (EFS.Client method), [1894](#)
[generate_presigned_url\(\)](#) (ElastiCache.Client method), [1972](#)
[generate_presigned_url\(\)](#) (ElasticBeanstalk.Client method), [2078](#)
[generate_presigned_url\(\)](#) (ElasticLoadBalancing.Client method), [2566](#)
[generate_presigned_url\(\)](#) (ElasticTranscoder.Client method), [2225](#)
[generate_presigned_url\(\)](#) (EMR.Client method), [2600](#)
[generate_presigned_url\(\)](#) (Glacier.Client method), [2647](#)
[generate_presigned_url\(\)](#) (IAM.Client method), [2743](#)
[generate_presigned_url\(\)](#) (ImportExport.Client method), [2973](#)
[generate_presigned_url\(\)](#) (Kinesis.Client method), [2985](#)
[generate_presigned_url\(\)](#) (KMS.Client method), [3013](#)
[generate_presigned_url\(\)](#) (Lambda.Client method), [3032](#)
[generate_presigned_url\(\)](#) (MachineLearning.Client method), [3120](#)
[generate_presigned_url\(\)](#) (OpsWorks.Client method), [3233](#)
[generate_presigned_url\(\)](#) (RDS.Client method), [3442](#)
[generate_presigned_url\(\)](#) (Redshift.Client method), [3710](#)
[generate_presigned_url\(\)](#) (Route53.Client method), [3831](#)
[generate_presigned_url\(\)](#) (Route53Domains.Client method), [3882](#)
[generate_presigned_url\(\)](#) (S3.Client method), [3967](#)
[generate_presigned_url\(\)](#) (SES.Client method), [4143](#)
[generate_presigned_url\(\)](#) (SimpleDB.Client method), [4134](#)
[generate_presigned_url\(\)](#) (SNS.Client method), [4171](#)
[generate_presigned_url\(\)](#) (SQS.Client method), [4223](#)
[generate_presigned_url\(\)](#) (SSM.Client method), [4267](#)
[generate_presigned_url\(\)](#) (StorageGateway.Client method), [4306](#)
[generate_presigned_url\(\)](#) (STS.Client method), [4342](#)
[generate_presigned_url\(\)](#) (Support.Client method), [4366](#)
[generate_presigned_url\(\)](#) (SWF.Client method), [4396](#)
[generate_presigned_url\(\)](#) (WorkSpaces.Client method), [4630](#)
[generate_random\(\)](#) (KMS.Client method), [3013](#)
[get\(\)](#) (S3.Object method), [4094](#)
[get\(\)](#) (S3.ObjectSummary method), [4111](#)
[get\(\)](#) (S3.ObjectVersion method), [4123](#)
[get_access_key_last_used\(\)](#) (IAM.Client method), [2743](#)
[get_account_authorization_details\(\)](#) (IAM.Client method), [2744](#)
[get_account_password_policy\(\)](#) (IAM.Client method), [2754](#)
[get_account_settings\(\)](#) (DeviceFarm.Client method), [647](#)
[get_account_summary\(\)](#) (IAM.Client method), [2755](#)
[get_application\(\)](#) (CodeDeploy.Client method), [477](#)
[get_application_revision\(\)](#) (CodeDeploy.Client method), [478](#)
[get_attributes\(\)](#) (boto3.resources.model.ResourceModel method), [4640](#)
[get_attributes\(\)](#) (SimpleDB.Client method), [4134](#)
[get_available_resources\(\)](#) (boto3.session.Session method), [4646](#)
[get_available_services\(\)](#) (boto3.session.Session method), [4646](#)
[get_batch_prediction\(\)](#) (MachineLearning.Client method), [3120](#)
[get_branch\(\)](#) (CodeCommit.Client method), [456](#)
[get_bucket_acl\(\)](#) (S3.Client method), [3968](#)
[get_bucket_cors\(\)](#) (S3.Client method), [3968](#)
[get_bucket_lifecycle\(\)](#) (S3.Client method), [3969](#)
[get_bucket_lifecycle_configuration\(\)](#) (S3.Client method), [3971](#)
[get_bucket_location\(\)](#) (S3.Client method), [3973](#)
[get_bucket_logging\(\)](#) (S3.Client method), [3973](#)
[get_bucket_notification\(\)](#) (S3.Client method), [3974](#)
[get_bucket_notification_configuration\(\)](#) (S3.Client method), [3976](#)
[get_bucket_policy\(\)](#) (S3.Client method), [3979](#)
[get_bucket_replication\(\)](#) (S3.Client method), [3979](#)
[get_bucket_request_payment\(\)](#) (S3.Client method), [3980](#)
[get_bucket_tagging\(\)](#) (S3.Client method), [3980](#)
[get_bucket_versioning\(\)](#) (S3.Client method), [3981](#)
[get_bucket_website\(\)](#) (S3.Client method), [3981](#)
[get_bulk_publish_details\(\)](#) (CognitoSync.Client method), [581](#)
[get_change\(\)](#) (Route53.Client method), [3831](#)
[get_checker_ip_ranges\(\)](#) (Route53.Client method), [3832](#)
[get_cloud_front_origin_access_identity\(\)](#) (CloudFront.Client method), [206](#)
[get_cloud_front_origin_access_identity_config\(\)](#) (CloudFront.Client method), [207](#)
[get_cognito_events\(\)](#) (CognitoSync.Client method), [582](#)
[get_config\(\)](#) (CloudHSM.Client method), [325](#)
[get_console_output\(\)](#) (EC2.Client method), [1362](#)
[get_context_keys_for_custom_policy\(\)](#) (IAM.Client method), [2757](#)
[get_context_keys_for_principal_policy\(\)](#) (IAM.Client method), [2758](#)
[get_credential_report\(\)](#) (IAM.Client method), [2759](#)

`get_credentials_for_identity()` (CognitoIdentity.Client method), 565

`get_data_member()` (in module `boto3.resources.params`), 4642

`get_data_retrieval_policy()` (Glacier.Client method), 2647

`get_data_source()` (MachineLearning.Client method), 3122

`get_deployment()` (CodeDeploy.Client method), 481

`get_deployment_config()` (CodeDeploy.Client method), 485

`get_deployment_group()` (CodeDeploy.Client method), 487

`get_deployment_instance()` (CodeDeploy.Client method), 490

`get_device()` (DeviceFarm.Client method), 648

`get_device_pool()` (DeviceFarm.Client method), 650

`get_device_pool_compatibility()` (DeviceFarm.Client method), 651

`get_directory_limits()` (DirectoryService.Client method), 751

`get_distribution()` (CloudFront.Client method), 208

`get_distribution_config()` (CloudFront.Client method), 223

`get_document()` (SSM.Client method), 4267

`get_domain_detail()` (Route53Domains.Client method), 3882

`get_endpoint_attributes()` (SNS.Client method), 4171

`get_evaluation()` (MachineLearning.Client method), 3125

`get_event_source_mapping()` (Lambda.Client method), 3032

`get_federation_token()` (STS.Client method), 4342

`get_file_size()` (`boto3.s3.transfer.OSUtils` method), 4653

`get_function()` (Lambda.Client method), 3033

`get_function_configuration()` (Lambda.Client method), 3035

`get_geo_location()` (Route53.Client method), 3833

`get_group()` (IAM.Client method), 2759

`get_group_policy()` (IAM.Client method), 2762

`get_health_check()` (Route53.Client method), 3834

`get_health_check_count()` (Route53.Client method), 3837

`get_health_check_last_failure_reason()` (Route53.Client method), 3837

`get_health_check_status()` (Route53.Client method), 3838

`get_hosted_zone()` (Route53.Client method), 3839

`get_hosted_zone_count()` (Route53.Client method), 3841

`get_hostname_suggestion()` (OpsWorks.Client method), 3233

`get_id()` (CognitoIdentity.Client method), 566

`get_identity_dkim_attributes()` (SES.Client method), 4144

`get_identity_notification_attributes()` (SES.Client method), 4145

`get_identity_policies()` (SES.Client method), 4146

`get_identity_pool_configuration()` (CognitoSync.Client method), 583

`get_identity_pool_roles()` (CognitoIdentity.Client method), 566

`get_identity_verification_attributes()` (SES.Client method), 4147

`get_instance_profile()` (IAM.Client method), 2763

`get_invalidation()` (CloudFront.Client method), 236

`get_item()` (DynamoDB.Client method), 834

`get_item()` (DynamoDB.Table method), 1023

`get_job()` (DeviceFarm.Client method), 656

`get_job_details()` (CodePipeline.Client method), 522

`get_job_output()` (Glacier.Client method), 2648

`get_key_policy()` (KMS.Client method), 3014

`get_key_rotation_status()` (KMS.Client method), 3014

`get_log_events()` (CloudWatchLogs.Client method), 3068

`get_login_profile()` (IAM.Client method), 2764

`get_metric_statistics()` (CloudWatch.Client method), 436

`get_ml_model()` (MachineLearning.Client method), 3127

`get_object()` (S3.Client method), 3983

`get_object_acl()` (S3.Client method), 3986

`get_object_torrent()` (S3.Client method), 3987

`get_on_premises_instance()` (CodeDeploy.Client method), 492

`get_open_id_connect_provider()` (IAM.Client method), 2765

`get_open_id_token()` (CognitoIdentity.Client method), 567

`get_open_id_token_for_developer_identity()` (CognitoIdentity.Client method), 568

`get_operation_detail()` (Route53Domains.Client method), 3898

`get_output()` (Glacier.Job method), 2687

`get_paginator()` (AutoScaling.Client method), 86

`get_paginator()` (CloudFormation.Client method), 132

`get_paginator()` (CloudFront.Client method), 237

`get_paginator()` (CloudHSM.Client method), 326

`get_paginator()` (CloudSearch.Client method), 398

`get_paginator()` (CloudSearchDomain.Client method), 405

`get_paginator()` (CloudTrail.Client method), 420

`get_paginator()` (CloudWatch.Client method), 438

`get_paginator()` (CloudWatchLogs.Client method), 3069

`get_paginator()` (CodeCommit.Client method), 457

`get_paginator()` (CodeDeploy.Client method), 493

`get_paginator()` (CodePipeline.Client method), 527

`get_paginator()` (CognitoIdentity.Client method), 569

`get_paginator()` (CognitoSync.Client method), 584

`get_paginator()` (ConfigService.Client method), 602

`get_paginator()` (DataPipeline.Client method), 618

`get_paginator()` (DeviceFarm.Client method), 660

`get_paginator()` (DirectConnect.Client method), 735

`get_paginator()` (DirectoryService.Client method), 752

`get_paginator()` (DynamoDB.Client method), 840

[get_paginator\(\)](#) (DynamoDBStreams.Client method), 1090
[get_paginator\(\)](#) (EC2.Client method), 1362
[get_paginator\(\)](#) (ECS.Client method), 1829
[get_paginator\(\)](#) (EFS.Client method), 1894
[get_paginator\(\)](#) (ElastiCache.Client method), 1972
[get_paginator\(\)](#) (ElasticBeanstalk.Client method), 2079
[get_paginator\(\)](#) (ElasticLoadBalancing.Client method), 2566
[get_paginator\(\)](#) (ElasticTranscoder.Client method), 2226
[get_paginator\(\)](#) (EMR.Client method), 2600
[get_paginator\(\)](#) (Glacier.Client method), 2650
[get_paginator\(\)](#) (IAM.Client method), 2766
[get_paginator\(\)](#) (ImportExport.Client method), 2973
[get_paginator\(\)](#) (Kinesis.Client method), 2985
[get_paginator\(\)](#) (KMS.Client method), 3015
[get_paginator\(\)](#) (Lambda.Client method), 3036
[get_paginator\(\)](#) (MachineLearning.Client method), 3131
[get_paginator\(\)](#) (OpsWorks.Client method), 3233
[get_paginator\(\)](#) (RDS.Client method), 3443
[get_paginator\(\)](#) (Redshift.Client method), 3711
[get_paginator\(\)](#) (Route53.Client method), 3841
[get_paginator\(\)](#) (Route53Domains.Client method), 3899
[get_paginator\(\)](#) (S3.Client method), 3988
[get_paginator\(\)](#) (SES.Client method), 4148
[get_paginator\(\)](#) (SimpleDB.Client method), 4135
[get_paginator\(\)](#) (SNS.Client method), 4172
[get_paginator\(\)](#) (SQS.Client method), 4223
[get_paginator\(\)](#) (SSM.Client method), 4268
[get_paginator\(\)](#) (StorageGateway.Client method), 4306
[get_paginator\(\)](#) (STS.Client method), 4345
[get_paginator\(\)](#) (Support.Client method), 4367
[get_paginator\(\)](#) (SWF.Client method), 4397
[get_paginator\(\)](#) (WorkSpaces.Client method), 4630
[get_password_data\(\)](#) (EC2.Client method), 1363
[get_pipeline\(\)](#) (CodePipeline.Client method), 527
[get_pipeline_definition\(\)](#) (DataPipeline.Client method), 618
[get_pipeline_state\(\)](#) (CodePipeline.Client method), 531
[get_platform_application_attributes\(\)](#) (SNS.Client method), 4172
[get_policy\(\)](#) (IAM.Client method), 2766
[get_policy\(\)](#) (Lambda.Client method), 3037
[get_policy_version\(\)](#) (IAM.Client method), 2768
[get_project\(\)](#) (DeviceFarm.Client method), 660
[get_queue_attributes\(\)](#) (SQS.Client method), 4224
[get_queue_by_name\(\)](#) (SQS.ServiceResource method), 4239
[get_queue_url\(\)](#) (SQS.Client method), 4225
[get_records\(\)](#) (DynamoDBStreams.Client method), 1090
[get_records\(\)](#) (Kinesis.Client method), 2985
[get_repository\(\)](#) (CodeCommit.Client method), 457
[get_resource_config_history\(\)](#) (ConfigService.Client method), 602
[get_reusable_delegation_set\(\)](#) (Route53.Client method), 3842
[get_role\(\)](#) (IAM.Client method), 2769
[get_role_policy\(\)](#) (IAM.Client method), 2770
[get_run\(\)](#) (DeviceFarm.Client method), 661
[get_saml_provider\(\)](#) (IAM.Client method), 2771
[get_send_quota\(\)](#) (SES.Client method), 4149
[get_send_statistics\(\)](#) (SES.Client method), 4149
[get_server_certificate\(\)](#) (IAM.Client method), 2772
[get_session_token\(\)](#) (STS.Client method), 4345
[get_shard_iterator\(\)](#) (DynamoDBStreams.Client method), 1097
[get_shard_iterator\(\)](#) (Kinesis.Client method), 2987
[get_shipping_label\(\)](#) (ImportExport.Client method), 2974
[get_snapshot_limits\(\)](#) (DirectoryService.Client method), 752
[get_ssh_public_key\(\)](#) (IAM.Client method), 2773
[get_stack_policy\(\)](#) (CloudFormation.Client method), 132
[get_status\(\)](#) (ImportExport.Client method), 2975
[get_streaming_distribution\(\)](#) (CloudFront.Client method), 238
[get_streaming_distribution_config\(\)](#) (CloudFront.Client method), 241
[get_subscription_attributes\(\)](#) (SNS.Client method), 4173
[get_suite\(\)](#) (DeviceFarm.Client method), 663
[get_template\(\)](#) (CloudFormation.Client method), 133
[get_template_summary\(\)](#) (CloudFormation.Client method), 133
[get_test\(\)](#) (DeviceFarm.Client method), 666
[get_third_party_job_details\(\)](#) (CodePipeline.Client method), 533
[get_topic_attributes\(\)](#) (SNS.Client method), 4174
[get_trail_status\(\)](#) (CloudTrail.Client method), 420
[get_upload\(\)](#) (DeviceFarm.Client method), 668
[get_user\(\)](#) (IAM.Client method), 2774
[get_user_policy\(\)](#) (IAM.Client method), 2776
[get_vault_access_policy\(\)](#) (Glacier.Client method), 2650
[get_vault_lock\(\)](#) (Glacier.Client method), 2651
[get_vault_notifications\(\)](#) (Glacier.Client method), 2652
[get_waiter\(\)](#) (AutoScaling.Client method), 86
[get_waiter\(\)](#) (CloudFormation.Client method), 135
[get_waiter\(\)](#) (CloudFront.Client method), 243
[get_waiter\(\)](#) (CloudHSM.Client method), 326
[get_waiter\(\)](#) (CloudSearch.Client method), 398
[get_waiter\(\)](#) (CloudSearchDomain.Client method), 405
[get_waiter\(\)](#) (CloudTrail.Client method), 421
[get_waiter\(\)](#) (CloudWatch.Client method), 439
[get_waiter\(\)](#) (CloudWatchLogs.Client method), 3070
[get_waiter\(\)](#) (CodeCommit.Client method), 459
[get_waiter\(\)](#) (CodeDeploy.Client method), 493
[get_waiter\(\)](#) (CodePipeline.Client method), 538
[get_waiter\(\)](#) (CognitoIdentity.Client method), 569
[get_waiter\(\)](#) (CognitoSync.Client method), 584
[get_waiter\(\)](#) (ConfigService.Client method), 605

`get_waiter()` (DataPipeline.Client method), 620
`get_waiter()` (DeviceFarm.Client method), 669
`get_waiter()` (DirectConnect.Client method), 735
`get_waiter()` (DirectoryService.Client method), 753
`get_waiter()` (DynamoDB.Client method), 840
`get_waiter()` (DynamoDBStreams.Client method), 1098
`get_waiter()` (EC2.Client method), 1364
`get_waiter()` (ECS.Client method), 1829
`get_waiter()` (EFS.Client method), 1894
`get_waiter()` (ElastiCache.Client method), 1972
`get_waiter()` (ElasticBeanstalk.Client method), 2079
`get_waiter()` (ElasticLoadBalancing.Client method), 2566
`get_waiter()` (ElasticTranscoder.Client method), 2226
`get_waiter()` (EMR.Client method), 2600
`get_waiter()` (Glacier.Client method), 2653
`get_waiter()` (IAM.Client method), 2776
`get_waiter()` (ImportExport.Client method), 2976
`get_waiter()` (Kinesis.Client method), 2988
`get_waiter()` (KMS.Client method), 3015
`get_waiter()` (Lambda.Client method), 3037
`get_waiter()` (MachineLearning.Client method), 3131
`get_waiter()` (OpsWorks.Client method), 3234
`get_waiter()` (RDS.Client method), 3443
`get_waiter()` (Redshift.Client method), 3711
`get_waiter()` (Route53.Client method), 3842
`get_waiter()` (Route53Domains.Client method), 3899
`get_waiter()` (S3.Client method), 3988
`get_waiter()` (SES.Client method), 4150
`get_waiter()` (SimpleDB.Client method), 4135
`get_waiter()` (SNS.Client method), 4175
`get_waiter()` (SQS.Client method), 4226
`get_waiter()` (SSM.Client method), 4268
`get_waiter()` (StorageGateway.Client method), 4306
`get_waiter()` (STS.Client method), 4347
`get_waiter()` (Support.Client method), 4367
`get_waiter()` (SWF.Client method), 4397
`get_waiter()` (WorkSpaces.Client method), 4630
`get_workflow_execution_history()` (SWF.Client method), 4397

`Glacier.Account` (built-in class), 2679
`Glacier.Archive` (built-in class), 2681
`Glacier.Client` (built-in class), 2634
`Glacier.Job` (built-in class), 2684
`Glacier.MultipartUpload` (built-in class), 2689
`Glacier.Notification` (built-in class), 2695
`Glacier.ServiceResource` (built-in class), 2675
`Glacier.Vault` (built-in class), 2698
`global_secondary_indexes` (DynamoDB.Table attribute), 999

`grant_access()` (OpsWorks.Client method), 3234
`grants` (S3.BucketAcl attribute), 4055
`grants` (S3.ObjectAcl attribute), 4105
`Group()` (IAM.GroupPolicy method), 2919
`Group()` (IAM.ServiceResource method), 2881

`group_id` (EC2.SecurityGroup attribute), 1697
`group_id` (IAM.Group attribute), 2910
`group_name` (EC2.PlacementGroup attribute), 1678
`group_name` (EC2.SecurityGroup attribute), 1697
`group_name` (IAM.Group attribute), 2910
`group_name` (IAM.GroupPolicy attribute), 2917
`GroupPolicy()` (IAM.ServiceResource method), 2881
`groups` (EC2.NetworkInterface attribute), 1670
`groups` (IAM.ServiceResource attribute), 2884
`groups` (IAM.User attribute), 2963
`gt()` (boto3.dynamodb.conditions.Attr method), 4649
`gt()` (boto3.dynamodb.conditions.Key method), 4648
`gte()` (boto3.dynamodb.conditions.Attr method), 4649
`gte()` (boto3.dynamodb.conditions.Key method), 4648

H

`handle_response_item()` (boto3.resources.response.ResourceHandler method), 4643
`hard_expiry` (IAM.AccountPasswordPolicy attribute), 2899

`head()` (S3.ObjectVersion method), 4126
`head_bucket()` (S3.Client method), 3988
`head_object()` (S3.Client method), 3988
`hostname_theme` (OpsWorks.Stack attribute), 3270
`hypervisor` (EC2.Image attribute), 1602
`hypervisor` (EC2.Instance attribute), 1613

I

`IAM.AccessKey` (built-in class), 2894
`IAM.AccessKeyPair` (built-in class), 2896
`IAM.AccountPasswordPolicy` (built-in class), 2898
`IAM.AccountSummary` (built-in class), 2901
`IAM.AssumeRolePolicy` (built-in class), 2903
`IAM.Client` (built-in class), 2713
`IAM.CurrentUser` (built-in class), 2904
`IAM.Group` (built-in class), 2909
`IAM.GroupPolicy` (built-in class), 2917
`IAM.InstanceProfile` (built-in class), 2919
`IAM.LoginProfile` (built-in class), 2922
`IAM.MfaDevice` (built-in class), 2924
`IAM.Paginator.GetGroup` (built-in class), 2840
`IAM.Paginator.ListAccessKeys` (built-in class), 2842
`IAM.Paginator.ListAccountAliases` (built-in class), 2844
`IAM.Paginator.ListGroupPolicies` (built-in class), 2845
`IAM.Paginator.ListGroups` (built-in class), 2846
`IAM.Paginator.ListGroupsForUser` (built-in class), 2848
`IAM.Paginator.ListInstanceProfiles` (built-in class), 2850
`IAM.Paginator.ListInstanceProfilesForRole` (built-in class), 2852
`IAM.Paginator.ListMFADevices` (built-in class), 2855
`IAM.Paginator.ListPolicies` (built-in class), 2856
`IAM.Paginator.ListRolePolicies` (built-in class), 2859
`IAM.Paginator.ListRoles` (built-in class), 2860

- IAM.Paginator.ListServerCertificates (built-in class), 2862
- IAM.Paginator.ListSigningCertificates (built-in class), 2864
- IAM.Paginator.ListUserPolicies (built-in class), 2865
- IAM.Paginator.ListUsers (built-in class), 2867
- IAM.Paginator.ListVirtualMFADevices (built-in class), 2869
- IAM.Policy (built-in class), 2926
- IAM.PolicyVersion (built-in class), 2936
- IAM.Role (built-in class), 2938
- IAM.RolePolicy (built-in class), 2945
- IAM.SamlProvider (built-in class), 2947
- IAM.ServerCertificate (built-in class), 2949
- IAM.ServiceResource (built-in class), 2872
- IAM.SigningCertificate (built-in class), 2951
- IAM.User (built-in class), 2953
- IAM.UserPolicy (built-in class), 2967
- IAM.VirtualMfaDevice (built-in class), 2969
- IAM.Waiter.InstanceProfileExists (built-in class), 2871
- IAM.Waiter.UserExists (built-in class), 2872
- iam_instance_profile (EC2.Instance attribute), 1613
- id (CloudFormation.Event attribute), 157
- id (EC2.DhcpOptions attribute), 1598
- id (EC2.Image attribute), 1602
- id (EC2.Instance attribute), 1612
- id (EC2.InternetGateway attribute), 1658
- id (EC2.NetworkAcl attribute), 1663
- id (EC2.NetworkInterface attribute), 1669
- id (EC2.RouteTable attribute), 1689
- id (EC2.RouteTableAssociation attribute), 1695
- id (EC2.SecurityGroup attribute), 1697
- id (EC2.Snapshot attribute), 1710
- id (EC2.Subnet attribute), 1720
- id (EC2.Volume attribute), 1743
- id (EC2.Vpc attribute), 1757
- id (EC2.VpcPeeringConnection attribute), 1787
- id (Glacier.Account attribute), 2679
- id (Glacier.Archive attribute), 2681
- id (Glacier.Job attribute), 2685
- id (Glacier.MultipartUpload attribute), 2690
- id (IAM.AccessKey attribute), 2895
- id (IAM.AccessKeyPair attribute), 2897
- id (IAM.SigningCertificate attribute), 2952
- id (OpsWorks.Layer attribute), 3265
- id (OpsWorks.Stack attribute), 3269
- id (S3.MultipartUpload attribute), 4080
- id (S3.ObjectVersion attribute), 4122
- Identifier (class in boto3.resources.model), 4639
- identifiers (boto3.resources.base.ResourceMeta attribute), 4645
- identifiers (boto3.resources.model.ResourceModel attribute), 4640
- identifiers (boto3.resources.model.ResponseResource attribute), 4641
- image (EC2.Instance attribute), 1615
- Image() (EC2.ServiceResource method), 1553
- image_id (EC2.Image attribute), 1602
- image_id (EC2.Instance attribute), 1613
- image_location (EC2.Image attribute), 1602
- image_owner_alias (EC2.Image attribute), 1602
- image_type (EC2.Image attribute), 1602
- images (EC2.ServiceResource attribute), 1557
- import_image() (EC2.Client method), 1364
- import_instance() (EC2.Client method), 1367
- import_key_pair() (EC2.Client method), 1373
- import_key_pair() (EC2.ServiceResource method), 1549
- import_snapshot() (EC2.Client method), 1374
- import_volume() (EC2.Client method), 1376
- ImportExport.Client (built-in class), 2971
- ImportExport.Paginator.ListJobs (built-in class), 2978
- index_document (S3.BucketWebsite attribute), 4076
- index_documents() (CloudSearch.Client method), 398
- initiate_archive_retrieval() (Glacier.Archive method), 2682
- initiate_inventory_retrieval() (Glacier.Vault method), 2701
- initiate_job() (Glacier.Client method), 2653
- initiate_multipart_upload() (Glacier.Client method), 2657
- initiate_multipart_upload() (Glacier.Vault method), 2702
- initiate_multipart_upload() (S3.Object method), 4097
- initiate_multipart_upload() (S3.ObjectSummary method), 4114
- initiate_vault_lock() (Glacier.Client method), 2658
- initiated (S3.MultipartUpload attribute), 4080
- initiator (S3.MultipartUpload attribute), 4080
- inject_bucket_methods() (in module boto3.s3.inject), 4650
- inject_object_methods() (in module boto3.s3.inject), 4650
- inject_s3_transfer_methods() (in module boto3.s3.inject), 4650
- install_updates_on_boot (OpsWorks.Layer attribute), 3266
- Instance() (EC2.ServiceResource method), 1553
- instance_id (EC2.Instance attribute), 1613
- instance_lifecycle (EC2.Instance attribute), 1613
- instance_profile_id (IAM.InstanceProfile attribute), 2920
- instance_profile_name (IAM.InstanceProfile attribute), 2920
- instance_profiles (IAM.Role attribute), 2942
- instance_profiles (IAM.ServiceResource attribute), 2885
- instance_tenancy (EC2.Vpc attribute), 1757
- instance_type (EC2.Instance attribute), 1613
- InstanceProfile() (IAM.ServiceResource method), 2881
- instances (EC2.PlacementGroup attribute), 1679
- instances (EC2.ServiceResource attribute), 1560

instances (EC2.Subnet attribute), 1730
instances (EC2.Vpc attribute), 1769
instances_count (OpsWorks.StackSummary attribute), 3277
internet_gateway_id (EC2.InternetGateway attribute), 1658
internet_gateways (EC2.ServiceResource attribute), 1573
internet_gateways (EC2.Vpc attribute), 1778
InternetGateway() (EC2.ServiceResource method), 1553
inventory_retrieval_parameters (Glacier.Job attribute), 2686
inventory_size_in_bytes (Glacier.Job attribute), 2686
invoke() (Lambda.Client method), 3037
invoke_async() (Lambda.Client method), 3039
iops (EC2.Volume attribute), 1743
ip_permissions (EC2.SecurityGroup attribute), 1697
ip_permissions_egress (EC2.SecurityGroup attribute), 1698
is_attachable (IAM.Policy attribute), 2927
is_default (EC2.NetworkAcl attribute), 1663
is_default (EC2.Vpc attribute), 1758
is_default_version (IAM.PolicyVersion attribute), 2937
is_in() (boto3.dynamodb.conditions.Attr method), 4649
is_latest (S3.ObjectVersion attribute), 4122
item_count (DynamoDB.Table attribute), 1000
iterator() (boto3.resources.collection.CollectionManager method), 4636

J

Job() (Glacier.ServiceResource method), 2677
Job() (Glacier.Vault method), 2705
job_description (Glacier.Job attribute), 2686
job_id (Glacier.Job attribute), 2686
jobs (Glacier.Vault attribute), 2707
jobs_in_progress (Glacier.Vault attribute), 2708

K

kernel_id (EC2.Image attribute), 1602
kernel_id (EC2.Instance attribute), 1613
Key (class in boto3.dynamodb.conditions), 4648
key (EC2.Tag attribute), 1741
key (S3.MultipartUpload attribute), 4080
key (S3.Object attribute), 4089
key (S3.ObjectSummary attribute), 4108
key (S3.ObjectVersion attribute), 4122
key_fingerprint (EC2.KeyPair attribute), 1661
key_name (EC2.Instance attribute), 1613
key_name (EC2.KeyPair attribute), 1661
key_pair (EC2.Instance attribute), 1615
key_pairs (EC2.ServiceResource attribute), 1575
key_schema (DynamoDB.Table attribute), 1000
KeyPair() (EC2.ServiceResource method), 1554
Kinesis.Client (built-in class), 2980
Kinesis.Paginator.DescribeStream (built-in class), 2997

Kinesis.Paginator.ListStreams (built-in class), 2999
Kinesis.Waiter.StreamExists (built-in class), 3001
KMS.Client (built-in class), 3001
kms_key_id (EC2.Snapshot attribute), 1710
kms_key_id (EC2.Volume attribute), 1744

L

Lambda.Client (built-in class), 3024
Lambda.Paginator.ListEventSourceMappings (built-in class), 3048
Lambda.Paginator.ListFunctions (built-in class), 3050
lambda_function_configurations (S3.BucketNotification attribute), 4064
last_inventory_date (Glacier.Vault attribute), 2699
last_modified (S3.MultipartUploadPart attribute), 4084
last_modified (S3.Object attribute), 4089
last_modified (S3.ObjectSummary attribute), 4108
last_modified (S3.ObjectVersion attribute), 4122
last_updated_time (CloudFormation.Stack attribute), 159
last_updated_timestamp (CloudFormation.StackResource attribute), 166
last_updated_timestamp (CloudFormation.StackResourceSummary attribute), 168
latest_stream_arn (DynamoDB.Table attribute), 1000
latest_stream_label (DynamoDB.Table attribute), 1000
launch_time (EC2.Instance attribute), 1613
Layer() (OpsWorks.ServiceResource method), 3263
layer_id (OpsWorks.Layer attribute), 3267
layers (OpsWorks.Stack attribute), 3275
layers_count (OpsWorks.StackSummary attribute), 3277
Lifecycle() (S3.Bucket method), 4046
lifecycle_event_configuration (OpsWorks.Layer attribute), 3267
limit() (boto3.resources.collection.CollectionManager method), 4636
limit() (boto3.resources.collection.ResourceCollection method), 4638
limit() (CloudFormation.ServiceResource method), 156
limit() (CloudFormation.Stack method), 164, 165
limit() (DynamoDB.ServiceResource method), 998
limit() (EC2.Instance method), 1641
limit() (EC2.PlacementGroup method), 1680
limit() (EC2.RouteTable method), 1694
limit() (EC2.ServiceResource method), 1557, 1560, 1565, 1574, 1576, 1578, 1581, 1582, 1585, 1587, 1589, 1591, 1594, 1596, 1597
limit() (EC2.Subnet method), 1732, 1740
limit() (EC2.Volume method), 1756
limit() (EC2.Vpc method), 1768, 1770, 1779–1783, 1785, 1786
limit() (Glacier.Account method), 2680
limit() (Glacier.ServiceResource method), 2678
limit() (Glacier.Vault method), 2706–2711
limit() (IAM.CurrentUser method), 2906–2908

- `limit()` (IAM.Group method), 2914–2916
- `limit()` (IAM.Policy method), 2932–2934, 2936
- `limit()` (IAM.Role method), 2942–2944
- `limit()` (IAM.ServiceResource method), 2885, 2886, 2888–2892, 2894
- `limit()` (IAM.User method), 2961–2964, 2966, 2967
- `limit()` (OpsWorks.ServiceResource method), 3264
- `limit()` (OpsWorks.Stack method), 3276
- `limit()` (S3.Bucket method), 4048, 4051, 4053
- `limit()` (S3.MultipartUpload method), 4082
- `limit()` (S3.ServiceResource method), 4040
- `limit()` (SNS.PlatformApplication method), 4201
- `limit()` (SNS.ServiceResource method), 4195–4197
- `limit()` (SNS.Topic method), 4213
- `limit()` (SQS.Queue method), 4259
- `limit()` (SQS.ServiceResource method), 4241
- `list_access_keys()` (IAM.Client method), 2776
- `list_account_aliases()` (IAM.Client method), 2778
- `list_action_types()` (CodePipeline.Client method), 538
- `list_activity_types()` (SWF.Client method), 4438
- `list_aliases()` (KMS.Client method), 3015
- `list_application_revisions()` (CodeDeploy.Client method), 493
- `list_applications()` (CodeDeploy.Client method), 496
- `list_artifacts()` (DeviceFarm.Client method), 669
- `list_associations()` (SSM.Client method), 4268
- `list_attached_group_policies()` (IAM.Client method), 2779
- `list_attached_role_policies()` (IAM.Client method), 2781
- `list_attached_user_policies()` (IAM.Client method), 2782
- `list_available_solution_stacks()` (ElasticBeanstalk.Client method), 2079
- `list_available_zones()` (CloudHSM.Client method), 326
- `list_bootstrap_actions()` (EMR.Client method), 2600
- `list_branches()` (CodeCommit.Client method), 459
- `list_buckets()` (S3.Client method), 3991
- `list_closed_workflow_executions()` (SWF.Client method), 4440
- `list_cloud_front_origin_access_identities()` (CloudFront.Client method), 243
- `list_clusters()` (ECS.Client method), 1830
- `list_clusters()` (EMR.Client method), 2601
- `list_container_instances()` (ECS.Client method), 1830
- `list_datasets()` (CognitoSync.Client method), 584
- `list_dead_letter_source_queues()` (SQS.Client method), 4226
- `list_deployment_configs()` (CodeDeploy.Client method), 496
- `list_deployment_groups()` (CodeDeploy.Client method), 497
- `list_deployment_instances()` (CodeDeploy.Client method), 498
- `list_deployments()` (CodeDeploy.Client method), 499
- `list_device_pools()` (DeviceFarm.Client method), 671
- `list_devices()` (DeviceFarm.Client method), 673
- `list_discovered_resources()` (ConfigService.Client method), 605
- `list_distributions()` (CloudFront.Client method), 245
- `list_documents()` (SSM.Client method), 4269
- `list_domain_names()` (CloudSearch.Client method), 399
- `list_domains()` (Route53Domains.Client method), 3899
- `list_domains()` (SimpleDB.Client method), 4135
- `list_domains()` (SWF.Client method), 4444
- `list_endpoints_by_platform_application()` (SNS.Client method), 4175
- `list_entities_for_policy()` (IAM.Client method), 2784
- `list_event_source_mappings()` (Lambda.Client method), 3040
- `list_functions()` (Lambda.Client method), 3041
- `list_gateways()` (StorageGateway.Client method), 4306
- `list_geo_locations()` (Route53.Client method), 3843
- `list_grants()` (KMS.Client method), 3016
- `list_group_policies()` (IAM.Client method), 2786
- `list_groups()` (IAM.Client method), 2788
- `list_groups_for_user()` (IAM.Client method), 2789
- `list_hags()` (CloudHSM.Client method), 327
- `list_health_checks()` (Route53.Client method), 3845
- `list_hosted_zones()` (Route53.Client method), 3848
- `list_hosted_zones_by_name()` (Route53.Client method), 3850
- `list_hsms()` (CloudHSM.Client method), 328
- `list_identities()` (CognitoIdentity.Client method), 569
- `list_identities()` (SES.Client method), 4150
- `list_identity_policies()` (SES.Client method), 4151
- `list_identity_pool_usage()` (CognitoSync.Client method), 586
- `list_identity_pools()` (CognitoIdentity.Client method), 570
- `list_instance_groups()` (EMR.Client method), 2603
- `list_instance_profiles()` (IAM.Client method), 2791
- `list_instance_profiles_for_role()` (IAM.Client method), 2793
- `list_instances()` (EMR.Client method), 2606
- `list_invalidations()` (CloudFront.Client method), 258
- `list_jobs()` (DeviceFarm.Client method), 675
- `list_jobs()` (Glacier.Client method), 2659
- `list_jobs()` (ImportExport.Client method), 2976
- `list_jobs_by_pipeline()` (ElasticTranscoder.Client method), 2226
- `list_jobs_by_status()` (ElasticTranscoder.Client method), 2267
- `list_key_policies()` (KMS.Client method), 3018
- `list_keys()` (KMS.Client method), 3019
- `list_local_disks()` (StorageGateway.Client method), 4307
- `list_luna_clients()` (CloudHSM.Client method), 328
- `list_metrics()` (CloudWatch.Client method), 439
- `list_mfa_devices()` (IAM.Client method), 2796
- `list_multipart_uploads()` (Glacier.Client method), 2663

`list_multipart_uploads()` (S3.Client method), 3992
`list_object_versions()` (S3.Client method), 3994
`list_objects()` (S3.Client method), 3996
`list_on_premises_instances()` (CodeDeploy.Client method), 500
`list_open_id_connect_providers()` (IAM.Client method), 2797
`list_open_workflow_executions()` (SWF.Client method), 4446
`list_operations()` (Route53Domains.Client method), 3901
`list_parts()` (Glacier.Client method), 2665
`list_parts()` (S3.Client method), 3998
`list_pipelines()` (CodePipeline.Client method), 541
`list_pipelines()` (DataPipeline.Client method), 620
`list_pipelines()` (ElasticTranscoder.Client method), 2309
`list_platform_applications()` (SNS.Client method), 4176
`list_policies()` (IAM.Client method), 2798
`list_policy_versions()` (IAM.Client method), 2801
`list_presets()` (ElasticTranscoder.Client method), 2316
`list_projects()` (DeviceFarm.Client method), 679
`list_queues()` (SQS.Client method), 4226
`list_records()` (CognitoSync.Client method), 587
`list_repositories()` (CodeCommit.Client method), 459
`list_resource_record_sets()` (Route53.Client method), 3853
`list_reusable_delegation_sets()` (Route53.Client method), 3857
`list_role_policies()` (IAM.Client method), 2802
`list_roles()` (IAM.Client method), 2804
`list_runs()` (DeviceFarm.Client method), 680
`list_saml_providers()` (IAM.Client method), 2806
`list_samples()` (DeviceFarm.Client method), 682
`list_server_certificates()` (IAM.Client method), 2806
`list_services()` (ECS.Client method), 1831
`list_signing_certificates()` (IAM.Client method), 2808
`list_ssh_public_keys()` (IAM.Client method), 2810
`list_stack_resources()` (CloudFormation.Client method), 135
`list_stacks()` (CloudFormation.Client method), 137
`list_steps()` (EMR.Client method), 2608
`list_streaming_distributions()` (CloudFront.Client method), 259
`list_streams()` (DynamoDBStreams.Client method), 1098
`list_streams()` (Kinesis.Client method), 2988
`list_subscriptions()` (SNS.Client method), 4177
`list_subscriptions_by_topic()` (SNS.Client method), 4178
`list_suites()` (DeviceFarm.Client method), 684
`list_tables()` (DynamoDB.Client method), 840
`list_tags_for_domain()` (Route53Domains.Client method), 3902
`list_tags_for_resource()` (ElastiCache.Client method), 1972
`list_tags_for_resource()` (RDS.Client method), 3443
`list_tags_for_resource()` (Route53.Client method), 3859
`list_tags_for_resource()` (StorageGateway.Client method), 4308
`list_tags_for_resources()` (Route53.Client method), 3860
`list_tags_for_stream()` (Kinesis.Client method), 2989
`list_tags_for_vault()` (Glacier.Client method), 2667
`list_task_definition_families()` (ECS.Client method), 1832
`list_task_definitions()` (ECS.Client method), 1833
`list_tasks()` (ECS.Client method), 1835
`list_tests()` (DeviceFarm.Client method), 686
`list_topics()` (SNS.Client method), 4179
`list_unique_problems()` (DeviceFarm.Client method), 689
`list_uploads()` (DeviceFarm.Client method), 692
`list_user_policies()` (IAM.Client method), 2811
`list_users()` (IAM.Client method), 2812
`list_vaults()` (Glacier.Client method), 2668
`list_verified_email_addresses()` (SES.Client method), 4152
`list_virtual_mfa_devices()` (IAM.Client method), 2815
`list_volume_initiators()` (StorageGateway.Client method), 4309
`list_volume_recovery_points()` (StorageGateway.Client method), 4310
`list_volumes()` (StorageGateway.Client method), 4310
`list_workflow_types()` (SWF.Client method), 4450
`load` (boto3.resources.model.ResourceModel attribute), 4640
`load()` (CloudFormation.Stack method), 160
`load()` (DynamoDB.Table method), 1027
`load()` (EC2.DhcpOptions method), 1600
`load()` (EC2.Image method), 1608
`load()` (EC2.Instance method), 1627
`load()` (EC2.InternetGateway method), 1660
`load()` (EC2.KeyPair method), 1662
`load()` (EC2.NetworkAcl method), 1666
`load()` (EC2.NetworkInterface method), 1675
`load()` (EC2.PlacementGroup method), 1678
`load()` (EC2.RouteTable method), 1692
`load()` (EC2.SecurityGroup method), 1704
`load()` (EC2.Snapshot method), 1716
`load()` (EC2.Subnet method), 1730
`load()` (EC2.Tag method), 1742
`load()` (EC2.Volume method), 1754
`load()` (EC2.Vpc method), 1766
`load()` (EC2.VpcPeeringConnection method), 1790
`load()` (Glacier.Job method), 2689
`load()` (Glacier.Notification method), 2696
`load()` (Glacier.Vault method), 2703
`load()` (IAM.AccountPasswordPolicy method), 2900
`load()` (IAM.Group method), 2912
`load()` (IAM.GroupPolicy method), 2918
`load()` (IAM.InstanceProfile method), 2921
`load()` (IAM.LoginProfile method), 2923
`load()` (IAM.Policy method), 2931

- load() (IAM.PolicyVersion method), 2938
 - load() (IAM.Role method), 2940
 - load() (IAM.RolePolicy method), 2946
 - load() (IAM.SamlProvider method), 2948
 - load() (IAM.ServerCertificate method), 2950
 - load() (IAM.User method), 2958
 - load() (IAM.UserPolicy method), 2968
 - load() (OpsWorks.Layer method), 3268
 - load() (OpsWorks.Stack method), 3274
 - load() (S3.Bucket method), 4041
 - load() (S3.BucketAcl method), 4055
 - load() (S3.BucketCors method), 4057
 - load() (S3.BucketLifecycle method), 4059
 - load() (S3.BucketLogging method), 4062
 - load() (S3.BucketNotification method), 4065
 - load() (S3.BucketPolicy method), 4069
 - load() (S3.BucketRequestPayment method), 4070
 - load() (S3.BucketTagging method), 4072
 - load() (S3.BucketVersioning method), 4074
 - load() (S3.BucketWebsite method), 4076
 - load() (S3.Object method), 4099
 - load() (S3.ObjectAcl method), 4105
 - load() (SNS.PlatformApplication method), 4199
 - load() (SNS.PlatformEndpoint method), 4202
 - load() (SNS.Subscription method), 4206
 - load() (SNS.Topic method), 4209
 - load() (SQS.Queue method), 4250
 - load_from_definition() (boto3.resources.collection.CollectionFactory method), 4635
 - load_from_definition() (boto3.resources.factory.ResourceFactory method), 4645
 - load_rename_map() (boto3.resources.model.ResourceModel method), 4640
 - local_secondary_indexes (DynamoDB.Table attribute), 1000
 - Logging() (S3.Bucket method), 4046
 - logging_enabled (S3.BucketLogging attribute), 4062
 - logical_id (CloudFormation.StackResource attribute), 166
 - logical_id (CloudFormation.StackResourceSummary attribute), 168
 - logical_resource_id (CloudFormation.Event attribute), 157
 - logical_resource_id (CloudFormation.StackResource attribute), 166
 - logical_resource_id (CloudFormation.StackResourceSummary attribute), 168
 - LoginProfile() (IAM.ServiceResource method), 2882
 - LoginProfile() (IAM.User method), 2960
 - lookup_developer_identity() (CognitoIdentity.Client method), 571
 - lookup_events() (CloudTrail.Client method), 421
 - lte() (boto3.dynamodb.conditions.Attr method), 4649
 - lte() (boto3.dynamodb.conditions.Key method), 4648
- ## M
- mac_address (EC2.NetworkInterface attribute), 1670
 - MachineLearning.Client (built-in class), 3086
 - MachineLearning.Paginator.DescribeBatchPredictions (built-in class), 3135
 - MachineLearning.Paginator.DescribeDataSources (built-in class), 3138
 - MachineLearning.Paginator.DescribeEvaluations (built-in class), 3143
 - MachineLearning.Paginator.DescribeMLModels (built-in class), 3146
 - MachineLearning.Waiter.BatchPredictionAvailable (built-in class), 3151
 - MachineLearning.Waiter.DataSourceAvailable (built-in class), 3153
 - MachineLearning.Waiter.EvaluationAvailable (built-in class), 3154
 - MachineLearning.Waiter.MLModelAvailable (built-in class), 3155
 - main (EC2.RouteTableAssociation attribute), 1695
 - map_public_ip_on_launch (EC2.Subnet attribute), 1721
 - max_password_age (IAM.AccountPasswordPolicy attribute), 2899
 - md5_of_body (SQS.Message attribute), 4242
 - md5_of_message_attributes (SQS.Message attribute), 4242
 - merge_developer_identities() (CognitoIdentity.Client method), 572
 - merge_shards() (Kinesis.Client method), 2990
 - Message() (SQS.Queue method), 4258
 - Message() (SQS.ServiceResource method), 4240
 - message_attributes (SQS.Message attribute), 4242
 - message_id (SQS.Message attribute), 4242
 - meta (boto3.resources.base.ServiceResource attribute), 4645
 - metadata (CloudFormation.StackResource attribute), 166
 - metadata (S3.Object attribute), 4089
 - mfa_delete (S3.BucketVersioning attribute), 4074
 - mfa_devices (IAM.CurrentUser attribute), 2906
 - mfa_devices (IAM.User attribute), 2964
 - MfaDevice() (IAM.ServiceResource method), 2882
 - MfaDevice() (IAM.User method), 2960
 - minimum_password_length (IAM.AccountPasswordPolicy attribute), 2899
 - missing_meta (S3.Object attribute), 4089
 - model (boto3.resources.model.ResponseResource attribute), 4641
 - modify_attribute() (EC2.Image method), 1608
 - modify_attribute() (EC2.Instance method), 1627
 - modify_attribute() (EC2.NetworkInterface method), 1675

`modify_attribute()` (EC2.Snapshot method), 1716
`modify_attribute()` (EC2.Volume method), 1754
`modify_attribute()` (EC2.Vpc method), 1766
`modify_cache_cluster()` (ElastiCache.Client method), 1973
`modify_cache_parameter_group()` (ElastiCache.Client method), 1983
`modify_cache_subnet_group()` (ElastiCache.Client method), 1984
`modify_cluster()` (Redshift.Client method), 3711
`modify_cluster_parameter_group()` (Redshift.Client method), 3720
`modify_cluster_subnet_group()` (Redshift.Client method), 3722
`modify_db_cluster()` (RDS.Client method), 3444
`modify_db_cluster_parameter_group()` (RDS.Client method), 3449
`modify_db_instance()` (RDS.Client method), 3451
`modify_db_parameter_group()` (RDS.Client method), 3465
`modify_db_subnet_group()` (RDS.Client method), 3467
`modify_event_subscription()` (RDS.Client method), 3468
`modify_event_subscription()` (Redshift.Client method), 3724
`modify_hapg()` (CloudHSM.Client method), 329
`modify_hsm()` (CloudHSM.Client method), 330
`modify_image_attribute()` (EC2.Client method), 1380
`modify_instance_attribute()` (EC2.Client method), 1382
`modify_instance_groups()` (EMR.Client method), 2610
`modify_load_balancer_attributes()` (ElasticLoadBalancing.Client method), 2566
`modify_luna_client()` (CloudHSM.Client method), 330
`modify_mount_target_security_groups()` (EFS.Client method), 1894
`modify_network_interface_attribute()` (EC2.Client method), 1385
`modify_option_group()` (RDS.Client method), 3470
`modify_replication_group()` (ElastiCache.Client method), 1985
`modify_reserved_instances()` (EC2.Client method), 1386
`modify_snapshot_attribute()` (EC2.Client method), 1387
`modify_snapshot_copy_retention_period()` (Redshift.Client method), 3726
`modify_subnet_attribute()` (EC2.Client method), 1388
`modify_volume_attribute()` (EC2.Client method), 1389
`modify_vpc_attribute()` (EC2.Client method), 1389
`modify_vpc_endpoint()` (EC2.Client method), 1390
`monitor()` (EC2.Instance method), 1630
`monitor()` (EC2.PlacementGroup method), 1681
`monitor()` (EC2.ServiceResource method), 1566
`monitor()` (EC2.Subnet method), 1732
`monitor()` (EC2.Vpc method), 1771
`monitor_instances()` (EC2.Client method), 1391
`monitoring` (EC2.Instance attribute), 1613

`move_address_to_vpc()` (EC2.Client method), 1392
`multipart_uploads` (Glacier.Vault attribute), 2710
`multipart_upload_id` (Glacier.MultipartUpload attribute), 2690
`multipart_upload_id` (S3.MultipartUploadPart attribute), 4084
`multipart_uploads` (S3.Bucket attribute), 4047
`MultipartDownloader` (class in boto3.s3.transfer), 4653
`MultipartUpload()` (Glacier.ServiceResource method), 2677
`MultipartUpload()` (Glacier.Vault method), 2705
`MultipartUpload()` (S3.MultipartUploadPart method), 4087
`MultipartUpload()` (S3.Object method), 4102
`MultipartUpload()` (S3.ObjectSummary method), 4119
`MultipartUpload()` (S3.ServiceResource method), 4038
`MultipartUploader` (class in boto3.s3.transfer), 4653
`MultipartUploadPart()` (S3.ServiceResource method), 4038

N

`name` (boto3.resources.model.Action attribute), 4639
`name` (boto3.resources.model.Identifier attribute), 4639
`name` (boto3.resources.model.Parameter attribute), 4639
`name` (boto3.resources.model.ResourceModel attribute), 4641
`name` (boto3.resources.model.Waiter attribute), 4642
`name` (CloudFormation.Stack attribute), 159
`name` (DynamoDB.Table attribute), 999
`name` (EC2.Image attribute), 1602
`name` (EC2.KeyPair attribute), 1661
`name` (EC2.PlacementGroup attribute), 1678
`name` (Glacier.Vault attribute), 2699
`name` (IAM.Group attribute), 2909
`name` (IAM.GroupPolicy attribute), 2917
`name` (IAM.InstanceProfile attribute), 2919
`name` (IAM.Role attribute), 2939
`name` (IAM.RolePolicy attribute), 2945
`name` (IAM.ServerCertificate attribute), 2950
`name` (IAM.User attribute), 2954
`name` (IAM.UserPolicy attribute), 2968
`name` (OpsWorks.Layer attribute), 3267
`name` (OpsWorks.Stack attribute), 3270
`name` (OpsWorks.StackSummary attribute), 3277
`name` (S3.Bucket attribute), 4041
`ne()` (boto3.dynamodb.conditions.Attr method), 4649
`network_acl_id` (EC2.NetworkAcl attribute), 1663
`network_acls` (EC2.ServiceResource attribute), 1576
`network_acls` (EC2.Vpc attribute), 1779
`network_interface_id` (EC2.NetworkInterface attribute), 1670
`network_interfaces` (EC2.Instance attribute), 1613
`network_interfaces` (EC2.ServiceResource attribute), 1578

network_interfaces (EC2.Subnet attribute), 1739
network_interfaces (EC2.Vpc attribute), 1780
NetworkAcl() (EC2.ServiceResource method), 1554
NetworkInterface() (EC2.ServiceResource method), 1554
not_exists() (boto3.dynamodb.conditions.Attr method), 4649
Notification() (Glacier.ServiceResource method), 2677
Notification() (Glacier.Vault method), 2705
Notification() (S3.Bucket method), 4046
notification_arns (CloudFormation.Stack attribute), 159
NullHandler (class in boto3), 4634
number_of_archives (Glacier.Vault attribute), 2699

O

Object() (S3.Bucket method), 4047
Object() (S3.MultipartUpload method), 4081
Object() (S3.ObjectAcl method), 4107
Object() (S3.ObjectSummary method), 4119
Object() (S3.ObjectVersion method), 4128
Object() (S3.ServiceResource method), 4038
object_download_file() (in module boto3.s3.inject), 4650
object_key (S3.MultipartUpload attribute), 4080
object_key (S3.MultipartUploadPart attribute), 4084
object_key (S3.ObjectAcl attribute), 4104
object_key (S3.ObjectVersion attribute), 4122
object_upload_file() (in module boto3.s3.inject), 4651
object_versions (S3.Bucket attribute), 4049
ObjectAcl() (S3.ServiceResource method), 4038
objects (S3.Bucket attribute), 4051
ObjectSummary() (S3.ServiceResource method), 4039
ObjectVersion() (S3.ServiceResource method), 4039
open() (boto3.s3.transfer.OSUtils method), 4653
open_file_chunk_reader() (boto3.s3.transfer.OSUtils method), 4653
operation (boto3.resources.model.Request attribute), 4640
OpsWorks.Client (built-in class), 3157
OpsWorks.Layer (built-in class), 3264
OpsWorks.ServiceResource (built-in class), 3258
OpsWorks.Stack (built-in class), 3268
OpsWorks.StackSummary (built-in class), 3276
OSUtils (class in boto3.s3.transfer), 4653
outputs (CloudFormation.Stack attribute), 159
owner (S3.BucketAcl attribute), 4055
owner (S3.MultipartUpload attribute), 4080
owner (S3.ObjectAcl attribute), 4105
owner (S3.ObjectSummary attribute), 4108
owner (S3.ObjectVersion attribute), 4122
owner_alias (EC2.Snapshot attribute), 1711
owner_id (EC2.Image attribute), 1603
owner_id (EC2.NetworkInterface attribute), 1670
owner_id (EC2.SecurityGroup attribute), 1698
owner_id (EC2.Snapshot attribute), 1711

P

packages (OpsWorks.Layer attribute), 3267
page_size() (boto3.resources.collection.CollectionManager method), 4636
page_size() (boto3.resources.collection.ResourceCollection method), 4638
page_size() (CloudFormation.ServiceResource method), 156
page_size() (CloudFormation.Stack method), 164, 165
page_size() (DynamoDB.ServiceResource method), 998
page_size() (EC2.Instance method), 1642
page_size() (EC2.PlacementGroup method), 1681
page_size() (EC2.RouteTable method), 1694
page_size() (EC2.ServiceResource method), 1557, 1560, 1566, 1574, 1576, 1578, 1581, 1583, 1585, 1587, 1589, 1591, 1594, 1596, 1598
page_size() (EC2.Subnet method), 1733, 1740
page_size() (EC2.Volume method), 1756
page_size() (EC2.Vpc method), 1769, 1772, 1779–1782, 1784–1786
page_size() (Glacier.Account method), 2681
page_size() (Glacier.ServiceResource method), 2678
page_size() (Glacier.Vault method), 2706–2710, 2712
page_size() (IAM.CurrentUser method), 2906–2908
page_size() (IAM.Group method), 2914–2916
page_size() (IAM.Policy method), 2932, 2933, 2935, 2936
page_size() (IAM.Role method), 2942, 2943, 2945
page_size() (IAM.ServiceResource method), 2885, 2886, 2888–2891, 2893, 2894
page_size() (IAM.User method), 2961, 2962, 2964–2967
page_size() (OpsWorks.ServiceResource method), 3264
page_size() (OpsWorks.Stack method), 3276
page_size() (S3.Bucket method), 4049, 4051, 4053
page_size() (S3.MultipartUpload method), 4083
page_size() (S3.ServiceResource method), 4040
page_size() (SNS.PlatformApplication method), 4201
page_size() (SNS.ServiceResource method), 4195–4197
page_size() (SNS.Topic method), 4214
page_size() (SQS.Queue method), 4259
page_size() (SQS.ServiceResource method), 4241
pages() (boto3.resources.collection.CollectionManager method), 4636
pages() (boto3.resources.collection.ResourceCollection method), 4638
paginate() (AutoScaling.Paginator.DescribeAutoScalingGroups method), 98
paginate() (AutoScaling.Paginator.DescribeAutoScalingInstances method), 102
paginate() (AutoScaling.Paginator.DescribeLaunchConfigurations method), 104
paginate() (AutoScaling.Paginator.DescribeNotificationConfigurations method), 108

[paginate\(\)](#) (AutoScaling.Paginator.DescribePolicies method), 109
[paginate\(\)](#) (AutoScaling.Paginator.DescribeScalingActivities method), 113
[paginate\(\)](#) (AutoScaling.Paginator.DescribeScheduledActions method), 115
[paginate\(\)](#) (AutoScaling.Paginator.DescribeTags method), 117
[paginate\(\)](#) (CloudFormation.Paginator.DescribeStackEvents method), 143
[paginate\(\)](#) (CloudFormation.Paginator.DescribeStacks method), 145
[paginate\(\)](#) (CloudFormation.Paginator.ListStackResources method), 148
[paginate\(\)](#) (CloudFormation.Paginator.ListStacks method), 150
[paginate\(\)](#) (CloudFront.Paginator.ListCloudFrontOriginAccessIdentifiers method), 297
[paginate\(\)](#) (CloudFront.Paginator.ListDistributions method), 298
[paginate\(\)](#) (CloudFront.Paginator.ListInvalidations method), 312
[paginate\(\)](#) (CloudFront.Paginator.ListStreamingDistributions method), 313
[paginate\(\)](#) (CloudWatch.Paginator.DescribeAlarmHistory method), 444
[paginate\(\)](#) (CloudWatch.Paginator.DescribeAlarms method), 446
[paginate\(\)](#) (CloudWatch.Paginator.ListMetrics method), 450
[paginate\(\)](#) (CloudWatchLogs.Paginator.DescribeDestinationGroups method), 3076
[paginate\(\)](#) (CloudWatchLogs.Paginator.DescribeLogGroups method), 3077
[paginate\(\)](#) (CloudWatchLogs.Paginator.DescribeLogStreams method), 3079
[paginate\(\)](#) (CloudWatchLogs.Paginator.DescribeMetricFilters method), 3081
[paginate\(\)](#) (CloudWatchLogs.Paginator.DescribeSubscriptionFilters method), 3082
[paginate\(\)](#) (CloudWatchLogs.Paginator.FilterLogEvents method), 3084
[paginate\(\)](#) (DataPipeline.Paginator.DescribeObjects method), 636
[paginate\(\)](#) (DataPipeline.Paginator.ListPipelines method), 638
[paginate\(\)](#) (DataPipeline.Paginator.QueryObjects method), 639
[paginate\(\)](#) (DynamoDB.Paginator.ListTables method), 940
[paginate\(\)](#) (DynamoDB.Paginator.Query method), 941
[paginate\(\)](#) (DynamoDB.Paginator.Scan method), 962
[paginate\(\)](#) (EC2.Paginator.DescribeInstances method), 1456
[paginate\(\)](#) (EC2.Paginator.DescribeInstanceStatus method), 1452
[paginate\(\)](#) (EC2.Paginator.DescribeReservedInstancesModifications method), 1470
[paginate\(\)](#) (EC2.Paginator.DescribeReservedInstancesOfferings method), 1473
[paginate\(\)](#) (EC2.Paginator.DescribeSnapshots method), 1477
[paginate\(\)](#) (EC2.Paginator.DescribeSpotPriceHistory method), 1480
[paginate\(\)](#) (EC2.Paginator.DescribeTags method), 1483
[paginate\(\)](#) (EC2.Paginator.DescribeVolumeStatus method), 1485
[paginate\(\)](#) (ECS.Paginator.ListClusters method), 1872
[paginate\(\)](#) (ECS.Paginator.ListContainerInstances method), 1873
[paginate\(\)](#) (ECS.Paginator.ListServices method), 1874
[paginate\(\)](#) (ECS.Paginator.ListTaskDefinitionFamilies method), 1875
[paginate\(\)](#) (ECS.Paginator.ListTaskDefinitions method), 1876
[paginate\(\)](#) (ECS.Paginator.ListTasks method), 1878
[paginate\(\)](#) (ElastiCache.Paginator.DescribeCacheClusters method), 2002
[paginate\(\)](#) (ElastiCache.Paginator.DescribeCacheEngineVersions method), 2009
[paginate\(\)](#) (ElastiCache.Paginator.DescribeCacheParameterGroups method), 2010
[paginate\(\)](#) (ElastiCache.Paginator.DescribeCacheParameters method), 2012
[paginate\(\)](#) (ElastiCache.Paginator.DescribeCacheSecurityGroups method), 2014
[paginate\(\)](#) (ElastiCache.Paginator.DescribeCacheSubnetGroups method), 2016
[paginate\(\)](#) (ElastiCache.Paginator.DescribeEngineDefaultParameters method), 2018
[paginate\(\)](#) (ElastiCache.Paginator.DescribeEvents method), 2021
[paginate\(\)](#) (ElastiCache.Paginator.DescribeReplicationGroups method), 2023
[paginate\(\)](#) (ElastiCache.Paginator.DescribeReservedCacheNodes method), 2026
[paginate\(\)](#) (ElastiCache.Paginator.DescribeReservedCacheNodesOfferings method), 2029
[paginate\(\)](#) (ElastiCache.Paginator.DescribeSnapshots method), 2033
[paginate\(\)](#) (ElasticBeanstalk.Paginator.DescribeEvents method), 2099
[paginate\(\)](#) (ElasticLoadBalancing.Paginator.DescribeLoadBalancers method), 2574
[paginate\(\)](#) (ElasticTranscoder.Paginator.ListJobsByPipeline

- method), 2428
- paginate() (ElasticTranscoder.Paginator.ListJobsByStatus method), 2469
- paginate() (ElasticTranscoder.Paginator.ListPipelines method), 2511
- paginate() (ElasticTranscoder.Paginator.ListPresets method), 2519
- paginate() (EMR.Paginator.ListBootstrapActions method), 2622
- paginate() (EMR.Paginator.ListClusters method), 2623
- paginate() (EMR.Paginator.ListInstanceGroups method), 2625
- paginate() (EMR.Paginator.ListInstances method), 2628
- paginate() (EMR.Paginator.ListSteps method), 2631
- paginate() (IAM.Paginator.GetGroup method), 2840
- paginate() (IAM.Paginator.ListAccessKeys method), 2842
- paginate() (IAM.Paginator.ListAccountAliases method), 2844
- paginate() (IAM.Paginator.ListGroupPolicies method), 2845
- paginate() (IAM.Paginator.ListGroups method), 2846
- paginate() (IAM.Paginator.ListGroupsForUser method), 2848
- paginate() (IAM.Paginator.ListInstanceProfiles method), 2850
- paginate() (IAM.Paginator.ListInstanceProfilesForRole method), 2852
- paginate() (IAM.Paginator.ListMFADevices method), 2855
- paginate() (IAM.Paginator.ListPolicies method), 2856
- paginate() (IAM.Paginator.ListRolePolicies method), 2859
- paginate() (IAM.Paginator.ListRoles method), 2860
- paginate() (IAM.Paginator.ListServerCertificates method), 2862
- paginate() (IAM.Paginator.ListSigningCertificates method), 2864
- paginate() (IAM.Paginator.ListUserPolicies method), 2866
- paginate() (IAM.Paginator.ListUsers method), 2867
- paginate() (IAM.Paginator.ListVirtualMFADevices method), 2869
- paginate() (ImportExport.Paginator.ListJobs method), 2978
- paginate() (Kinesis.Paginator.DescribeStream method), 2997
- paginate() (Kinesis.Paginator.ListStreams method), 3000
- paginate() (Lambda.Paginator.ListEventSourceMappings method), 3048
- paginate() (Lambda.Paginator.ListFunctions method), 3050
- paginate() (MachineLearning.Paginator.DescribeBatchPredictions method), 3135
- paginate() (MachineLearning.Paginator.DescribeDataSources method), 3138
- paginate() (MachineLearning.Paginator.DescribeEvaluations method), 3143
- paginate() (MachineLearning.Paginator.DescribeMLModels method), 3146
- paginate() (RDS.Paginator.DescribeDBEngineVersions method), 3536
- paginate() (RDS.Paginator.DescribeDBInstances method), 3538
- paginate() (RDS.Paginator.DescribeDBLogFiles method), 3548
- paginate() (RDS.Paginator.DescribeDBParameterGroups method), 3550
- paginate() (RDS.Paginator.DescribeDBParameters method), 3551
- paginate() (RDS.Paginator.DescribeDBSecurityGroups method), 3554
- paginate() (RDS.Paginator.DescribeDBSnapshots method), 3556
- paginate() (RDS.Paginator.DescribeDBSubnetGroups method), 3560
- paginate() (RDS.Paginator.DescribeEngineDefaultParameters method), 3562
- paginate() (RDS.Paginator.DescribeEvents method), 3567
- paginate() (RDS.Paginator.DescribeEventSubscriptions method), 3565
- paginate() (RDS.Paginator.DescribeOptionGroupOptions method), 3569
- paginate() (RDS.Paginator.DescribeOptionGroups method), 3572
- paginate() (RDS.Paginator.DescribeOrderableDBInstanceOptions method), 3576
- paginate() (RDS.Paginator.DescribeReservedDBInstances method), 3579
- paginate() (RDS.Paginator.DescribeReservedDBInstancesOfferings method), 3582
- paginate() (RDS.Paginator.DownloadDBLogFilePortion method), 3585
- paginate() (Redshift.Paginator.DescribeClusterParameterGroups method), 3767
- paginate() (Redshift.Paginator.DescribeClusterParameters method), 3769
- paginate() (Redshift.Paginator.DescribeClusters method), 3782
- paginate() (Redshift.Paginator.DescribeClusterSecurityGroups method), 3770
- paginate() (Redshift.Paginator.DescribeClusterSnapshots method), 3774

paginate() (Redshift.Paginator.DescribeClusterSubnetGroups method), 3778
 paginate() (Redshift.Paginator.DescribeClusterVersions method), 3781
 paginate() (Redshift.Paginator.DescribeDefaultClusterParameters method), 3790
 paginate() (Redshift.Paginator.DescribeEvents method), 3795
 paginate() (Redshift.Paginator.DescribeEventSubscriptions method), 3792
 paginate() (Redshift.Paginator.DescribeHsmClientCertificates method), 3797
 paginate() (Redshift.Paginator.DescribeHsmConfigurations method), 3799
 paginate() (Redshift.Paginator.DescribeOrderableClusterOptions method), 3801
 paginate() (Redshift.Paginator.DescribeReservedNodeOfferings method), 3803
 paginate() (Redshift.Paginator.DescribeReservedNodes method), 3805
 paginate() (Route53.Paginator.ListHealthChecks method), 3866
 paginate() (Route53.Paginator.ListHostedZones method), 3870
 paginate() (Route53.Paginator.ListResourceRecordSets method), 3872
 paginate() (Route53Domains.Paginator.ListDomains method), 3950
 paginate() (Route53Domains.Paginator.ListOperations method), 3952
 paginate() (S3.Paginator.ListMultipartUploads method), 4023
 paginate() (S3.Paginator.ListObjects method), 4027
 paginate() (S3.Paginator.ListObjectVersions method), 4025
 paginate() (S3.Paginator.ListParts method), 4030
 paginate() (SES.Paginator.ListIdentities method), 4164
 paginate() (SimpleDB.Paginator.ListDomains method), 4139
 paginate() (SimpleDB.Paginator.Select method), 4140
 paginate() (SNS.Paginator.ListEndpointsByPlatformApplications method), 4186
 paginate() (SNS.Paginator.ListPlatformApplications method), 4187
 paginate() (SNS.Paginator.ListSubscriptions method), 4188
 paginate() (SNS.Paginator.ListSubscriptionsByTopic method), 4190
 paginate() (SNS.Paginator.ListTopics method), 4191
 paginate() (StorageGateway.Paginator.DescribeTapeArchives method), 4321
 paginate() (StorageGateway.Paginator.DescribeClusterSubnetGroups method), 4323
 paginate() (StorageGateway.Paginator.DescribeTapes method), 4325
 paginate() (StorageGateway.Paginator.DescribeVTLDevices method), 4326
 paginate() (StorageGateway.Paginator.ListGateways method), 4328
 paginate() (StorageGateway.Paginator.ListVolumes method), 4330
 paginate() (Support.Paginator.DescribeCases method), 4368
 paginate() (Support.Paginator.DescribeCommunications method), 4373
 paginate() (SWF.Paginator.GetWorkflowExecutionHistory method), 4525
 paginate() (SWF.Paginator.ListActivityTypes method), 4566
 paginate() (SWF.Paginator.ListClosedWorkflowExecutions method), 4568
 paginate() (SWF.Paginator.ListDomains method), 4572
 paginate() (SWF.Paginator.ListOpenWorkflowExecutions method), 4574
 paginate() (SWF.Paginator.ListWorkflowTypes method), 4577
 paginate() (SWF.Paginator.PollForDecisionTask method), 4579
 Parameter (class in boto3.resources.model), 4639
 parameters (CloudFormation.Stack attribute), 159
 params (boto3.resources.model.DefinitionWithParams attribute), 4639
 params (boto3.resources.model.Request attribute), 4640
 params (boto3.resources.model.Waiter attribute), 4642
 Part() (S3.MultipartUpload method), 4081
 part_number (S3.MultipartUploadPart attribute), 4084
 part_size_in_bytes (Glacier.MultipartUpload attribute), 2690
 parts (S3.MultipartUpload attribute), 4082
 parts() (Glacier.MultipartUpload method), 2692
 password_data() (EC2.Instance method), 1631
 password_last_used (IAM.CurrentUser attribute), 2904
 password_last_used (IAM.User attribute), 2955
 password_reset_required (IAM.LoginProfile attribute), 2922
 password_reuse_prevention (IAM.AccountPasswordPolicy attribute), 2899
 path (boto3.resources.model.Action attribute), 4639
 path (boto3.resources.model.Parameter attribute), 4639
 path (boto3.resources.model.ResponseResource attribute), 4641
 path (IAM.CurrentUser attribute), 2904
 path (IAM.Group attribute), 2910
 path (IAM.InstanceProfile attribute), 2920

- path (IAM.Policy attribute), 2927
- path (IAM.Role attribute), 2939
- path (IAM.User attribute), 2955
- payer (S3.BucketRequestPayment attribute), 4070
- physical_resource_id (CloudFormation.Event attribute), 157
- physical_resource_id (CloudFormation.StackResource attribute), 166
- physical_resource_id (CloudFormation.StackResourceSummary attribute), 168
- placement (EC2.Instance attribute), 1614
- placement_group (EC2.Instance attribute), 1615
- placement_groups (EC2.ServiceResource attribute), 1581
- PlacementGroup() (EC2.ServiceResource method), 1554
- platform (EC2.Image attribute), 1603
- platform (EC2.Instance attribute), 1614
- platform_applications (SNS.ServiceResource attribute), 4194
- PlatformApplication() (SNS.ServiceResource method), 4194
- PlatformEndpoint() (SNS.ServiceResource method), 4194
- policies (IAM.Group attribute), 2914
- policies (IAM.Role attribute), 2944
- policies (IAM.ServiceResource attribute), 2886
- policies (IAM.User attribute), 2965
- policy (S3.BucketPolicy attribute), 4069
- Policy() (IAM.Group method), 2913
- Policy() (IAM.Role method), 2941
- Policy() (IAM.ServiceResource method), 2882
- Policy() (IAM.User method), 2960
- Policy() (S3.Bucket method), 4047
- policy_document (IAM.GroupPolicy attribute), 2917
- policy_document (IAM.RolePolicy attribute), 2946
- policy_document (IAM.UserPolicy attribute), 2968
- policy_id (IAM.Policy attribute), 2927
- policy_name (IAM.GroupPolicy attribute), 2917
- policy_name (IAM.Policy attribute), 2928
- policy_name (IAM.RolePolicy attribute), 2946
- policy_name (IAM.UserPolicy attribute), 2968
- PolicyVersion() (IAM.ServiceResource method), 2882
- poll_for_activity_task() (SWF.Client method), 4452
- poll_for_decision_task() (SWF.Client method), 4454
- poll_for_jobs() (CodePipeline.Client method), 542
- poll_for_task() (DataPipeline.Client method), 621
- poll_for_third_party_jobs() (CodePipeline.Client method), 547
- predict() (MachineLearning.Client method), 3131
- PREFIX (boto3.resources.model.Waiter attribute), 4642
- private_dns_name (EC2.Instance attribute), 1614
- private_dns_name (EC2.NetworkInterface attribute), 1670
- private_ip_address (EC2.Instance attribute), 1614
- private_ip_address (EC2.NetworkInterface attribute), 1670
- private_ip_addresses (EC2.NetworkInterface attribute), 1670
- product_codes (EC2.Image attribute), 1603
- product_codes (EC2.Instance attribute), 1614
- profile_name (boto3.session.Session attribute), 4647
- progress (EC2.Snapshot attribute), 1711
- promote_read_replica() (RDS.Client method), 3475
- propagating_vgws (EC2.RouteTable attribute), 1689
- provisioned_throughput (DynamoDB.Table attribute), 1001
- public (EC2.Image attribute), 1603
- public_dns_name (EC2.Instance attribute), 1614
- public_ip_address (EC2.Instance attribute), 1614
- publish() (SNS.Client method), 4180
- publish() (SNS.PlatformEndpoint method), 4202
- publish() (SNS.Topic method), 4209
- purchase_reserved_cache_nodes_offering() (ElastiCache.Client method), 1990
- purchase_reserved_db_instances_offering() (RDS.Client method), 3484
- purchase_reserved_instances_offering() (EC2.Client method), 1393
- purchase_reserved_node_offering() (Redshift.Client method), 3733
- purge() (SQS.Queue method), 4250
- purge_queue() (SQS.Client method), 4227
- put() (boto3.s3.transfer.ShutdownQueue method), 4654
- put() (IAM.GroupPolicy method), 2918
- put() (IAM.RolePolicy method), 2946
- put() (IAM.UserPolicy method), 2969
- put() (S3.BucketAcl method), 4055
- put() (S3.BucketCors method), 4057
- put() (S3.BucketLifecycle method), 4060
- put() (S3.BucketLogging method), 4062
- put() (S3.BucketNotification method), 4065
- put() (S3.BucketPolicy method), 4069
- put() (S3.BucketRequestPayment method), 4070
- put() (S3.BucketTagging method), 4072
- put() (S3.BucketVersioning method), 4074
- put() (S3.BucketWebsite method), 4077
- put() (S3.Object method), 4099
- put() (S3.ObjectAcl method), 4105
- put() (S3.ObjectSummary method), 4116
- put_action_revision() (CodePipeline.Client method), 548
- put_attributes() (SimpleDB.Client method), 4136
- put_bucket_acl() (S3.Client method), 3999
- put_bucket_cors() (S3.Client method), 4000
- put_bucket_lifecycle() (S3.Client method), 4001
- put_bucket_lifecycle_configuration() (S3.Client method), 4003
- put_bucket_logging() (S3.Client method), 4005
- put_bucket_notification() (S3.Client method), 4006

`put_bucket_notification_configuration()` (S3.Client method), 4007
`put_bucket_policy()` (S3.Client method), 4010
`put_bucket_replication()` (S3.Client method), 4010
`put_bucket_request_payment()` (S3.Client method), 4011
`put_bucket_tagging()` (S3.Client method), 4011
`put_bucket_versioning()` (S3.Client method), 4012
`put_bucket_website()` (S3.Client method), 4012
`put_configuration_recorder()` (ConfigService.Client method), 607
`put_delivery_channel()` (ConfigService.Client method), 608
`put_destination()` (CloudWatchLogs.Client method), 3070
`put_destination_policy()` (CloudWatchLogs.Client method), 3071
`put_group_policy()` (IAM.Client method), 2817
`put_identity_policy()` (SES.Client method), 4152
`put_item()` (DynamoDB.Client method), 841
`put_item()` (DynamoDB.Table method), 1027
`put_job_failure_result()` (CodePipeline.Client method), 549
`put_job_success_result()` (CodePipeline.Client method), 550
`put_key_policy()` (KMS.Client method), 3020
`put_lifecycle_hook()` (AutoScaling.Client method), 86
`put_log_events()` (CloudWatchLogs.Client method), 3071
`put_metric_alarm()` (CloudWatch.Client method), 440
`put_metric_data()` (CloudWatch.Client method), 442
`put_metric_filter()` (CloudWatchLogs.Client method), 3073
`put_notification_configuration()` (AutoScaling.Client method), 88
`put_object()` (S3.Bucket method), 4044
`put_object()` (S3.Client method), 4014
`put_object_acl()` (S3.Client method), 4016
`put_pipeline_definition()` (DataPipeline.Client method), 624
`put_record()` (Kinesis.Client method), 2991
`put_records()` (Kinesis.Client method), 2993
`put_retention_policy()` (CloudWatchLogs.Client method), 3073
`put_role_policy()` (IAM.Client method), 2818
`put_scaling_policy()` (AutoScaling.Client method), 89
`put_scheduled_update_group_action()` (AutoScaling.Client method), 91
`put_subscription_filter()` (CloudWatchLogs.Client method), 3074
`put_third_party_job_failure_result()` (CodePipeline.Client method), 550
`put_third_party_job_success_result()` (CodePipeline.Client method), 551
`put_user_policy()` (IAM.Client method), 2818

Q

`qr_code_png` (IAM.VirtualMfaDevice attribute), 2970
`query()` (DynamoDB.Client method), 861
`query()` (DynamoDB.Table method), 1039
`query_objects()` (DataPipeline.Client method), 627
`Queue()` (SQS.Message method), 4244
`Queue()` (SQS.ServiceResource method), 4240
`queue_configurations` (S3.BucketNotification attribute), 4064
`queue_url` (SQS.Message attribute), 4242
`queues` (SQS.ServiceResource attribute), 4240
`QueueShutdownError`, 4653

R

`ramdisk_id` (EC2.Image attribute), 1603
`ramdisk_id` (EC2.Instance attribute), 1614
`random_file_extension()` (in module `boto3.s3.transfer`), 4654
`RawHandler` (class in `boto3.resources.response`), 4643
`RDS.Client` (built-in class), 3278
`RDS.Paginator.DescribeDBEngineVersions` (built-in class), 3536
`RDS.Paginator.DescribeDBInstances` (built-in class), 3538
`RDS.Paginator.DescribeDBLogFiles` (built-in class), 3548
`RDS.Paginator.DescribeDBParameterGroups` (built-in class), 3549
`RDS.Paginator.DescribeDBParameters` (built-in class), 3551
`RDS.Paginator.DescribeDBSecurityGroups` (built-in class), 3554
`RDS.Paginator.DescribeDBSnapshots` (built-in class), 3556
`RDS.Paginator.DescribeDBSubnetGroups` (built-in class), 3560
`RDS.Paginator.DescribeEngineDefaultParameters` (built-in class), 3562
`RDS.Paginator.DescribeEvents` (built-in class), 3567
`RDS.Paginator.DescribeEventSubscriptions` (built-in class), 3564
`RDS.Paginator.DescribeOptionGroupOptions` (built-in class), 3569
`RDS.Paginator.DescribeOptionGroups` (built-in class), 3572
`RDS.Paginator.DescribeOrderableDBInstanceOptions` (built-in class), 3576
`RDS.Paginator.DescribeReservedDBInstances` (built-in class), 3579
`RDS.Paginator.DescribeReservedDBInstancesOfferings` (built-in class), 3582
`RDS.Paginator.DownloadDBLogFilePortion` (built-in class), 3585
`RDS.Waiter.DBInstanceAvailable` (built-in class), 3586

- RDS.Waiter.DBInstanceDeleted (built-in class), 3587
- RDS.Waiter.DBSnapshotCompleted (built-in class), 3588
- re_encrypt() (KMS.Client method), 3021
- read() (boto3.s3.transfer.ReadFileChunk method), 4654
- read() (boto3.s3.transfer.StreamReaderProgress method), 4654
- read_job() (ElasticTranscoder.Client method), 2333
- read_pipeline() (ElasticTranscoder.Client method), 2374
- read_preset() (ElasticTranscoder.Client method), 2382
- ReadFileChunk (class in boto3.s3.transfer), 4653
- reboot() (EC2.Instance method), 1631
- reboot() (EC2.PlacementGroup method), 1682
- reboot() (EC2.ServiceResource method), 1567
- reboot() (EC2.Subnet method), 1733
- reboot() (EC2.Vpc method), 1772
- reboot_cache_cluster() (ElastiCache.Client method), 1992
- reboot_cluster() (Redshift.Client method), 3735
- reboot_db_instance() (RDS.Client method), 3486
- reboot_instance() (OpsWorks.Client method), 3235
- reboot_instances() (EC2.Client method), 1394
- reboot_workspaces() (WorkSpaces.Client method), 4630
- rebuild_environment() (ElasticBeanstalk.Client method), 2080
- rebuild_workspaces() (WorkSpaces.Client method), 4631
- receipt_handle (SQS.Message attribute), 4242
- receive_message() (SQS.Client method), 4227
- receive_messages() (SQS.Queue method), 4250
- record_activity_task_heartbeat() (SWF.Client method), 4496
- record_lifecycle_action_heartbeat() (AutoScaling.Client method), 92
- redirect_all_requests_to (S3.BucketWebsite attribute), 4076
- Redshift.Client (built-in class), 3590
- Redshift.Paginator.DescribeClusterParameterGroups (built-in class), 3767
- Redshift.Paginator.DescribeClusterParameters (built-in class), 3768
- Redshift.Paginator.DescribeClusters (built-in class), 3782
- Redshift.Paginator.DescribeClusterSecurityGroups (built-in class), 3770
- Redshift.Paginator.DescribeClusterSnapshots (built-in class), 3774
- Redshift.Paginator.DescribeClusterSubnetGroups (built-in class), 3778
- Redshift.Paginator.DescribeClusterVersions (built-in class), 3781
- Redshift.Paginator.DescribeDefaultClusterParameters (built-in class), 3790
- Redshift.Paginator.DescribeEvents (built-in class), 3795
- Redshift.Paginator.DescribeEventSubscriptions (built-in class), 3792
- Redshift.Paginator.DescribeHsmClientCertificates (built-in class), 3797
- Redshift.Paginator.DescribeHsmConfigurations (built-in class), 3799
- Redshift.Paginator.DescribeOrderableClusterOptions (built-in class), 3801
- Redshift.Paginator.DescribeReservedNodeOfferings (built-in class), 3803
- Redshift.Paginator.DescribeReservedNodes (built-in class), 3805
- Redshift.Waiter.ClusterAvailable (built-in class), 3807
- Redshift.Waiter.ClusterDeleted (built-in class), 3808
- Redshift.Waiter.SnapshotAvailable (built-in class), 3809
- references (boto3.resources.model.ResourceModel attribute), 4641
- refresh_trusted_advisor_check() (Support.Client method), 4367
- region (OpsWorks.Stack attribute), 3270
- register_activity_type() (SWF.Client method), 4498
- register_application_revision() (CodeDeploy.Client method), 501
- register_container_instance() (ECS.Client method), 1836
- register_device() (CognitoSync.Client method), 588
- register_domain() (Route53Domains.Client method), 3903
- register_domain() (SWF.Client method), 4500
- register_ecs_cluster() (OpsWorks.Client method), 3235
- register_elastic_ip() (OpsWorks.Client method), 3236
- register_image() (EC2.Client method), 1394
- register_image() (EC2.ServiceResource method), 1550
- register_instance() (OpsWorks.Client method), 3236
- register_instances_with_load_balancer() (ElasticLoadBalancing.Client method), 2570
- register_on_premises_instance() (CodeDeploy.Client method), 503
- register_rds_db_instance() (OpsWorks.Client method), 3237
- register_task_definition() (ECS.Client method), 1840
- register_volume() (OpsWorks.Client method), 3238
- register_workflow_type() (SWF.Client method), 4500
- reject() (EC2.VpcPeeringConnection method), 1790
- reject_vpc_peering_connection() (EC2.Client method), 1397
- release_address() (EC2.Client method), 1398
- reload() (CloudFormation.Stack method), 160
- reload() (DynamoDB.Table method), 1055
- reload() (EC2.DhcpOptions method), 1601
- reload() (EC2.Image method), 1610
- reload() (EC2.Instance method), 1632
- reload() (EC2.InternetGateway method), 1660
- reload() (EC2.KeyPair method), 1662
- reload() (EC2.NetworkAcl method), 1666
- reload() (EC2.NetworkInterface method), 1676
- reload() (EC2.PlacementGroup method), 1679

- reload() (EC2.RouteTable method), 1692
- reload() (EC2.SecurityGroup method), 1704
- reload() (EC2.Snapshot method), 1717
- reload() (EC2.Subnet method), 1730
- reload() (EC2.Tag method), 1742
- reload() (EC2.Volume method), 1754
- reload() (EC2.Vpc method), 1767
- reload() (EC2.VpcPeeringConnection method), 1791
- reload() (Glacier.Job method), 2689
- reload() (Glacier.Notification method), 2697
- reload() (Glacier.Vault method), 2703
- reload() (IAM.AccountPasswordPolicy method), 2900
- reload() (IAM.Group method), 2912
- reload() (IAM.GroupPolicy method), 2918
- reload() (IAM.InstanceProfile method), 2921
- reload() (IAM.LoginProfile method), 2923
- reload() (IAM.Policy method), 2931
- reload() (IAM.PolicyVersion method), 2938
- reload() (IAM.Role method), 2941
- reload() (IAM.RolePolicy method), 2947
- reload() (IAM.SamlProvider method), 2948
- reload() (IAM.ServerCertificate method), 2951
- reload() (IAM.User method), 2958
- reload() (IAM.UserPolicy method), 2969
- reload() (OpsWorks.Layer method), 3268
- reload() (OpsWorks.Stack method), 3275
- reload() (S3.BucketAcl method), 4056
- reload() (S3.BucketCors method), 4058
- reload() (S3.BucketLifecycle method), 4061
- reload() (S3.BucketLogging method), 4063
- reload() (S3.BucketNotification method), 4068
- reload() (S3.BucketPolicy method), 4069
- reload() (S3.BucketRequestPayment method), 4071
- reload() (S3.BucketTagging method), 4073
- reload() (S3.BucketVersioning method), 4075
- reload() (S3.BucketWebsite method), 4078
- reload() (S3.Object method), 4101
- reload() (S3.ObjectAcl method), 4106
- reload() (SNS.PlatformApplication method), 4199
- reload() (SNS.PlatformEndpoint method), 4204
- reload() (SNS.Subscription method), 4206
- reload() (SNS.Topic method), 4212
- reload() (SQS.Queue method), 4252
- remove_client_id_from_open_id_connect_provider() (IAM.Client method), 2819
- remove_file() (boto3.s3.transfer.OSUtils method), 4653
- remove_group() (IAM.User method), 2959
- remove_permission() (Lambda.Client method), 3043
- remove_permission() (SNS.Client method), 4182
- remove_permission() (SNS.Topic method), 4212
- remove_permission() (SQS.Client method), 4231
- remove_permission() (SQS.Queue method), 4252
- remove_role() (IAM.InstanceProfile method), 2921
- remove_role_from_instance_profile() (IAM.Client method), 2819
- remove_source_identifier_from_subscription() (RDS.Client method), 3495
- remove_tags() (DataPipeline.Client method), 629
- remove_tags() (ElasticLoadBalancing.Client method), 2571
- remove_tags() (EMR.Client method), 2610
- remove_tags_from_on_premises_instances() (CodeDeploy.Client method), 503
- remove_tags_from_resource() (ElastiCache.Client method), 1998
- remove_tags_from_resource() (RDS.Client method), 3496
- remove_tags_from_resource() (StorageGateway.Client method), 4311
- remove_tags_from_stream() (Kinesis.Client method), 2995
- remove_tags_from_vault() (Glacier.Client method), 2669
- remove_user() (IAM.Group method), 2912
- remove_user_from_group() (IAM.Client method), 2820
- rename_file() (boto3.s3.transfer.OSUtils method), 4653
- replace_association() (EC2.NetworkAcl method), 1667
- replace_entry() (EC2.NetworkAcl method), 1667
- replace_network_acl_association() (EC2.Client method), 1399
- replace_network_acl_entry() (EC2.Client method), 1399
- replace_route() (EC2.Client method), 1400
- replace_route_table_association() (EC2.Client method), 1401
- replace_subnet() (EC2.RouteTableAssociation method), 1696
- replication_status (S3.Object attribute), 4090
- report_instance_status() (EC2.Client method), 1402
- report_status() (EC2.Instance method), 1632
- report_task_progress() (DataPipeline.Client method), 629
- report_task_runner_heartbeat() (DataPipeline.Client method), 630
- request (boto3.resources.model.Action attribute), 4639
- Request (class in boto3.resources.model), 4640
- request_cancel_workflow_execution() (SWF.Client method), 4502
- request_charged (S3.Object attribute), 4090
- request_charged (S3.ObjectAcl attribute), 4105
- request_environment_info() (ElasticBeanstalk.Client method), 2080
- request_spot_fleet() (EC2.Client method), 1403
- request_spot_instances() (EC2.Client method), 1410
- request_vpc_peering_connection() (EC2.Vpc method), 1767
- requested_vpc_peering_connections (EC2.Vpc attribute), 1782
- requester_id (EC2.NetworkInterface attribute), 1670
- requester_managed (EC2.NetworkInterface attribute),

- 1670
- requester_vpc (EC2.VpcPeeringConnection attribute), 1787
- requester_vpc_info (EC2.VpcPeeringConnection attribute), 1787
- RequestPayment() (S3.Bucket method), 4047
- require_lowercase_characters (IAM.AccountPasswordPolicy attribute), 2899
- require_numbers (IAM.AccountPasswordPolicy attribute), 2899
- require_symbols (IAM.AccountPasswordPolicy attribute), 2899
- require_uppercase_characters (IAM.AccountPasswordPolicy attribute), 2899
- reset_attribute() (EC2.Image method), 1610
- reset_attribute() (EC2.Instance method), 1633
- reset_attribute() (EC2.NetworkInterface method), 1677
- reset_attribute() (EC2.Snapshot method), 1718
- reset_cache() (StorageGateway.Client method), 4312
- reset_cache_parameter_group() (ElastiCache.Client method), 1999
- reset_cluster_parameter_group() (Redshift.Client method), 3742
- reset_db_cluster_parameter_group() (RDS.Client method), 3497
- reset_db_parameter_group() (RDS.Client method), 3499
- reset_image_attribute() (EC2.Client method), 1423
- reset_instance_attribute() (EC2.Client method), 1423
- reset_kernel() (EC2.Instance method), 1633
- reset_network_interface_attribute() (EC2.Client method), 1424
- reset_ramdisk() (EC2.Instance method), 1634
- reset_snapshot_attribute() (EC2.Client method), 1424
- reset_source_dest_check() (EC2.Instance method), 1634
- resolve_case() (Support.Client method), 4368
- resource (boto3.resources.model.Action attribute), 4639
- resource() (boto3.session.Session method), 4647
- Resource() (CloudFormation.Stack method), 163
- Resource() (CloudFormation.StackResourceSummary method), 168
- resource() (in module boto3), 4634
- resource_id (EC2.Tag attribute), 1741
- resource_properties (CloudFormation.Event attribute), 157
- resource_status (CloudFormation.Event attribute), 157
- resource_status (CloudFormation.StackResource attribute), 166
- resource_status (CloudFormation.StackResourceSummary attribute), 168
- resource_status_reason (CloudFormation.Event attribute), 157
- resource_status_reason (CloudFormation.StackResource attribute), 166
- resource_status_reason (CloudFormation.StackResourceSummary attribute), 168
- resource_summaries (CloudFormation.Stack attribute), 164
- resource_type (CloudFormation.Event attribute), 157
- resource_type (CloudFormation.StackResource attribute), 167
- resource_type (CloudFormation.StackResourceSummary attribute), 168
- resource_type (EC2.Tag attribute), 1741
- ResourceCollection (class in boto3.resources.collection), 4637
- ResourceFactory (class in boto3.resources.factory), 4645
- ResourceHandler (class in boto3.resources.response), 4643
- ResourceMeta (class in boto3.resources.base), 4644
- ResourceModel (class in boto3.resources.model), 4640
- respond_activity_task_canceled() (SWF.Client method), 4503
- respond_activity_task_completed() (SWF.Client method), 4504
- respond_activity_task_failed() (SWF.Client method), 4505
- respond_decision_task_completed() (SWF.Client method), 4506
- ResponseResource (class in boto3.resources.model), 4641
- restart_app_server() (ElasticBeanstalk.Client method), 2081
- restore (S3.Object attribute), 4090
- restore_address_to_classic() (EC2.Client method), 1425
- restore_db_cluster_from_snapshot() (RDS.Client method), 3501
- restore_db_cluster_to_point_in_time() (RDS.Client method), 3505
- restore_db_instance_from_db_snapshot() (RDS.Client method), 3510
- restore_db_instance_to_point_in_time() (RDS.Client method), 3521
- restore_from_cluster_snapshot() (Redshift.Client method), 3744
- restore_from_snapshot() (DirectoryService.Client method), 753
- restore_object() (S3.Client method), 4018
- resume_processes() (AutoScaling.Client method), 93
- resync() (IAM.MfaDevice method), 2925
- resync_mfa_device() (IAM.Client method), 2820
- retire_grant() (KMS.Client method), 3022
- retrieval_byte_range (Glacier.Job attribute), 2686
- retrieve_domain_auth_code() (Route53Domains.Client method), 3917
- retrieve_environment_info() (ElasticBeanstalk.Client method), 2081

retrieve_tape_archive() (StorageGateway.Client method), 4313

retrieve_tape_recovery_point() (StorageGateway.Client method), 4313

revoke_cache_security_group_ingress() (Elasticache.Client method), 2000

revoke_cluster_security_group_ingress() (Redshift.Client method), 3753

revoke_db_security_group_ingress() (RDS.Client method), 3533

revoke_egress() (EC2.SecurityGroup method), 1704

revoke_grant() (KMS.Client method), 3022

revoke_ingress() (EC2.SecurityGroup method), 1707

revoke_security_group_egress() (EC2.Client method), 1425

revoke_security_group_ingress() (EC2.Client method), 1428

revoke_snapshot_access() (Redshift.Client method), 3756

Role() (IAM.AssumeRolePolicy method), 2903

Role() (IAM.RolePolicy method), 2947

Role() (IAM.ServiceResource method), 2882

role_id (IAM.Role attribute), 2939

role_name (IAM.AssumeRolePolicy attribute), 2903

role_name (IAM.Role attribute), 2939

role_name (IAM.RolePolicy attribute), 2945

RolePolicy() (IAM.ServiceResource method), 2883

roles (IAM.InstanceProfile attribute), 2920

roles (IAM.ServiceResource attribute), 2888

roles_attribute (IAM.InstanceProfile attribute), 2920

root_device_name (EC2.Image attribute), 1603

root_device_name (EC2.Instance attribute), 1614

root_device_type (EC2.Image attribute), 1603

root_device_type (EC2.Instance attribute), 1614

rotate_encryption_key() (Redshift.Client method), 3759

Route53.Client (built-in class), 3811

Route53.Paginator.ListHealthChecks (built-in class), 3866

Route53.Paginator.ListHostedZones (built-in class), 3870

Route53.Paginator.ListResourceRecordSets (built-in class), 3872

Route53.Waiter.ResourceRecordSetsChanged (built-in class), 3876

Route53Domains.Client (built-in class), 3877

Route53Domains.Paginator.ListDomains (built-in class), 3950

Route53Domains.Paginator.ListOperations (built-in class), 3951

route_table (EC2.RouteTableAssociation attribute), 1696

route_table_association_id (EC2.RouteTableAssociation attribute), 1695

route_table_id (EC2.RouteTable attribute), 1689

route_table_id (EC2.RouteTableAssociation attribute), 1695

route_tables (EC2.ServiceResource attribute), 1583

route_tables (EC2.Vpc attribute), 1783

routes (EC2.RouteTable attribute), 1689

RouteTable() (EC2.ServiceResource method), 1554

RouteTableAssociation() (EC2.ServiceResource method), 1554

routing_rules (S3.BucketWebsite attribute), 4076

rules (S3.BucketLifecycle attribute), 4059

run_instances() (EC2.Client method), 1430

run_job_flow() (EMR.Client method), 2611

run_task() (ECS.Client method), 1851

S

S3.Bucket (built-in class), 4040

S3.BucketAcl (built-in class), 4054

S3.BucketCors (built-in class), 4056

S3.BucketLifecycle (built-in class), 4059

S3.BucketLogging (built-in class), 4062

S3.BucketNotification (built-in class), 4064

S3.BucketPolicy (built-in class), 4068

S3.BucketRequestPayment (built-in class), 4070

S3.BucketTagging (built-in class), 4071

S3.BucketVersioning (built-in class), 4073

S3.BucketWebsite (built-in class), 4075

S3.Client (built-in class), 3954

S3.MultipartUpload (built-in class), 4079

S3.MultipartUploadPart (built-in class), 4083

S3.Object (built-in class), 4087

S3.ObjectAcl (built-in class), 4104

S3.ObjectSummary (built-in class), 4107

S3.ObjectVersion (built-in class), 4121

S3.Paginator.ListMultipartUploads (built-in class), 4023

S3.Paginator.ListObjects (built-in class), 4027

S3.Paginator.ListObjectVersions (built-in class), 4025

S3.Paginator.ListParts (built-in class), 4029

S3.ServiceResource (built-in class), 4035

S3.Waiter.BucketExists (built-in class), 4032

S3.Waiter.BucketNotExists (built-in class), 4032

S3.Waiter.ObjectExists (built-in class), 4032

S3.Waiter.ObjectNotExists (built-in class), 4033

S3Transfer (class in boto3.s3.transfer), 4654

saml_metadata_document (IAM.SamlProvider attribute), 2948

saml_providers (IAM.ServiceResource attribute), 2889

SamlProvider() (IAM.ServiceResource method), 2883

scan() (DynamoDB.Client method), 885

scan() (DynamoDB.Table method), 1055

schedule_run() (DeviceFarm.Client method), 694

search() (CloudSearchDomain.Client method), 405

secret (IAM.AccessKeyPair attribute), 2897

secret_access_key (IAM.AccessKeyPair attribute), 2897

security_groups (EC2.Instance attribute), 1614

security_groups (EC2.ServiceResource attribute), 1585

security_groups (EC2.Vpc attribute), 1784

SecurityGroup() (EC2.ServiceResource method), 1554

- seek() (boto3.s3.transfer.ReadFileChunk method), 4654
- select() (SimpleDB.Client method), 4137
- send_email() (SES.Client method), 4153
- send_message() (SQS.Client method), 4231
- send_message() (SQS.Queue method), 4252
- send_message_batch() (SQS.Client method), 4233
- send_messages() (SQS.Queue method), 4254
- send_raw_email() (SES.Client method), 4156
- serial_number (IAM.MfaDevice attribute), 2924
- serial_number (IAM.VirtualMfaDevice attribute), 2970
- server_certificate_metadata (IAM.ServerCertificate attribute), 2950
- server_certificates (IAM.ServiceResource attribute), 2890
- server_side_encryption (S3.Object attribute), 4090
- ServerCertificate() (IAM.ServiceResource method), 2883
- service_name (boto3.resources.base.ResourceMeta attribute), 4645
- service_role_arn (OpsWorks.Stack attribute), 3271
- ServiceAction (class in boto3.resources.action), 4644
- ServiceResource (class in boto3.resources.base), 4645
- SES.Client (built-in class), 4141
- SES.Paginator.ListIdentities (built-in class), 4164
- SES.Waiter.IdentityExists (built-in class), 4165
- Session (class in boto3.session), 4645
- set() (Glacier.Notification method), 2697
- set_alarm_state() (CloudWatch.Client method), 444
- set_as_default() (IAM.PolicyVersion method), 2938
- set_attributes() (SNS.PlatformApplication method), 4199
- set_attributes() (SNS.PlatformEndpoint method), 4205
- set_attributes() (SNS.Subscription method), 4207
- set_attributes() (SNS.Topic method), 4212
- set_attributes() (SQS.Queue method), 4257
- set_cognito_events() (CognitoSync.Client method), 589
- set_data_retrieval_policy() (Glacier.Client method), 2670
- set_default_policy_version() (IAM.Client method), 2821
- set_desired_capacity() (AutoScaling.Client method), 93
- set_endpoint_attributes() (SNS.Client method), 4182
- set_identity_dkim_enabled() (SES.Client method), 4159
- set_identity_feedback_forwarding_enabled() (SES.Client method), 4160
- set_identity_notification_topic() (SES.Client method), 4161
- set_identity_pool_configuration() (CognitoSync.Client method), 589
- set_identity_pool_roles() (CognitoIdentity.Client method), 573
- set_instance_health() (AutoScaling.Client method), 94
- set_load_balancer_listener_ssl_certificate() (ElasticLoadBalancing.Client method), 2572
- set_load_balancer_policies_for_backend_server() (ElasticLoadBalancing.Client method), 2573
- set_load_balancer_policies_of_listener() (ElasticLoadBalancing.Client method), 2573
- set_load_based_auto_scaling() (OpsWorks.Client method), 3239
- set_permission() (OpsWorks.Client method), 3241
- set_platform_application_attributes() (SNS.Client method), 4183
- set_queue_attributes() (SQS.Client method), 4236
- set_stack_policy() (CloudFormation.Client method), 138
- set_status() (DataPipeline.Client method), 631
- set_stream_logger() (in module boto3), 4634
- set_subscription_attributes() (SNS.Client method), 4184
- set_task_status() (DataPipeline.Client method), 632
- set_termination_protection() (EMR.Client method), 2620
- set_time_based_auto_scaling() (OpsWorks.Client method), 3242
- set_topic_attributes() (SNS.Client method), 4184
- set_vault_access_policy() (Glacier.Client method), 2671
- set_vault_notifications() (Glacier.Client method), 2671
- set_visible_to_all_users() (EMR.Client method), 2620
- setup_default_session() (in module boto3), 4634
- sha256_tree_hash (Glacier.Job attribute), 2686
- shape (boto3.resources.model.ResourceModel attribute), 4641
- shortname (OpsWorks.Layer attribute), 3267
- shutdown_gateway() (StorageGateway.Client method), 4314
- ShutdownQueue (class in boto3.s3.transfer), 4654
- signal_resource() (CloudFormation.Client method), 139
- signal_workflow_execution() (SWF.Client method), 4520
- signing_certificates (IAM.CurrentUser attribute), 2907
- signing_certificates (IAM.User attribute), 2966
- SigningCertificate() (IAM.ServiceResource method), 2883
- SigningCertificate() (IAM.User method), 2960
- SimpleDB.Client (built-in class), 4129
- SimpleDB.Paginator.ListDomains (built-in class), 4139
- SimpleDB.Paginator.Select (built-in class), 4139
- simulate_custom_policy() (IAM.Client method), 2821
- simulate_principal_policy() (IAM.Client method), 2825
- size (EC2.Volume attribute), 1744
- size (S3.MultipartUploadPart attribute), 4084
- size (S3.ObjectSummary attribute), 4108
- size (S3.ObjectVersion attribute), 4122
- size() (boto3.dynamodb.conditions.Attr method), 4649
- size_in_bytes (Glacier.Vault attribute), 2699
- Snapshot() (EC2.ServiceResource method), 1555
- snapshot_id (EC2.Snapshot attribute), 1711
- snapshot_id (EC2.Volume attribute), 1744
- snapshots (EC2.ServiceResource attribute), 1587
- snapshots (EC2.Volume attribute), 1754
- SNS.Client (built-in class), 4166
- SNS.Paginator.ListEndpointsByPlatformApplication (built-in class), 4186
- SNS.Paginator.ListPlatformApplications (built-in class), 4187

`SNS.Paginator.ListSubscriptions` (built-in class), 4188
`SNS.Paginator.ListSubscriptionsByTopic` (built-in class), 4190
`SNS.Paginator.ListTopics` (built-in class), 4191
`SNS.PlatformApplication` (built-in class), 4197
`SNS.PlatformEndpoint` (built-in class), 4201
`SNS.ServiceResource` (built-in class), 4192
`SNS.Subscription` (built-in class), 4205
`SNS.Topic` (built-in class), 4207
`sns_topic` (Glacier.Job attribute), 2686
`sns_topic` (Glacier.Notification attribute), 2696
`source` (boto3.resources.model.Parameter attribute), 4639
`source_dest_check` (EC2.Instance attribute), 1614
`source_dest_check` (EC2.NetworkInterface attribute), 1670
`split_shard()` (Kinesis.Client method), 2996
`spot_instance_request_id` (EC2.Instance attribute), 1614
`SQS.Client` (built-in class), 4214
`SQS.Message` (built-in class), 4241
`SQS.Queue` (built-in class), 4244
`SQS.ServiceResource` (built-in class), 4238
`sriov_net_support` (EC2.Image attribute), 1603
`sriov_net_support` (EC2.Instance attribute), 1615
`sse_customer_algorithm` (S3.Object attribute), 4090
`sse_customer_key_md5` (S3.Object attribute), 4090
`ssekms_key_id` (S3.Object attribute), 4090
`SSM.Client` (built-in class), 4260
`stack` (OpsWorks.Layer attribute), 3267
`Stack()` (CloudFormation.ServiceResource method), 155
`Stack()` (CloudFormation.StackResource method), 167
`Stack()` (OpsWorks.ServiceResource method), 3263
`Stack()` (OpsWorks.StackSummary method), 3277
`stack_id` (CloudFormation.Event attribute), 158
`stack_id` (CloudFormation.Stack attribute), 159
`stack_id` (CloudFormation.StackResource attribute), 167
`stack_id` (OpsWorks.Layer attribute), 3267
`stack_id` (OpsWorks.Stack attribute), 3271
`stack_id` (OpsWorks.StackSummary attribute), 3277
`stack_name` (CloudFormation.Event attribute), 158
`stack_name` (CloudFormation.Stack attribute), 159
`stack_name` (CloudFormation.StackResource attribute), 166
`stack_name` (CloudFormation.StackResourceSummary attribute), 167
`stack_status` (CloudFormation.Stack attribute), 160
`stack_status_reason` (CloudFormation.Stack attribute), 160
`StackResource()` (CloudFormation.ServiceResource method), 155
`StackResourceSummary()` (CloudFormation.ServiceResource method), 155
`stacks` (CloudFormation.ServiceResource attribute), 155
`stacks` (OpsWorks.ServiceResource attribute), 3263
`StackSummary()` (OpsWorks.ServiceResource method), 3263
`start()` (EC2.Instance method), 1635
`start()` (EC2.PlacementGroup method), 1682
`start()` (EC2.ServiceResource method), 1567
`start()` (EC2.Subnet method), 1733
`start()` (EC2.Vpc method), 1772
`start_configuration_recorder()` (ConfigService.Client method), 608
`start_gateway()` (StorageGateway.Client method), 4315
`start_instance()` (OpsWorks.Client method), 3243
`start_instances()` (EC2.Client method), 1445
`start_logging()` (CloudTrail.Client method), 423
`start_pipeline_execution()` (CodePipeline.Client method), 552
`start_stack()` (OpsWorks.Client method), 3243
`start_task()` (ECS.Client method), 1855
`start_time` (EC2.Snapshot attribute), 1711
`start_workflow_execution()` (SWF.Client method), 4520
`state` (EC2.Image attribute), 1603
`state` (EC2.Instance attribute), 1615
`state` (EC2.PlacementGroup attribute), 1678
`state` (EC2.Snapshot attribute), 1711
`state` (EC2.Subnet attribute), 1721
`state` (EC2.Volume attribute), 1744
`state` (EC2.Vpc attribute), 1758
`state_message` (EC2.Snapshot attribute), 1711
`state_reason` (EC2.Image attribute), 1603
`state_reason` (EC2.Instance attribute), 1615
`state_transition_reason` (EC2.Instance attribute), 1615
`status` (EC2.NetworkInterface attribute), 1670
`status` (EC2.VpcPeeringConnection attribute), 1787
`status` (IAM.AccessKey attribute), 2895
`status` (IAM.AccessKeyPair attribute), 2897
`status` (IAM.SigningCertificate attribute), 2952
`status` (S3.BucketVersioning attribute), 4074
`status_code` (Glacier.Job attribute), 2686
`status_message` (Glacier.Job attribute), 2686
`stop()` (EC2.Instance method), 1636
`stop()` (EC2.PlacementGroup method), 1684
`stop()` (EC2.ServiceResource method), 1569
`stop()` (EC2.Subnet method), 1735
`stop()` (EC2.Vpc method), 1774
`stop_configuration_recorder()` (ConfigService.Client method), 609
`stop_deployment()` (CodeDeploy.Client method), 504
`stop_instance()` (OpsWorks.Client method), 3244
`stop_instances()` (EC2.Client method), 1447
`stop_logging()` (CloudTrail.Client method), 424
`stop_stack()` (OpsWorks.Client method), 3244
`stop_task()` (ECS.Client method), 1860
`storage_class` (S3.MultipartUpload attribute), 4080
`storage_class` (S3.Object attribute), 4090
`storage_class` (S3.ObjectSummary attribute), 4108

- storage_class (S3.ObjectVersion attribute), 4122
 - StorageGateway.Client (built-in class), 4272
 - StorageGateway.Paginator.DescribeTapeArchives (built-in class), 4321
 - StorageGateway.Paginator.DescribeTapeRecoveryPoints (built-in class), 4323
 - StorageGateway.Paginator.DescribeTapes (built-in class), 4325
 - StorageGateway.Paginator.DescribeVTLDevices (built-in class), 4326
 - StorageGateway.Paginator.ListGateways (built-in class), 4328
 - StorageGateway.Paginator.ListVolumes (built-in class), 4329
 - strategy (EC2.PlacementGroup attribute), 1678
 - stream_specification (DynamoDB.Table attribute), 1001
 - StreamReaderProgress (class in boto3.s3.transfer), 4654
 - STS.Client (built-in class), 4331
 - submit_container_state_change() (ECS.Client method), 1863
 - submit_task_state_change() (ECS.Client method), 1864
 - subnet (EC2.Instance attribute), 1615
 - subnet (EC2.NetworkInterface attribute), 1671
 - subnet (EC2.RouteTableAssociation attribute), 1696
 - Subnet() (EC2.ServiceResource method), 1555
 - subnet_id (EC2.Instance attribute), 1615
 - subnet_id (EC2.NetworkInterface attribute), 1670
 - subnet_id (EC2.RouteTableAssociation attribute), 1695
 - subnet_id (EC2.Subnet attribute), 1721
 - subnets (EC2.ServiceResource attribute), 1590
 - subnets (EC2.Vpc attribute), 1785
 - subresources (boto3.resources.model.ResourceModel attribute), 4641
 - subscribe() (SNS.Client method), 4184
 - subscribe() (SNS.Topic method), 4212
 - subscribe_to_dataset() (CognitoSync.Client method), 591
 - Subscription() (SNS.ServiceResource method), 4194
 - subscriptions (SNS.ServiceResource attribute), 4195
 - subscriptions (SNS.Topic attribute), 4213
 - succeeded_jobs (Glacier.Vault attribute), 2711
 - suggest() (CloudSearchDomain.Client method), 413
 - Summary() (OpsWorks.Stack method), 3275
 - summary_map (IAM.AccountSummary attribute), 2901
 - Support.Client (built-in class), 4347
 - Support.Paginator.DescribeCases (built-in class), 4368
 - Support.Paginator.DescribeCommunications (built-in class), 4373
 - suspend() (S3.BucketVersioning method), 4075
 - suspend_processes() (AutoScaling.Client method), 94
 - swap_environment_cnames() (ElasticBeanstalk.Client method), 2082
 - SWF.Client (built-in class), 4375
 - SWF.Paginator.GetWorkflowExecutionHistory (built-in class), 4525
 - SWF.Paginator.ListActivityTypes (built-in class), 4566
 - SWF.Paginator.ListClosedWorkflowExecutions (built-in class), 4568
 - SWF.Paginator.ListDomains (built-in class), 4572
 - SWF.Paginator.ListOpenWorkflowExecutions (built-in class), 4573
 - SWF.Paginator.ListWorkflowTypes (built-in class), 4577
 - SWF.Paginator.PollForDecisionTask (built-in class), 4579
- ## T
- Table() (DynamoDB.ServiceResource method), 997
 - table_arn (DynamoDB.Table attribute), 1001
 - table_name (DynamoDB.Table attribute), 1001
 - table_size_bytes (DynamoDB.Table attribute), 1001
 - table_status (DynamoDB.Table attribute), 1001
 - tables (DynamoDB.ServiceResource attribute), 997
 - Tag() (EC2.ServiceResource method), 1555
 - tag_set (EC2.NetworkInterface attribute), 1671
 - tag_set (S3.BucketTagging attribute), 4072
 - Tagging() (S3.Bucket method), 4047
 - tags (CloudFormation.Stack attribute), 160
 - tags (EC2.DhcpOptions attribute), 1599
 - tags (EC2.Image attribute), 1603
 - tags (EC2.Instance attribute), 1615
 - tags (EC2.InternetGateway attribute), 1658
 - tags (EC2.NetworkAcl attribute), 1663
 - tags (EC2.RouteTable attribute), 1689
 - tags (EC2.SecurityGroup attribute), 1698
 - tags (EC2.Snapshot attribute), 1711
 - tags (EC2.Subnet attribute), 1721
 - tags (EC2.Volume attribute), 1744
 - tags (EC2.Vpc attribute), 1758
 - tags (EC2.VpcPeeringConnection attribute), 1787
 - target (boto3.resources.model.Parameter attribute), 4639
 - tell() (boto3.s3.transfer.ReadFileChunk method), 4654
 - terminate() (EC2.Instance method), 1638
 - terminate() (EC2.PlacementGroup method), 1686
 - terminate() (EC2.ServiceResource method), 1570
 - terminate() (EC2.Subnet method), 1737
 - terminate() (EC2.Vpc method), 1776
 - terminate_environment() (ElasticBeanstalk.Client method), 2083
 - terminate_instance_in_auto_scaling_group() (AutoScaling.Client method), 95
 - terminate_instances() (EC2.Client method), 1449
 - terminate_job_flows() (EMR.Client method), 2621
 - terminate_workflow_execution() (SWF.Client method), 4524
 - terminate_workspaces() (WorkSpaces.Client method), 4633
 - test_metric_filter() (CloudWatchLogs.Client method), 3075
 - test_role() (ElasticTranscoder.Client method), 2399

`timeout_in_minutes` (CloudFormation.Stack attribute), 160

`timestamp` (CloudFormation.Event attribute), 158

`Topic()` (SNS.ServiceResource method), 4194

`topic_configurations` (S3.BucketNotification attribute), 4064

`topics` (SNS.ServiceResource attribute), 4196

`transfer_domain()` (Route53Domains.Client method), 3918

`TransferConfig` (class in boto3.s3.transfer), 4654

`trigger_shutdown()` (boto3.s3.transfer.ShutdownQueue method), 4654

`type` (boto3.resources.model.ResponseResource attribute), 4641

`type` (OpsWorks.Layer attribute), 3267

U

`unassign_instance()` (OpsWorks.Client method), 3244

`unassign_private_ip_addresses()` (EC2.Client method), 1450

`unassign_private_ip_addresses()` (EC2.NetworkInterface method), 1677

`unassign_volume()` (OpsWorks.Client method), 3244

`unlink_developer_identity()` (CognitoIdentity.Client method), 574

`unlink_identity()` (CognitoIdentity.Client method), 574

`unmonitor()` (EC2.Instance method), 1640

`unmonitor()` (EC2.PlacementGroup method), 1687

`unmonitor()` (EC2.ServiceResource method), 1572

`unmonitor()` (EC2.Subnet method), 1739

`unmonitor()` (EC2.Vpc method), 1777

`unmonitor_instances()` (EC2.Client method), 1451

`unsubscribe()` (SNS.Client method), 4185

`unsubscribe_from_dataset()` (CognitoSync.Client method), 592

`update()` (CloudFormation.Stack method), 161

`update()` (DynamoDB.Table method), 1066

`update()` (IAM.AccountPasswordPolicy method), 2900

`update()` (IAM.AssumeRolePolicy method), 2903

`update()` (IAM.Group method), 2912

`update()` (IAM.LoginProfile method), 2923

`update()` (IAM.SamlProvider method), 2949

`update()` (IAM.ServerCertificate method), 2951

`update()` (IAM.User method), 2959

`update_access_key()` (IAM.Client method), 2829

`update_account_password_policy()` (IAM.Client method), 2829

`update_alias()` (KMS.Client method), 3023

`update_app()` (OpsWorks.Client method), 3245

`update_application()` (CodeDeploy.Client method), 504

`update_application()` (ElasticBeanstalk.Client method), 2086

`update_application_version()` (ElasticBeanstalk.Client method), 2088

`update_association_status()` (SSM.Client method), 4270

`update_assume_role_policy()` (IAM.Client method), 2831

`update_auto_scaling_group()` (AutoScaling.Client method), 96

`update_availability_options()` (CloudSearch.Client method), 399

`update_bandwidth_rate_limit()` (StorageGateway.Client method), 4316

`update_batch_prediction()` (MachineLearning.Client method), 3132

`update_chap_credentials()` (StorageGateway.Client method), 4316

`update_cloud_front_origin_access_identity()` (CloudFront.Client method), 262

`update_configuration_template()` (ElasticBeanstalk.Client method), 2089

`update_container_agent()` (ECS.Client method), 1865

`update_data_source()` (MachineLearning.Client method), 3133

`update_date` (IAM.Policy attribute), 2928

`update_default_branch()` (CodeCommit.Client method), 460

`update_deployment_group()` (CodeDeploy.Client method), 505

`update_distribution()` (CloudFront.Client method), 263

`update_domain_contact()` (Route53Domains.Client method), 3933

`update_domain_contact_privacy()` (Route53Domains.Client method), 3945

`update_domain_nameservers()` (Route53Domains.Client method), 3947

`update_elastic_ip()` (OpsWorks.Client method), 3248

`update_environment()` (ElasticBeanstalk.Client method), 2092

`update_evaluation()` (MachineLearning.Client method), 3134

`update_event_source_mapping()` (Lambda.Client method), 3043

`update_function_code()` (Lambda.Client method), 3045

`update_function_configuration()` (Lambda.Client method), 3046

`update_gateway_information()` (StorageGateway.Client method), 4317

`update_gateway_software_now()` (StorageGateway.Client method), 4318

`update_group()` (IAM.Client method), 2831

`update_health_check()` (Route53.Client method), 3861

`update_hosted_zone_comment()` (Route53.Client method), 3865

`update_identity_pool()` (CognitoIdentity.Client method), 575

`update_instance()` (OpsWorks.Client method), 3248

`update_item()` (DynamoDB.Client method), 902

`update_item()` (DynamoDB.Table method), 1071

- `update_job()` (ImportExport.Client method), 2977
- `update_key_description()` (KMS.Client method), 3023
- `update_layer()` (OpsWorks.Client method), 3250
- `update_login_profile()` (IAM.Client method), 2831
- `update_maintenance_start_time()` (StorageGateway.Client method), 4319
- `update_ml_model()` (MachineLearning.Client method), 3134
- `update_my_user_profile()` (OpsWorks.Client method), 3253
- `update_open_id_connect_provider_thumbprint()` (IAM.Client method), 2832
- `update_pipeline()` (CodePipeline.Client method), 553
- `update_pipeline()` (ElasticTranscoder.Client method), 2400
- `update_pipeline_notifications()` (ElasticTranscoder.Client method), 2413
- `update_pipeline_status()` (ElasticTranscoder.Client method), 2420
- `update_radius()` (DirectoryService.Client method), 753
- `update_rds_db_instance()` (OpsWorks.Client method), 3253
- `update_records()` (CognitoSync.Client method), 593
- `update_repository_description()` (CodeCommit.Client method), 461
- `update_repository_name()` (CodeCommit.Client method), 461
- `update_saml_provider()` (IAM.Client method), 2833
- `update_scaling_parameters()` (CloudSearch.Client method), 401
- `update_server_certificate()` (IAM.Client method), 2833
- `update_service()` (ECS.Client method), 1868
- `update_service_access_policies()` (CloudSearch.Client method), 403
- `update_signing_certificate()` (IAM.Client method), 2834
- `update_snapshot_schedule()` (StorageGateway.Client method), 4320
- `update_ssh_public_key()` (IAM.Client method), 2834
- `update_stack()` (CloudFormation.Client method), 139
- `update_stack()` (OpsWorks.Client method), 3253
- `update_streaming_distribution()` (CloudFront.Client method), 291
- `update_table()` (DynamoDB.Client method), 926
- `update_tags_for_domain()` (Route53Domains.Client method), 3948
- `update_trail()` (CloudTrail.Client method), 424
- `update_user()` (IAM.Client method), 2835
- `update_user_profile()` (OpsWorks.Client method), 3257
- `update_volume()` (OpsWorks.Client method), 3258
- `update_vtl_device_type()` (StorageGateway.Client method), 4320
- `upload()` (S3.MultipartUploadPart method), 4086
- `upload_archive()` (Glacier.Client method), 2672
- `upload_archive()` (Glacier.Vault method), 2703
- `upload_date` (IAM.SigningCertificate attribute), 2952
- `upload_documents()` (CloudSearchDomain.Client method), 414
- `upload_file()` (boto3.s3.transfer.MultipartUploader method), 4653
- `upload_file()` (boto3.s3.transfer.S3Transfer method), 4654
- `upload_file()` (in module boto3.s3.inject), 4651
- `upload_file()` (S3.Bucket method), 4041
- `upload_file()` (S3.Client method), 4019
- `upload_file()` (S3.Object method), 4090
- `upload_id` (S3.MultipartUpload attribute), 4080
- `upload_multipart_part()` (Glacier.Client method), 2674
- `upload_part()` (Glacier.MultipartUpload method), 2694
- `upload_part()` (S3.Client method), 4019
- `UPLOAD_PART_ARGS` (boto3.s3.transfer.MultipartUploader attribute), 4653
- `upload_part_copy()` (S3.Client method), 4020
- `upload_server_certificate()` (IAM.Client method), 2835
- `upload_signing_certificate()` (IAM.Client method), 2837
- `upload_ssh_public_key()` (IAM.Client method), 2838
- `url` (SQS.Queue attribute), 4244
- `use_custom_cookbooks` (OpsWorks.Stack attribute), 3271
- `use_ebs_optimized_instances` (OpsWorks.Layer attribute), 3267
- `use_opsworks_security_groups` (OpsWorks.Stack attribute), 3271
- `user` (IAM.CurrentUser attribute), 2905
- `user` (IAM.VirtualMfaDevice attribute), 2971
- `User()` (IAM.AccessKey method), 2896
- `User()` (IAM.LoginProfile method), 2924
- `User()` (IAM.MfaDevice method), 2926
- `User()` (IAM.ServiceResource method), 2883
- `User()` (IAM.SigningCertificate method), 2953
- `User()` (IAM.UserPolicy method), 2969
- `user_attribute` (IAM.VirtualMfaDevice attribute), 2970
- `user_id` (IAM.CurrentUser attribute), 2904
- `user_id` (IAM.User attribute), 2955
- `user_name` (IAM.AccessKey attribute), 2895
- `user_name` (IAM.AccessKeyPair attribute), 2897
- `user_name` (IAM.CurrentUser attribute), 2905
- `user_name` (IAM.LoginProfile attribute), 2922
- `user_name` (IAM.MfaDevice attribute), 2924
- `user_name` (IAM.SigningCertificate attribute), 2952
- `user_name` (IAM.User attribute), 2955
- `user_name` (IAM.UserPolicy attribute), 2968
- `UserPolicy()` (IAM.ServiceResource method), 2883
- `users` (IAM.Group attribute), 2915
- `users` (IAM.ServiceResource attribute), 2891

V

`valid_until` (IAM.SamlProvider attribute), 2948

`validate_configuration_settings()` (ElasticBeanstalk.Client method), 2097
`validate_pipeline_definition()` (DataPipeline.Client method), 633
`validate_template()` (CloudFormation.Client method), 142
`value` (boto3.resources.model.Parameter attribute), 4639
`value` (EC2.Tag attribute), 1741
`Vault()` (Glacier.Account method), 2680
`Vault()` (Glacier.Archive method), 2684
`Vault()` (Glacier.Job method), 2689
`Vault()` (Glacier.MultipartUpload method), 2695
`Vault()` (Glacier.Notification method), 2698
`Vault()` (Glacier.ServiceResource method), 2677
`vault_arn` (Glacier.Job attribute), 2687
`vault_arn` (Glacier.MultipartUpload attribute), 2690
`vault_arn` (Glacier.Vault attribute), 2699
`vault_name` (Glacier.Archive attribute), 2681
`vault_name` (Glacier.Job attribute), 2685
`vault_name` (Glacier.MultipartUpload attribute), 2690
`vault_name` (Glacier.Notification attribute), 2696
`vault_name` (Glacier.Vault attribute), 2699
`vaults` (Glacier.Account attribute), 2680
`vaults` (Glacier.ServiceResource attribute), 2677
`verify_domain_dkim()` (SES.Client method), 4162
`verify_domain_identity()` (SES.Client method), 4162
`verify_email_address()` (SES.Client method), 4163
`verify_email_identity()` (SES.Client method), 4163
`Version()` (S3.Object method), 4102
`Version()` (S3.ObjectSummary method), 4119
`version_id` (IAM.PolicyVersion attribute), 2937
`version_id` (S3.Object attribute), 4090
`version_id` (S3.ObjectVersion attribute), 4122
`Versioning()` (S3.Bucket method), 4047
`versions` (IAM.Policy attribute), 2935
`virtual_mfa_devices` (IAM.ServiceResource attribute), 2893
`virtualization_type` (EC2.Image attribute), 1603
`virtualization_type` (EC2.Instance attribute), 1615
`VirtualMfaDevice()` (IAM.ServiceResource method), 2884
`volume` (EC2.Snapshot attribute), 1711
`Volume()` (EC2.ServiceResource method), 1555
`volume_configurations` (OpsWorks.Layer attribute), 3267
`volume_id` (EC2.Snapshot attribute), 1711
`volume_id` (EC2.Volume attribute), 1744
`volume_size` (EC2.Snapshot attribute), 1711
`volume_type` (EC2.Volume attribute), 1744
`volumes` (EC2.Instance attribute), 1640
`volumes` (EC2.ServiceResource attribute), 1592
`vpc` (EC2.Instance attribute), 1615
`vpc` (EC2.NetworkAcl attribute), 1663
`vpc` (EC2.NetworkInterface attribute), 1671
`vpc` (EC2.RouteTable attribute), 1689
`vpc` (EC2.Subnet attribute), 1721
`Vpc()` (EC2.ServiceResource method), 1555
`vpc_id` (EC2.Instance attribute), 1615
`vpc_id` (EC2.NetworkAcl attribute), 1663
`vpc_id` (EC2.NetworkInterface attribute), 1671
`vpc_id` (EC2.RouteTable attribute), 1689
`vpc_id` (EC2.SecurityGroup attribute), 1698
`vpc_id` (EC2.Subnet attribute), 1721
`vpc_id` (EC2.Vpc attribute), 1758
`vpc_id` (OpsWorks.Stack attribute), 3271
`vpc_peering_connection_id` (EC2.VpcPeeringConnection attribute), 1787
`vpc_peering_connections` (EC2.ServiceResource attribute), 1594
`VpcPeeringConnection()` (EC2.ServiceResource method), 1555
`vpcs` (EC2.ServiceResource attribute), 1596

W

`wait()` (CloudFront.Waiter.DistributionDeployed method), 316
`wait()` (CloudFront.Waiter.InvalidationCompleted method), 316
`wait()` (CloudFront.Waiter.StreamingDistributionDeployed method), 317
`wait()` (DynamoDB.Waiter.TableExists method), 975
`wait()` (DynamoDB.Waiter.TableNotExists method), 976
`wait()` (EC2.Waiter.BundleTaskComplete method), 1489
`wait()` (EC2.Waiter.ConversionTaskCancelled method), 1490
`wait()` (EC2.Waiter.ConversionTaskCompleted method), 1491
`wait()` (EC2.Waiter.ConversionTaskDeleted method), 1492
`wait()` (EC2.Waiter.CustomerGatewayAvailable method), 1492
`wait()` (EC2.Waiter.ExportTaskCancelled method), 1494
`wait()` (EC2.Waiter.ExportTaskCompleted method), 1494
`wait()` (EC2.Waiter.ImageAvailable method), 1494
`wait()` (EC2.Waiter.InstanceExists method), 1496
`wait()` (EC2.Waiter.InstanceRunning method), 1501
`wait()` (EC2.Waiter.InstanceStatusOk method), 1505
`wait()` (EC2.Waiter.InstanceStopped method), 1506
`wait()` (EC2.Waiter.InstanceTerminated method), 1510
`wait()` (EC2.Waiter.PasswordDataAvailable method), 1514
`wait()` (EC2.Waiter.SnapshotCompleted method), 1515
`wait()` (EC2.Waiter.SpotInstanceRequestFulfilled method), 1517
`wait()` (EC2.Waiter.SubnetAvailable method), 1519
`wait()` (EC2.Waiter.SystemStatusOk method), 1520
`wait()` (EC2.Waiter.VolumeAvailable method), 1522
`wait()` (EC2.Waiter.VolumeDeleted method), 1524
`wait()` (EC2.Waiter.VolumeInUse method), 1525

- wait() (EC2.Waiter.VpcAvailable method), [1527](#)
- wait() (EC2.Waiter.VpnConnectionAvailable method), [1528](#)
- wait() (EC2.Waiter.VpnConnectionDeleted method), [1530](#)
- wait() (ECS.Waiter.ServicesInactive method), [1879](#)
- wait() (ECS.Waiter.ServicesStable method), [1880](#)
- wait() (ECS.Waiter.TasksRunning method), [1880](#)
- wait() (ECS.Waiter.TasksStopped method), [1881](#)
- wait() (ElastiCache.Waiter.CacheClusterAvailable method), [2037](#)
- wait() (ElastiCache.Waiter.CacheClusterDeleted method), [2038](#)
- wait() (ElastiCache.Waiter.ReplicationGroupAvailable method), [2038](#)
- wait() (ElastiCache.Waiter.ReplicationGroupDeleted method), [2039](#)
- wait() (ElasticTranscoder.Waiter.JobComplete method), [2536](#)
- wait() (EMR.Waiter.ClusterRunning method), [2633](#)
- wait() (IAM.Waiter.InstanceProfileExists method), [2872](#)
- wait() (IAM.Waiter.UserExists method), [2872](#)
- wait() (Kinesis.Waiter.StreamExists method), [3001](#)
- wait() (MachineLearning.Waiter.BatchPredictionAvailable method), [3151](#)
- wait() (MachineLearning.Waiter.DataSourceAvailable method), [3153](#)
- wait() (MachineLearning.Waiter.EvaluationAvailable method), [3154](#)
- wait() (MachineLearning.Waiter.MLModelAvailable method), [3155](#)
- wait() (RDS.Waiter.DBInstanceAvailable method), [3586](#)
- wait() (RDS.Waiter.DBInstanceDeleted method), [3587](#)
- wait() (RDS.Waiter.DBSnapshotCompleted method), [3588](#)
- wait() (Redshift.Waiter.ClusterAvailable method), [3807](#)
- wait() (Redshift.Waiter.ClusterDeleted method), [3808](#)
- wait() (Redshift.Waiter.SnapshotAvailable method), [3809](#)
- wait() (Route53.Waiter.ResourceRecordSetsChanged method), [3876](#)
- wait() (S3.Waiter.BucketExists method), [4032](#)
- wait() (S3.Waiter.BucketNotExists method), [4032](#)
- wait() (S3.Waiter.ObjectExists method), [4032](#)
- wait() (S3.Waiter.ObjectNotExists method), [4033](#)
- wait() (SES.Waiter.IdentityExists method), [4165](#)
- wait_until_completed() (EC2.Snapshot method), [1718](#)
- wait_until_exists() (EC2.Instance method), [1642](#)
- wait_until_exists() (S3.Bucket method), [4054](#)
- wait_until_exists() (S3.Object method), [4102](#)
- wait_until_exists() (S3.ObjectSummary method), [4119](#)
- wait_until_not_exists() (S3.Bucket method), [4054](#)
- wait_until_not_exists() (S3.Object method), [4103](#)
- wait_until_not_exists() (S3.ObjectSummary method), [4120](#)
- wait_until_running() (EC2.Instance method), [1646](#)
- wait_until_stopped() (EC2.Instance method), [1650](#)
- wait_until_terminated() (EC2.Instance method), [1654](#)
- Waiter (class in boto3.resources.model), [4641](#)
- waiter_name (boto3.resources.model.Waiter attribute), [4642](#)
- WaiterAction (class in boto3.resources.action), [4644](#)
- waiters (boto3.resources.model.ResourceModel attribute), [4641](#)
- Website() (S3.Bucket method), [4047](#)
- website_redirect_location (S3.Object attribute), [4090](#)
- WorkSpaces.Client (built-in class), [4621](#)